

Han Youjia

Website Programming Based on J2EE

Bachelor's Thesis
Information technology


January 2011



MIKKELIN AMMATTIKORKEAKOULU

Mikkeli University of Applied Sciences

DESCRIPTION

 <p>MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences</p>		Date of the bachelor's thesis 10.12.2010
Author(s) Han Youjia		Degree programme and option Information Technology
Name of the bachelor's thesis Website Programming Based on J2EE		
Abstract <p>During the last decade, websites have become increasing popular all over the world. Almost every company has its own website. The tools for web developers are becoming more diversified. In this thesis, I choose J2EE technology to develop a website.</p> <p>The website implemented includes a login system, an online shopping system and serach system. Online shoping is very popular nowadays not only among young people but also with middle aged and elderly people. Online shopping is also the most difficult part of in this website.</p> <p>In my thesis, I used J2EE technology to establish a website on Apache Tomcat and I used JSP to develop webpage, Servlet cope with the web logic and MySQL as my database. Users can buy books from this website and find suitable methods to help them get a brighter day.</p>		
Subject headings, (keywords) J2EE JSP MySQL Tomcat Servlet API HTTP HTML JDK EJB		
Pages Pag51+Res65	Language English	URN
Remarks, notes on appendices		
Tutor Matti Koivisto		Employer of the bachelor's thesis Mikkeli University of Applied Science

ACKNOWLEDGEMENT

First and foremost, I would like to show my deepest gratitude to my supervisor, Matti Koivisto, a respectable, responsible and resourceful scholar, who has provided me with valuable guidance in every stage of the writing of this thesis. Without his enlightening instruction, impressive kindness and patience, I could not have completed my thesis. His keen and vigorous academic observation enlightens me not only in this thesis but also in my future study.

I shall extend my thanks to my friend and classmate Yan Liang for all his kindness and help. I would also like to thank all my teachers who have helped me to develop the fundamental and essential academic competence.

Last but not least, I'd like to thank all my friends, especially my two nice roommates Lv ShuaiYi and Yan Liang for their encouragement and support.

LIST OF ABBREVIATION

API	Application Programming Interface
PHP	Hypertext Preprocessor
J2EE	Java 2 Platform, Enterprise Edition
XML	Extensible Markup Language
PERL	Practical Extraction and Report Language
CGI	Common Gate Interface
HTML	Hyper Text Mark-up Language
POD	PHP Data Objects
JDBC	Java Data Base Connectivity
EJB	Enterprise JavaBeans
JSP	Java Server Pages
JDK	Java Development Kit
WAN	Wide Area Network
LAN	Local Area Network
JSP	Java Server Pages
HTTP	HyperText Transfer Protocol
SQL	Structured Query Language
CGI	common gateway interface
CSS	Cascading Style Sheet
SWF	Shock Wave Flash

LIST OF FIGURES AND TABLES

Table 2.1 List of famous website and their main programming tools.....	5
Figure 2.2 PHP Logo.....	5
Table 2.3 The versions of .Net framework.....	6
Figure 2.4 Python logo.....	8
Figure 3.1 Basic J2EE structure	10
Figure 3.2 MyEclipse logo	12
Figure 4.1 Simling face.....	13
Figure 5.1 Structure of B/S.....	14
Figure 5.2 Structure of C/S	15
Figure 5.3 Tomcat logo	16
Figure 5.4 JSP.....	18
Figure 5.5 JSP structure.....	19
Figure 5.6 MySQL logo	20
Figure 6.1 The result showing the successful configuration of Servlet and Tomcat.....	23
Figure 6.2 Turn on the webserver.....	25
Figure 6.3 Turn on the MySQL server.....	25
Figure 6.4 Redeploy the project	26
Figure 7.1 Structure of website	28
Figure 7.2 Structure of Programming a Web site.....	29
Figure 7.3 Login page.....	30
Figure 7.4 Welcome page.....	30
Figure 7.5 show IP address and computer name.....	31
Figure 7.6 The visited number changes	33
Figure 7.7 The main page	32
Figure 7.8 The modify page	34
Figure 7.9 The update page.....	34
Figure 7.10 The congratulations' page.....	36
Figure 7.11 The modify page after editing	37

Figure 7.12 The notice window	37
Figure 7.13 The calculate page	38
Figure 7.14 Index page framework.....	39
Figure 7.15 Head Page Framework	39
Figure 7.16 Tail Page Framework	39
Figure 7.17 Left Page Framework	40
Figure 7.18 Right Page Framework	40
Figure 7.19 The Framework of showgoodsdetail	42
Figure 7.20 The Framework of showMycart.jsp	43
Figure 7.21 The Form of shoppingCl2.jsp.....	44
Figure 7.22 The Form of shopping3.jsp	44
Figure 7.23 The Form of shopping4.jsp.....	45
Figure7.24 Enough text filed	46
Figure 7.25 the framework of enough page	47
Figure 7.26 The index page	47
Figure 7.27 The Book page	48
Figure 7.28 The book detail page	48
Figure 7.29 The ShowMycart page.....	49
Figure 7.30 Login Page	49
Figure 7.31 Shopping3 Page	50
Figure 7.32 shopping4 page	50

TABLE OF CONTENTS

1 INTRODUCTION.....	1
2 SOFTWARE FOR WEBSITE DEVELOPMENT AND DESIGN.....	3
2.1 Overview.....	3
2.2 PHP technology.....	4
2.2.1 What is PHP?	4
2.2.2 Brief introduce of PHP language.....	4
2.2.3 The development of PHP.....	4
2.2.4 Characters of PHP.....	5
2.3 .Net technology.....	6
2.3.1 What is .Net?	6
2.3.2 The characters of .Net	6
2.4 Python technology	7
2.4.1 What is python?	7
2.4.2 The character of python.....	7
2.4.3 Status of python in programming language.....	8
3 PROGRAMMING ENVIRONMENT	9
3.1 Java enterprise edition.....	9
3.2 Java platform enterprise edition software development kit.....	11
3.3 MyEclipse 8.5.....	11
4 HOW TO MAKE US HAPPY.....	12
4.1 What is happiness?	12
4.2 What make us happy?	12
4.3 The characteristics of happiness	13
5 SYSTEM DEVELOPMENT TECHNOLOGIES.....	14
5.1 The advantage of browser server model	14
5.2 The tomcat server	16
5.3 Servlet technology	17
5.4 JSP technology	18

5.4.1 Advantages of JSP over competing technologies.....	19
5.4.2 Drawbacks of JSP	19
5.5 My database MySQL	20
5.5.1 The characteristics of MySQL.....	20
6 IMPLEMNETING A WEB SITE BASE ON J2EE	21
6.1 Connection	21
6.6 Web page design	24
7 PROGRAMMING A WEB SITE BASE ON J2EE	27
7.1 The structure of the web site.....	27
7.2 The structure of running a web site.....	28
7.3 The Login Page.....	29
7.4 Show visitors IP address and PC name	30
7.4.1Show request number	30
7.5 User Management System	32
7.5.1 Delete Users	36
7.5.2 Calculated Technology	37
7.6 Web Page Design	37
7.7 The result of the web site	46
8 CONCLUSION	50
8.1 Challenges and solutions	50
8.2 Possible future work	51
REFERENCE	52
APPENDICES.....	55
Appendix A-1: ConnDB.Java	56
Appendix A-2:enoughBean.java	57
Appendix A-3:enoughBeanBO.Java.....	58
Appendix A-4:GoodsBean.java.....	59
Appendix A-5:GoodBeanBo.java.....	61
Appendix A-6:MycartBO.java.....	64
Appendix A-7:orderBean.java.....	67

Appendix A-8:orderBeanBO.java.....	68
Appendix A-9:orderdetailBean.java.....	71
Appendix A-10:orderInfoBean.java.....	72
Appendix A-11:UserBean.java.....	74
Appendix A-12:UserBeanBO.java.....	76
Appendix B-1:EnoughCl.java.....	78
Appendix B-2:LoginCl.java.....	79
Appendix B-3: OrderClServlet.java.....	80
Appendix B-4:ShoppingCl2.java.....	81
Appendix B-5:ShoppingClServlet.java.....	82
Appendix B-6:ShowGoodsServlet.java.....	84
Appendix C-1:book.jsp.....	85
Appendix C-2:enough.jsp.....	86
Appendix C-3:head.jsp.....	87
Appendix C-4:index.jsp.....	88
Appendix C-5:left.jsp.....	90
Appendix C-6:right.jsp.....	92
Appendix C-7:shopping2.jsp.....	94
Appendix C-8: shopping3.jsp.....	96
Appendix C-9: shopping4.jsp.....	99
Appendix C-10:showdetail.jsp.....	101
Appendix C-11:showmycart.jsp.....	103
Appendix C-12:tail.jsp.....	106

1 INTRODUCTION

With the proliferation of advanced technology, people are now living in a more comfortable environment. But so many people feel less sense of happiness. So what makes us happy comes to a question? Based on a study of TIME magazine, typical Americans are overwhelmingly happy and optimistic people, regardless of income [1] However people under high pressure of the society and work cannot find any sense of happiness.

In this thesis, I want to establish a website to help people to get rid of their bad mood and to maintain positive and optimistic attitudes towards lives. My website can hopefully give people some efficient suggestion to face their problems that they encounter in their studies, work, or other area of their life.

There are several website exploitation tools that are very popular in the world. The most popular ones are:

- J2EE Java 2 Enterprise Edition from Oracle
- PHP A hypertext Preprocessor produces dynamic web pages
- .NET A Microsoft XML Web service platform
- PYTHON A General-purpose high-level programming language designed by Guido van Rossum [2]

I have chosen the first exploit tool, J2EE to process this job.

The aim of the study has two parts, the first one is to get familiar with the J2EE and then using this tool to set up a website. The second part is to learn some knowledge of human psychology in order to give advice to these people who need help. As an engineering student the main focus is naturally on the technical side of the project.

The structure of the study is as follows:

In Chapter2, I introduce several popular website platforms. These tools are software engineers' most important building blocks when programming a website. In Chapter3, I make a short introduction of J2EE which is my favorite software and the programming environment. In chapter4, I make a basic introduction of happiness and what can makes people happy. In chapter 5 I introduce the system development technology and my developing tools such as Tomcat and Mysql. In chapter 6 I discuss the method of developing a website and provide my running environment. In chapter 7 I introduce the coding method and the whole structure of the website.

In chapter 8 I make some fined conclusion and mention the problems and challenges I face and how do I overcome them. Furthermore, I introuduce some information about the future work of this website.

2 PROGRAMMING LANGUAGE FOR WEBSITE DEVELOPMENT AND DESIGN

2.1 Overview

We have been considered for a long time, what kind of software we will use when we want to develop a powerful website. There are many alternatives available like J2EE, .Net or Ruby, Python and PHP? Each of these programming languages have the whole system, including a platform, a server, and a framework. So without any doubts, grasp any of these programming language could establish a perfect website.

However, I want to find out the most suitable programming language for my needs. So I make a list of famous websites in the world and then check what kind of programming language they use. The result of my survey is listed in Table 1.1 and it clearly highlights that there is no single right answer to the question what is the best tools for website development.

Table 2.1 List of famous website and their main programming tools

Name of the site	Programming language
Dangdang.com	.Net
Ganji.com	.Net
Amazon.com	Java and ASP
Facebook.com	PHP
Youtube.com	Python
Twitter.com	Ruby

As far as I know, Java and .Net are the most perfect exploiting 'platform' for enterprises. They provide not only perfect programming language but also include a full set of methods for people to solve problems.

With these kinds of tools, we can have many choices from webpage design, reflection and remote access. Furthermore Java has a lot of open source programs which could serve as demo for my own project [3]

2.2 PHP Technology

2.2.1 What is PHP?

PHP is an abbreviation of hypertext preprocessor, a server side, cross-platform, HTML embedded scripting language. The language style of PHP is very similar to C programming language. It has been widely applied in each level of internet field[4].

2.2.2 Brief introduction of PHP language

The grammar of PHP language is a blend of C, Java, Perl and PHP's own grammar. The speed of executing dynamic web pages is faster than with PERL or CGI. Comparing with other programming language in the field of dynamic web pages, PHP embeds its programs to the HTML document. The efficiency of this method is higher than generative complete HTML tags. PHP also can execute the code after it is compiled, which means it can make codes to operate faster. Beside of these, PHP is powerful; it almost can replace of CGI and perfectly supports the mainstream databases and operating systems[4].

2.2.3 The Development of PHP

PHP was created by Rasmus Lerdorf in 1994. It was first a simple program written with Perl for the purpose of counting the number of website visitors. Then he rewrote the program using C language, which provided the program with an access to the databases. Lerdorf released the first version in 1995. In the same year, Lerdorf released the second version which including the functions to support MySQL. So that established a dominant position for PHP in dynamic web pages. At the end of 1996, PHP was used by 15000 websites. In 1997 the number of PHP users overtakes 50000. In the same year, PHP3 was released. In 2000, PHP4 and PHP5 were both released, PHP5 included of several functions such as new object mode, import POD, an extension library to the access database [4]

2.2.4 Characters of PHP

Comparing with other technologies, PHP is a free platform and all the source codes are open to the public. PHP is more suitable for beginners, because it is an embed HTML language, easy to edit and has a strong practicability. PHP can perfectly run in many operating systems such as Windows, Linux and Unix. It has very high efficiency but consume little system resources. The logo of PHP is shown in Figure2.2

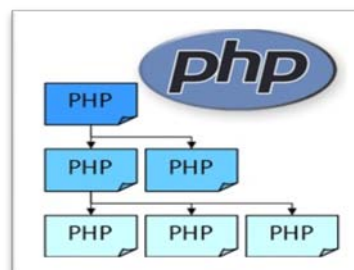


Figure 2.2 PHP LOGO [5]

2.3 .Net Technology

2.3.1 What is .Net?

.Net is a platform of Microsoft XML Web services. XML Web services allow internet to communicate and share data, whatever kind of operating system, programming language and equipment you use. .Net provides the function of creating XML Web services and concentrates all services together. [6]

2.3.2 The characters of .Net

.Net framework is using virtual machine technology and based on Common Language Runtime support. This means that programmers can use many programming languages such as C#, VB.NET, C++, etc.

.Net also affords many new functions and technologies with its API. Using these new features software engineers can develop Windows application software, network application software and web server at the same time. .NET provides a new reflected and object oriented interface. [6] Table 2.3 lists the development versions of .Net environment.

Figure 2.3 The versions of .Net framework [6]

Version	Version Number	Release Date	Visual Studio	Default in Windows
1.0	1.0.3705.0	2002-02-13	Visual Studio .NET	
1.1	1.1.4322.573	2003-04-24	Visual Studio .NET 2003	Windows Server 2003
2.0	2.0.50727.42	2005-11-07	Visual Studio 2005	Windows Server 2003 R2
3.0	3.0.4506.30	2006-11-06		WindowsVista, Windows Server 2008
3.5	3.5.21022.8	2007-11-19	Visual Studio 2008	Windows7, Windows Server 2008 R2
4.0	3.0.4506.30	2010-04-12	Visual Studio 2010	

2.4 Python Technology

2.4.1 What is Python?

Python is designed by Guido van Rossum during Christmas time in 1989. Python is an object-oriented programming language. It is a powerful and well-size-fits-all language. It has been developed for more than a decade, and therefore it is mature and stable technology. Python is a scripting language with rich and powerful class library to support the vast majority of day-to-day applications. Figure 2.4 show the logo of Python.[7]



Figure 2.4 Python Logo [8]

This language has very simple and clear grammar, for a variety of high-level mission to complete. It runs on almost all operating system to run. At present, this language is related to the rapid technological development, the rapid expansion of the number of users, and related resources.[9]

2.4.2 The characteristic of Python

Extendable is one of the characteristic of Python as a programming language. New inner modules can be written by C and C++ and we also can add interface to the completed module. Python can help users to bypass the difficult grammar which typically takes a lot of time. Python is a very clear programming language as Tim Peter said “There should be one-- and preferably only one --obvious way to do it” [8]. This characteristic of Python is quite the opposite of the Perl language. The other interesting feature of Python is the designer deliberately making the language very serious so that people with bad programming habits cannot use it.

A big difference between Python and other programming languages is that each module bounds is decided by the position of the first letter of each line. [9]

2.4.3 Status of Python in programming language

Python has been implicated in many large scale software developments such as Zope, Mnet, BitTorrent and Google. When Python is executing, it will translate the code of .py file to Python's byte code and then using Python virtual machine to execute these codes. This idea is very similar to Java and .Net, but the difference is Python's virtual machine is better than Java's and .Net's. This does not mean Python's virtual machine is more powerful than others but rather the distance between virtual machine and true machine is longer. In other words, Python's virtual machine is at a high abstract level. [9]

In the development environment, Python also is called glue language. It does not mean this language will stick on your fingers but rather means it is easy to make its module connected to other modules especially written in C++ and C. It has a high speed in creating program prototypes and can rewrite the code [7]

3 PORGRAMMING ENVIRONMENT

3.1 Java Enterprise Edition

In my project, I will use Java as my programming language. To develop Java applications for web sites, developers have to use J2EE as their tool. J2EE is using Java2 platform to simplify enterprise plan, solve and arrange complex problems. J2EE is extension of Java so it is inherited from the core technology of Java. It makes users easy to access APIS and JDBC databases and can protect data in internet applications. Moreover it supports EJB, Java Servlets, JSP and XML technologies [10].

J2EE provides middle level framework to help enterprises to achieve high usability, reliability, expandability in the shortest time. J2EE reduces the complexity and cost of developing multi layer applications and it perfectly supports Enterprise JavaBeans. The basic J2EE structure shown in Figure 3.1 will be introduced in Chapter5.

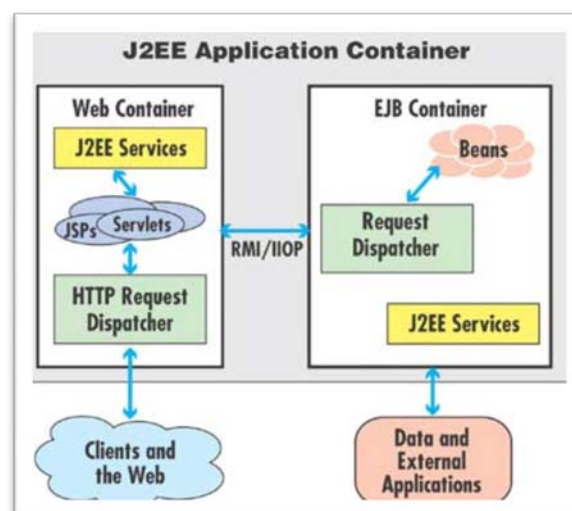


Figure 3.1 Basic J2EE structure [11]

3.2 Java Platform Enterprise Edition Software Development Kit

If you want to develop a Java program, you have to download a JDK which is the core technology in Java. Without JDK you cannot do anything even you have installed Java developing software. Different operating systems need different JDKs and the official Java website has clearly description of this. There are three kinds of JDKs. J2SE, J2ME, and J2EE. J2EE is a common version and after JDK 5.0 release, JDK 5.0 changes its name to J2EE. J2SE and J2ME also in the same situation, both of them change its name from JDK 5.0 to J2SE and J2ME separately [12]

J2EE platform SDK provides not only a development environment for writing applets and applications but also a collection of developing tools. JDK consist of javac, a compiler which help translate source program to byte code; javah which provides a function to transfer C code or building a connection between C and Java; jdb help user to find out bugs and solve it; jar which can compress several class file into one class file; appletviewer which is small java program browser, execute java program in HTML file. [12]

3.3 MyEclipse 8.5

MyEclipse is a commercially available Java EE and AJAX IDE created and maintained by the company Genuitec, a founding member of the Eclipse Foundation. It is built upon the Eclipse platform, and integrates both proprietary and open source solutions into the development environment. MyEclipse has two primary versions (apart from the "Blue Edition and "MyEclipse Spring Edition referred to below): a professional and a standard edition. [13]

The standard edition adds database tools, a visual web designer, persistence tools, Spring tools, Struts and JSF tooling, and a number of other features to the basic Eclipse Java Developer profile. It competes with the Web Tools Project, which is a part of Eclipse itself, but MyEclipse is a separate project entirely and offers a different feature set.[13]

Most recently, MyEclipse has been made available via Pulse (ALM), a provisioning tool that maintains Eclipse software profiles, including those that use MyEclipse. Additionally, MyEclipse is offering a customized version for IBM products, "MyEclipse Blue Edition", that adds specific support for Rational Software and WebSphere development. Currently, MyEclipse Blue Edition is available for Windows and Linux, though Mac is unsupported [13]. You can see the logo of MyEclipse in Figure 3.2



Figure 3.2 MyEclipse logo [14]

4 HOW TO MAKES US HAPPY?

4.1 What is happiness ?

The definition of happiness is different from people to people. Some people like to relate happiness to money; they think money can buy happiness. However, many others like to relate happiness to spiritual enjoyment. In my opinion, Happiness is a state of mind or feeling characterized by contentment, love, satisfaction, pleasure, or joy. A variety of biological, psychological, religious, and philosophical approaches have striven to define happiness and identify its sources. [15] We can also connect happiness to some well-known symbols like a smiling face show in Figure 4.1.



Figure 4.1 Smiling face[16]

4.2 What make us happy?

So what makes individuals heart delight? Hold wealth, for example, and money can buy all the attractive stuff. A former research [17] shows when people get the basis needs; additional revenue cannot improve your sense of satisfaction in lives. A nice educational history? Not at all. Youth? No again, a recent research [17] indicates

older people are more consistently satisfied with their lives than the young. And they are less prone to dark moods. A recent survey by the Centers for Disease Control and Prevention found that people ages 20 to 24 are sad for an average of 3.4 days a month, as opposed to just 2.3 days for people ages 65 to 74 [17] If you feel happiness is directly related to material possessions, you are making a mistake. Happiness can be gained just through being content with a small amount of possessions. Our inner wealth is not just about what we have, but what we are content to live without.[17] This is true the moment when we accomplish our desire or get something new such as get kid, a new car, new house, new mobile or better vocation we will feel happy. When you recall some nice experience in your mind, you will feel happy also. So we can get happiness through realizing our exceptions.

4.3 The characteristics of happiness

All the sense of happiness human can get are temporary, just like bad or good luck. With the passage of time, happiness and unfortunate will slowly and gradually disappeared. So if people want to keep his happiness, he has to accomplish more desire. Once you have one type of happiness, the next time when you get the same type of happiness, you will feel less happy than the first time. At last, you will gradually lose it. The other character is that the more hardship people are subjected the more happiness people will easily get. If someone does not have desire, he will not feel a sense of happiness. The last character of happiness is that when you process a lot of material wealth, it's very hard for them to get happiness from field of materials, but you can get happiness from spirit level. [18]

5 SYSTEM DEVELOPMENT TECHNOLOGIES

5.1 The advantage of browser server model

With the proliferation of internet, the browser server model (B/S) is the revolution of client server (C/S) Structure. In this kind of structure, most of the work is done by server and the users' interface is the browser (see Figure 5.1). Users can have a access to data, pictures, Flash animations and videos through the browser. In this way, not only computers' burdens on the user side are reduced but costs from updating and system maintenance on the company side are also reduced. As a recent technology, a local network established with B/S structure using network applications and databases with SQL under internet mode is easy to handle and it costs less. This structure offers for different people at different position with different networks (like WAN, LAN, internet, and intranet) access to common databases. Especially if some transplatform programming languages like Java use the B/S model is a convenient and highly efficient environment [19]

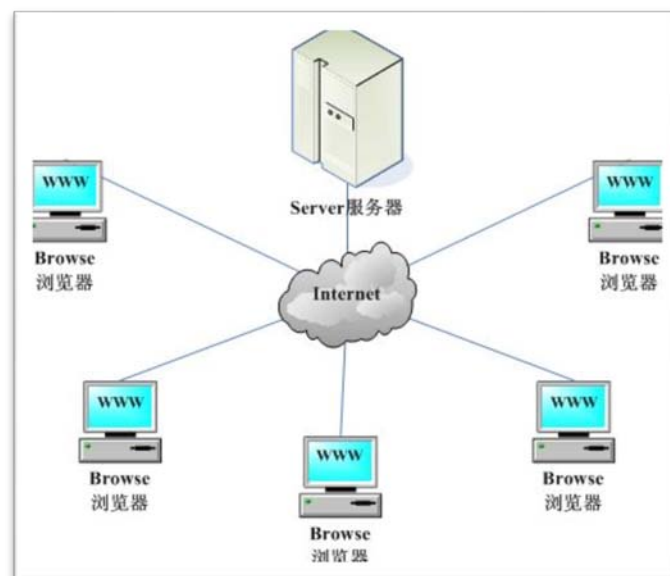


Figure 5.1 Structure of B/S [20]

Recently, software system updates and maintenance has become very frequent. So in the C/S model the updates are typically needed both on the server and on the client side and this can cost large sums of more. Because in C/S mode, each module is close to each other, the change in one module may require updates in others. The basic structure of the C/S is show in Figure 5.2[21]

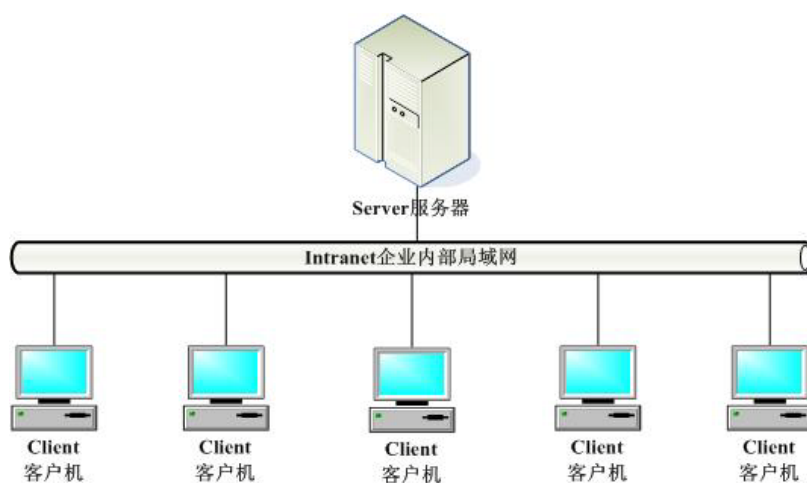


Figure 5.2 Structure of C/S [22]

Comparing with C/S, B/S is easy to update and maintain, because almost all the work is concentrated on the server. Users' side needs not to be updated software. In other words, this way we can reduce remote users' costs. Users need not to install any other software, but browsers can do all the things, no matter what kind of operating systems you are using. Therefore B/S is cheaper than C/S. B/S does not need to take extra cost in spite of increasing the number of users. For these reasons more and more companies are starting to establish they application structure to B/S [21]

5.2 The Tomcat server

Tomcat is a kernel item of Apache Software foundation and Jakarta. The logo of Tomcat is shown in Figure 5.3. It is developed by Apache, Sun and other companies. The latest technologies like JSP and Servlet always operates smoothly on it because of the support from Sun MicroSystem. Tomcat 5.0 supports the lasted version of JSP 2.0 and Servlet 2.4. As Tomcat uses advanced technology, is free and has nice stability, scalability and security features many Java developers love this server. When Tomcat operates, it takes a little system resource. It supports load balancing, mail service and many other functions that developers need.



Figure 5.3 Tomcat logo [23]

Tomcat is a small and lightweight application server. It is developers' first choice for developing programs for middle sized system and no concurrent access systems. Many beginners think that when you complete configuration of an Apache server, you can use it to respond HTML pages requests. In fact, Tomcat is an extension of Apache server, but it is operating independently. So when you are using Tomcat, it works as an Apache independent process. Apache serves HTML pages; in fact Tomcat operates with JSP pages and Servlets. Besides, like IIS、Apache's web server, Tomcat has the function of handling web pages. It is also a container of JSP and Servlet. The lasted Tomcat version is 7 [24].

5.3 Servlet technology

Servlet is a Java program at server fraction. It is independent of platforms and protocols developing dynamic web pages. It acts as a middle layer between a client request (web browser and other HTTP client programs) and a server response (databases or application programs on HTTP server). Servlet locates in inner web server's Java application programs. It is different from traditional Java application programs, because Servlets are loaded by a web server. This web server must naturally support Servlets and Java virtual machine.

Servlet is a small program which operates in web server. And this word "Servlet" is created in Java applet environment the following way.

Server + Applet =Servlet

A web server almost always needs some programs that are based on data stored on a database. These application programs are achieved by using CGI (Common Gateway Interface). The advantage of Servlet is that they are more efficient than CGI programs. Each user is served with a thread in a single program. It means servers do not need to establish a separate process for each user and thereby reduce the system response time.

One Servlet is one class in Java programming language. It is used to extend server's capability. Servlet is very suitable for data processing. The drawback of Servlet is that it does not have Graphical User Interface at Server side. So Sun develop Java Server Page (JSP) technology to cope with this problem. Next part I will introduce JSP technology.[25]

5.4 JSP Technology

JSP is a dynamic webpage standard developed by Sun and other company. JSP is very similar to ASP technology. It inserts Java scriptlet and JSP tags in traditional HTML files. Web application developed by JSP is a cross platform. It can be used with all kinds of operating systems. The operation logic of JSP is shown in Figure 5.4

JSP uses Java language to edit XML, tags and scriptlets in order to encapsulation dynamic web pages' process logic on an HTML page. Web pages also can visit server resource through tags and scriptlets. JSP set apart web logic and web design; it makes web development easy and fast. [26]

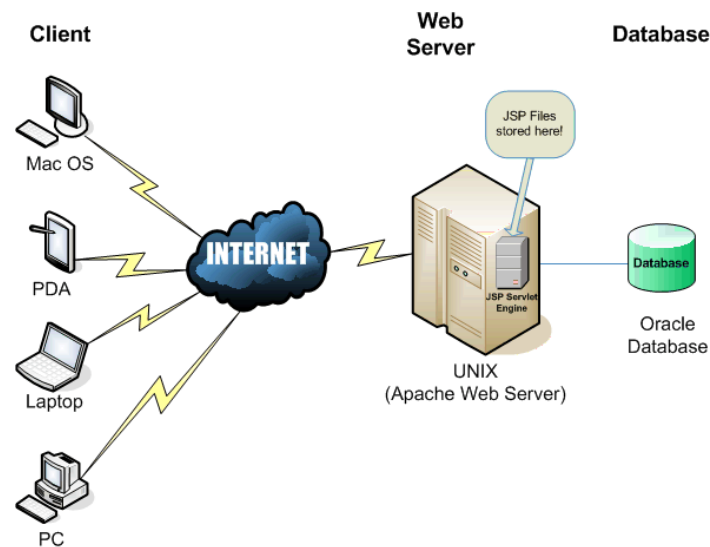


Figure 5.4 JSP [27]

When web server receives a request from JSP pages, it will execute the program segments and then return both JSP files and HTML codes to the client. Inserted Java segments can manipulate a database and so achieving more functions which dynamic webs need. Both JSP and Servlet are executed at server side. It returns a HTML text to client, so at client side, users only need to have a browser.

5.4.1 Advantages of JSP over Competing Technologies

When comparing with ASP or ColdFusion, JSP has a better language for dynamic part. As you can see the JSP structure in Figure 5.5 It is portable to multiple servers and Comparing with pure Servlet, it is more convenient to create HTML and can use standard tools such as Dreamweaver. It has a divided- and-conquer policy. JSP developers still need to know Servlet language. Comparing with client-side JavaScript (in browser), capabilities mostly do not overlap with JSP, but you control server, not client and has a richer language support.[26]

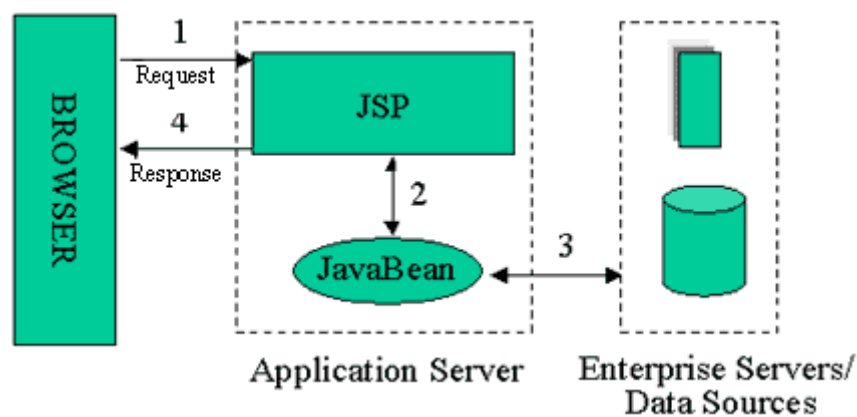


Figure 5.5 JSP structure [28]

5.4.2 Drawbacks of JSP

Like ASP, the biggest advantage of JSP is it's fatally shortage, because of JSP support many operating system, it will increase more complexity into it. It is not an easy task to choose the appropriate servlet engine. There isn't a single organization which conducts independent benchmark testing on servlet engines. [26]

5.5 My Database MySQL

In my project, I choose MySQL as my database (see the logo of MySQL in Figure 5.6) MySQL is a small database system; it was developed by a Swedish Company. In 2008, MySQL was purchased by Sun Microsystems. However, in 2009, Sun was purchased by Oracle which is the world leader in commercial database products. Nobody maintains a positive attitude to the future of MySQL. The reason why I choose MySQL is because it's free and I know how to use it. Many small and medium-sized enterprises like it because it has a small size, high efficiency and above all it is open source software. It is very popular among the internet because of the combination LAMP (Linux + Apache+MySQL+ PHP) [29]



Figure 5.6 MySQL logo[30]

5.5.1 The characteristics of MySQL

MySQL is implemented with C and C++, so this ensures it high degree of portability. It supports the following operating systems' HP-UX、Linux、Mac OS、Novell Netware、OpenBSD、OS/2 Wrap、Solaris and Windows. It provides API for many programming languages such as C、Python、Java、Perl、PHP、Eiffel、Ruby and Tcl. It also supports multi-thread and optimized the algorithm of SQL. [29]

6 IMPLEMENTING A WEB SITE BASE ON J2EE

6.1 Connection

Servlet or JSP, Database and Tomcat are independent from each other. So the first step is to make a connection between them. In my project, I firstly connect Tomcat with my Servlet. In order to connect Tomcat to Servlet, you have to configure Tomcat. You should establish a new folder named it WEB-INF in the catalogue of %TOMCAT_HOME%\webapps. And then create a text file and named it web.xml.

You have to configure web.xml file including one Servlet as follows:

EXAMPLE 6.1

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
version="3.0"
metadata-complete="true">
<display-name>Servlet Examples</display-name>
<description>
Servlet Examples.
</description>
<servlet>
<servlet-name>Hello</servlet-name>
<servlet-class>sb.sb.Hello</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>Hello</servlet-name>
<url-pattern>/han</url-pattern>
</servlet-mapping>
```

The second step is to configure your Servlet. The number of Servlets must be the same as your webpage number; otherwise Tomcat cannot recognize your web site.

My Servlet configuration is shown as follows:

Example 6.2

```
package sb.sb;
import javax.servlet.*;
import java.io.*;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import java.io.IOException;
public class Hello implements Servlet{
public void init(ServletConfig parm1) throws ServletException {
    // TODO: Add your code here
    System.out.println("init it");
}
public ServletConfig getServletConfig() {
    // TODO: Add your code here
    return null;
}
public void service(ServletRequest req, ServletResponse res) throws ServletException,
IOException {
    // TODO: Add your code here
    System.out.println("service it");
    PrintWriter pw=res.getWriter();
    pw.println("hello,world");
}
public String getServletInfo() {
    // TODO: Add your code here
    return "";
}
public void destroy() {
    // TODO: Add your code here
    System.out.println("destory!");
}
}}
```


After these two steps, you can test that your system works fine by insert the following URL to your browsers: `http://localhost:8080/myWebSite/han`. Han means the URL of my webpage. The result of the test page is shown in Figure 6.1

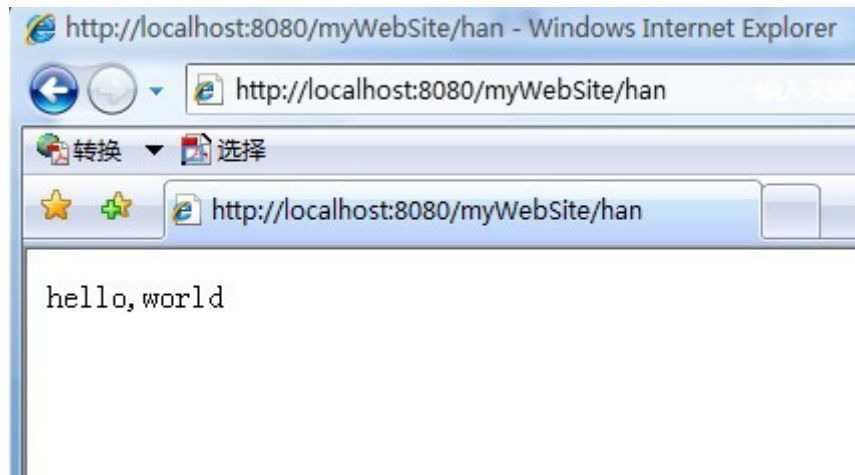


Figure 6.1 the result showing the successful configuration of Servlet and Tomcat

After you completed the connection between Servlet and Tomcat, you can start to configure the connection between MySQL and Servlet. The example 6.3 shows the required Java code.

EXAMPLE 6.3

```
package sb.sb;
import java.sql.*;
public class ConnDB
{
private Connection ct=null;
//this is a database connection
public Connection getConn()
{
    try
    {
```

```

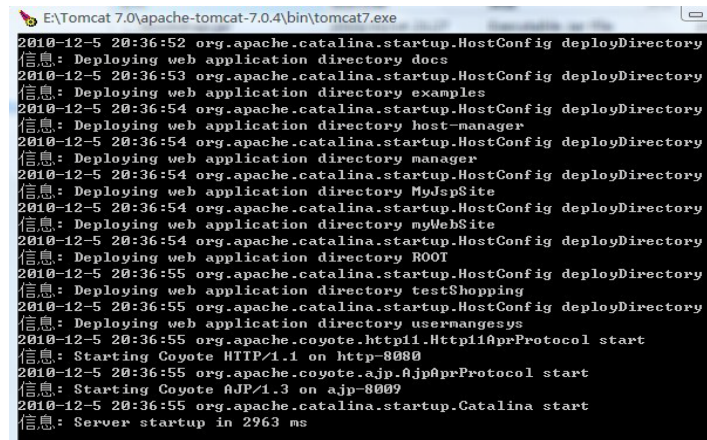
        //connect database
        Class.forName("com.mysql.jdbc.Driver");
        //get connection
ct=DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/handb?user=root&password=12
3");
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
    }
    return ct;
}
}

```

Before you can do that, you have to import the package of `java.sql.*` or otherwise you cannot find out the required method `getConnection()`. `com.mysql.jdbc.Driver` is the constant parameter for Servlet to load a driver. `jdbc:mysql://127.0.0.1:3306/handb?user=root&password=123`. `127.0.0.1` is the IP address on your local computer, `root` is my MySQL's username and `123` is the password. `Handb` is the name of the MySQL package where I have edited some data. At this point, we have finished all the connection steps.

6.2 Turn on the servers

Nest chapter mainly discusses how to implement my website with J2EE. My website is consisted of three parts, webserver, MySQL and Java. In order to running the website you have to turn on the webserver, these issues are explained in this section. You can see how to do it in Figure 6.2



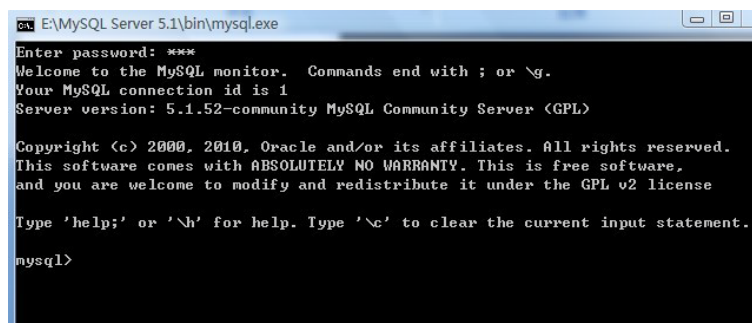
```

E:\Tomcat 7.0\apache-tomcat-7.0.4\bin\tomcat7.exe
2010-12-5 20:36:52 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory docs
2010-12-5 20:36:53 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory examples
2010-12-5 20:36:54 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory host-manager
2010-12-5 20:36:54 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory manager
2010-12-5 20:36:54 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory MyJspSite
2010-12-5 20:36:54 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory myWebSite
2010-12-5 20:36:54 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory ROOT
2010-12-5 20:36:55 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory testShopping
2010-12-5 20:36:55 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory usermangesys
2010-12-5 20:36:55 org.apache.coyote.http11.Http11AprProtocol start
信息: Starting Coyote HTTP/1.1 on http-8080
2010-12-5 20:36:55 org.apache.coyote.ajp.AjpAprProtocol start
信息: Starting Coyote AJP/1.3 on ajp-8009
2010-12-5 20:36:55 org.apache.catalina.startup.Catalina start
信息: Server startup in 2963 ms

```

Figure 6.2 Turn on the webserver

After you switch on the WebServer, you can turn on the Mysql server (in Figure 6.3) which is my database.



```

E:\MySQL Server 5.1\bin\mysql.exe
Enter password: ***
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.52-community MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

Figure 6.3 Turn on the MySQL server

After we turn on the two servers, I should redeploy my project to the webserver. The redeploy Figure in shown in Figure 6.4

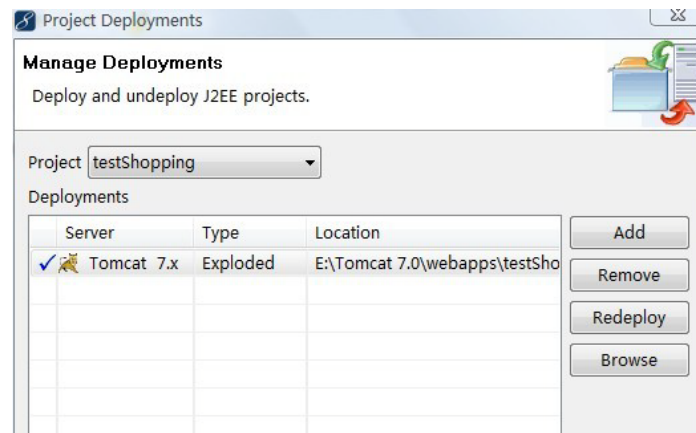


Figure 6.4 Redeploy the project

7 PROGRAMMING A WEB SITE WITH J2EE

7.1 The structure of the web site

You can see the whole logic and stricture of this website in Figure 7.1

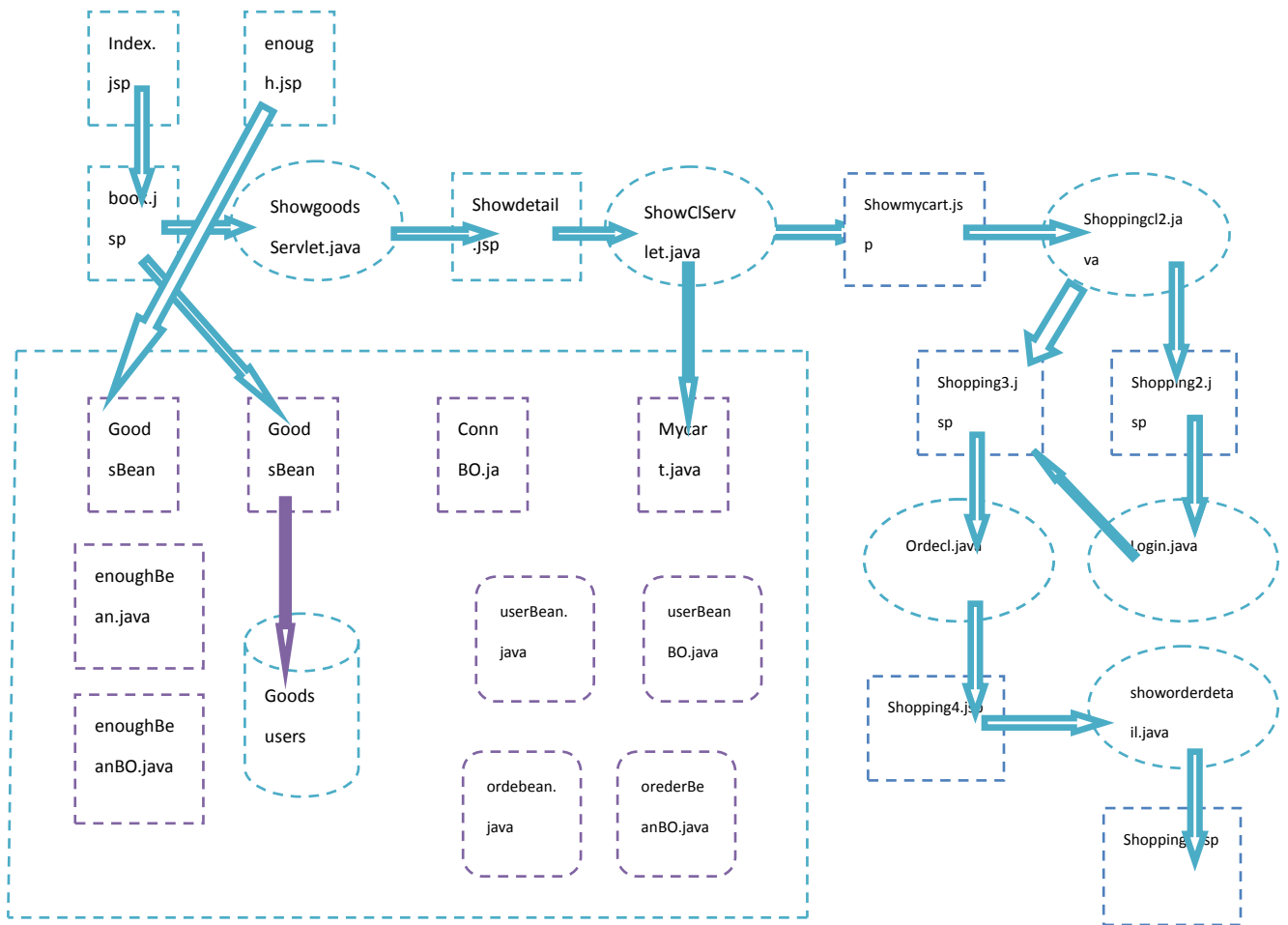


Figure 7.1 Structure of website

7.2 The structure of running a web site

The Figure 7.2 shows the basic structure of my programming. So in order to establish a web site, you have to handle not only the part of J2EE but also cope with a database, HTML and JavaScript.

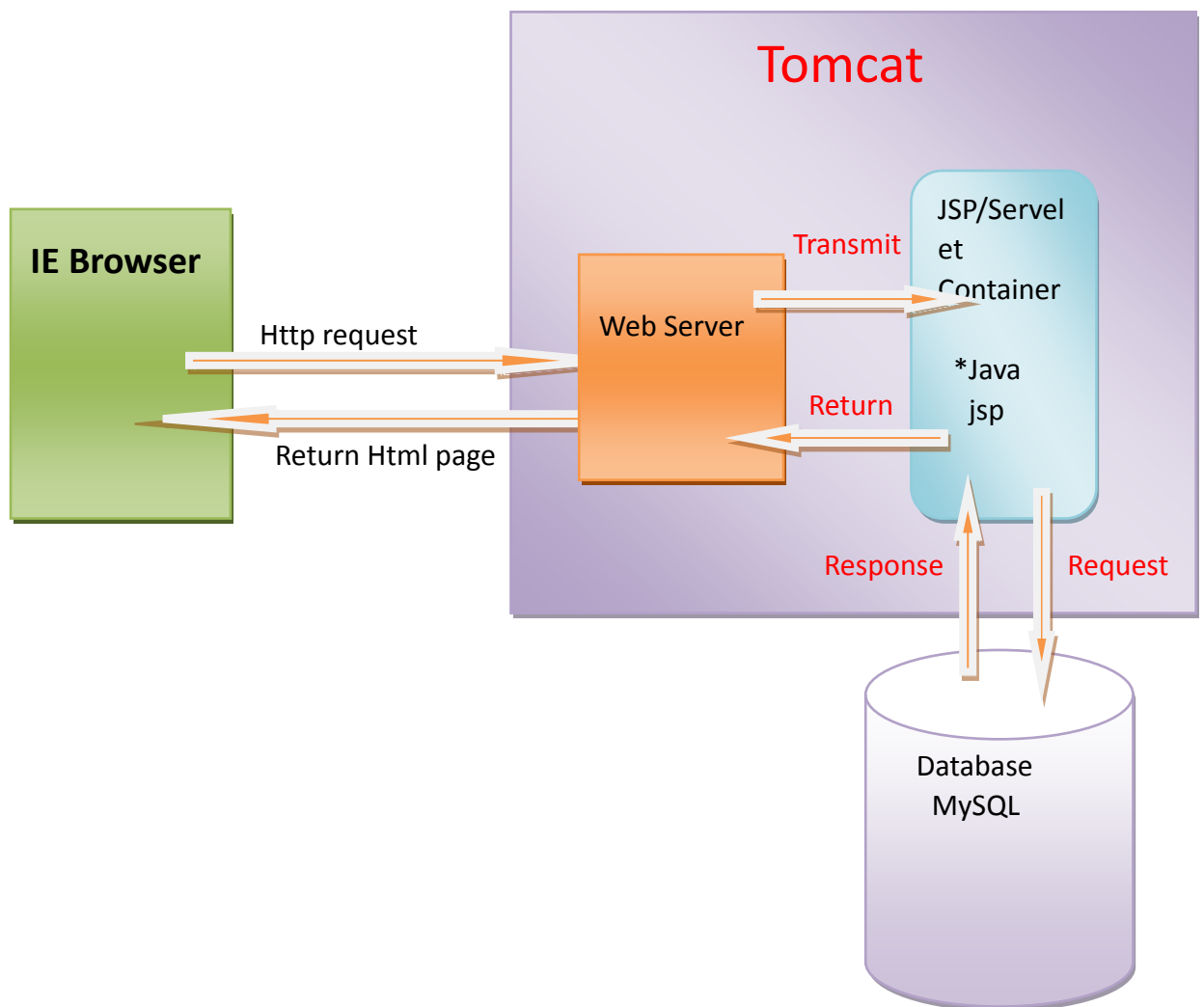


Figure 7.2 Structure of Programming a Web site

7.3 The Login Page

A login page show in Figure 7.3 is a very important page for my web site. It can distinguish normal users from super users. For the security reason, I do a lot of work on the Login page. Verification steps are like this. Server firstly receives the username which you put on the login page and then check it in database to find out the corresponding password and to check whether the password is the same as your input.

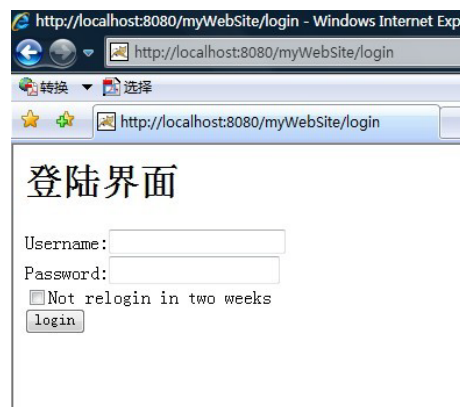


Figure 7.3 Login page

If you put the correct username and password, you will access the welcome page (see Figure 7.4), otherwise you will stay on the login page

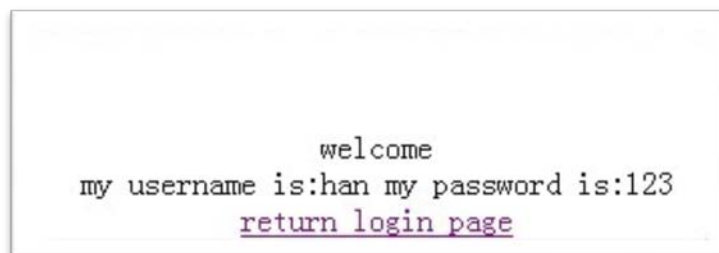


Figure 7.4 Welcome page

7.4 Show visitors IP address and PC name

My website provides a function to show visitors' IP address and his computer's name. This function is very easy to accomplish, because Servlet offers it. The Java code for doing this is show in example 7.1 and the outcome can be seen in Figure 7.5

EXAMPLE 7.1

```
pw.println("Thispageisvisited"+this.getServletContext().getAttribute("visitTimes").toString()+"
times <br>");
pw.println("Your IP address is="+req.getRemoteAddr()+"<br>");
```

```

  1 2 3 4 5 Next
This page is visited 66 times
Your IP address is=127.0.0.1
Your Computer Name is=127.0.0.1
```

Figure 7.5 show IP address and computer name

7.4.1 Show request number

I accomplish the function of show visiting number by doing three steps. First, you have to establish a text file which is used to record the number of visitors in your web site. You can also write the original visited number in the text file. In order to reduce reading and writing work, I put the reading work in the Init() function which will be executed only once and put the writing work in the destroy() function which will be executed after your reload the server or exit the web site.

Reading work in Init() function is shown in example 7.2:

EXAMPLE 7.2

```

public void init()
{
    try {
        FileReader f=new FileReader("f:\\myCounter.txt");
        BufferedReader br=new BufferedReader(f);
        String numVal=br.readLine();
        br.close();
        int times=Integer.parseInt(numVal);
        this.getContext().setAttribute("visitTimes",numVal);
        System.out.println("init is used");
    }
    catch (Exception ex) {
        ex.printStackTrace();
    }
}

```

Writing work in destroy() function is shown in example 7.3:

EXAMPLE 7.3

```

public void destroy()
{
    try {
        FileWriter fw=new FileWriter("f:\\myCounter.txt");
        BufferedWriter bw=new BufferedWriter(fw);
        bw.write(this.getContext().getAttribute("visitTimes").toString());
        bw.close();
        System.out.println("destroy is used");
    }
    catch (Exception ex)
    {
        ex.printStackTrace();
    }
}

```

After you have completed the functions of writing and reading, you can create a variable which is used to record the increasing number in Servlet Context. Only after the users exit the web site, the value of the variable will be written in the text file. Next time, when someone visits your web site, his computer will load the value in the text file first. You can see the visited number changes in Figure 7.6 In this way, computer avoid the work of frequently operate the text file and gives it to the memory.

<hr/> 1 2 3 4 5 Next This page is visited 67 times Your IP address is=127.0.0.1 Your Computer Name is=127.0.0.1	1 2 3 4 5 Next This page is visited 68 times Your IP address is=127.0.0.1 Your Computer Name is=127.0.0.1
--	--

Figure 7.6 The visited number changes

7.5 User Management System

In my web site, each user has the following information: User ID, Password, User Name, Email address and Grade. You cannot modify the User ID and User Name. As a administrator, I can delete the user from my database and modify the password information as well as email address and grade. In the main page (you can see in Figure 7.7), you can choose users managements to accomplish these functions.



Figure 7.7 The main page

You can click the User Management to access the modify page (you can see in Figure 7.8).

ID	name	password	Email	Grade	Modify User	Cancel User
3	admin3	123	admin3@live.cn	3	Modify User	Cancel User
4	admin4	123	admin@live.cn	4	Modify User	Cancel User
8	test4	test4	test4@live.cn	4	Modify User	Cancel User

[1](#) [2](#) [3](#) [4](#) [5](#) [Next](#)
 This page is visited 160 times

Figure 7.8 The modify page

You can choose anyone to modify. In order to accomplish the modification, you have to create a class to help you solve it. First, you have to create an update page and send all the values from modify page to the function of update. The code for transmit values is shown in example 7.4:

Example 7.4

```
pw.println("<td><a href=Updata?uid="+ub.getUserId()+"&uName="+ub.getUserName()+"&uPass="+ub.getPassword()+"&uEmail="+ub.getEmail()+"&uGrade="+ub.getGrade()+">Modify User</a></td>");
```

The second step is to create an update page to receive all the values from the modify page and provide a function of editing values. The update page is like this:

Editing Users

ID	<input type="text" value="4"/>
Name	<input type="text" value="admin4"/>
Password	<input type="text" value="admin4"/>
Email	<input type="text" value="admin@live.cn"/>
Grade	<input type="text" value="4"/>
<input type="button" value="edit"/>	

Figure 7.9 The update page

The code of the update page is shown in example 7.5:

Example 7.5

```

package sb.sb;
import java.io.*;
import javax.servlet.http.*;
public class Updata extends HttpServlet{
public void doGet(HttpServletRequest req,HttpServletResponse res)
{try{ res.setContentType("text/html;charset=GBK");
PrintWriter pw=res.getWriter();
pw.println("<html>");
pw.println("<bodybgcolor=#FDCE5F background=images/untitled.bmp><hr><center>");
pw.println("<h1>Editing Users</h1>");
pw.println("<form action=UpdataCl>");
pw.println("<table border=1>");
pw.println("<tr><td>ID</td><td><input        readonly        name=uid        type=text
value="+req.getParameter("uid")+ "></td></tr>");
pw.println("<tr><td>Name</td><td><input                readonly                type=text
value="+req.getParameter("uName")+ "></td></tr>");
pw.println("<tr><td>Password</td><td><input                name=newPasswd                type=text
value="+req.getParameter("uPass")+ "></td></tr>");
pw.println("<tr><td>Email</td><td><input                name=newEmail                type=text
value="+req.getParameter("uEmail")+ "></td></tr>");
pw.println("<tr><td>Grade</td><td><input                name=newgrade                type=text
value="+req.getParameter("uGrade")+ "></td></tr>");
pw.println("<tr><td colspan=2><input type=submit value=edit user></td></tr>");
pw.println("</table></form>"); pw.println("</center><hr>");pw.println("</body>");
pw.println("</html>");}catch (Exception ex)
{ex.printStackTrace();
}}public void doPost(HttpServletRequest req,HttpServletResponse res)
{
this.doGet(req,res);
}}
```

When you click edit, all the values you fill in the screen will be transmitted to the Servlet of updatecl. The functions of updatecl are connecting to database and transmit the values to database. When you operate successfully, you will jump to successful page, otherwise you will jump to the error page. The codes of the updatecl are shown in example 7.6:

Example 7.6

```

package sb.sb;
import java.io.*;
import javax.servlet.http.*;
public class UpdataCl extends HttpServlet{
    public void doGet(HttpServletRequest req,HttpServletResponse res)
    {
try{res.setContentType("text/html;charset=GBK");
UserBeanTest ubc=new UserBeanTest();
    String id=req.getParameter("userId");
if(ubc.updataUser(req.getParameter("uld"),req.getParameter("newEmail"),req.getPa
rameter("newPasswd"),req.getParameter("newgrade")))
{res.sendRedirect("Ok");}else{
    res.sendRedirect("Err");
}}catch (Exception ex)
    {ex.printStackTrace();
    }}
public void doPost(HttpServletRequest req,HttpServletResponse res)
{ this.doGet(req,res);}}

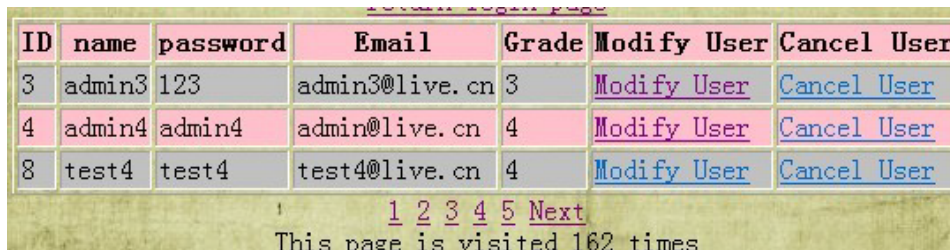
```

You can see results in Figure 7.10 after you click edit.



Figure 7.10 The congregations' page

You can see the new modify page after I updated the new data in Figure 7.11. The password of admin4 is change from “123” to “admin4”.



ID	name	password	Email	Grade	Modify User	Cancel User
3	admin3	123	admin3@live.cn	3	Modify User	Cancel User
4	admin4	admin4	admin@live.cn	4	Modify User	Cancel User
8	test4	test4	test4@live.cn	4	Modify User	Cancel User

1 2 3 4 5 Next
This page is visited 162 times

Figure 7.11 Modify page after editing

7.5.1 Delete Users

As you can see from Figure 7.11, the modify page also provide the function of removing users. When you click Cancel User, there is a popup window out as you can see in Figure 7.12 The content of that window is “Are sure to Cancel this users?”

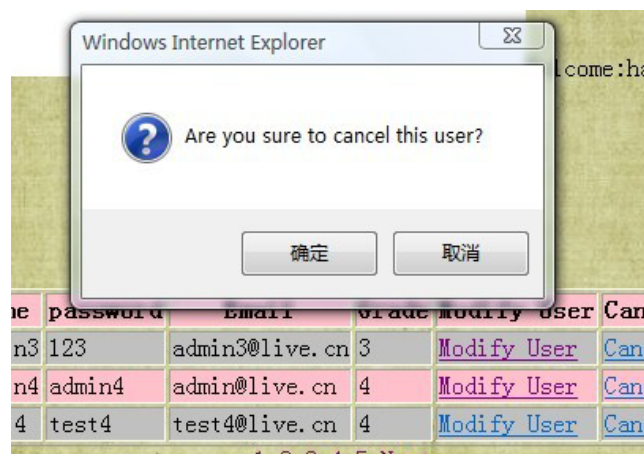


Figure 7.12 The notice window

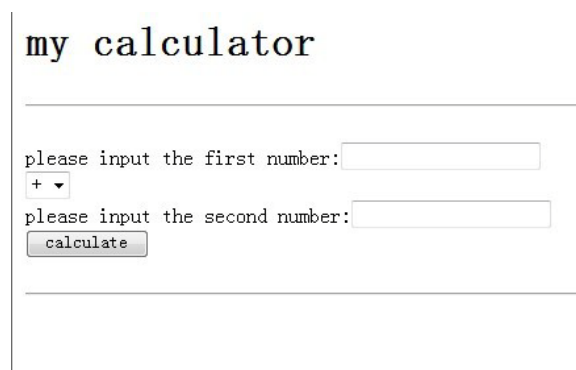
If you click yes, you will access to the conjurations page or error page. It depends on whether the user existed or not. The window function is completed by typing these codes shown in exmample 7.7:

Example 7.7

```
onclick=\"return window.confirm('Are you sure to cancel this user?');
```

7.5.2 Calculated Technology

I create a function of calculating which I will use later. In order to accomplish the function, I develop two web pages and I will put these two pages together later. The first page provide user to input two numbers and submit it. The web page Figure is shown in Figure 7.13:



my calculator

please input the first number:

+ ▾

please input the second number:

Figure 7.13 The calculate page

It is easy to calculate when you input numbers, but as a developer you have to consider about when user input the types of null and character. So I write the code in Appendix-1:

7.6 Web Page Design

In order to achieving high efficiency and better visual effects, I use Macromedia Dreamweaver as my developing tools. And I design a framework of index page shown in Figure 7.14:

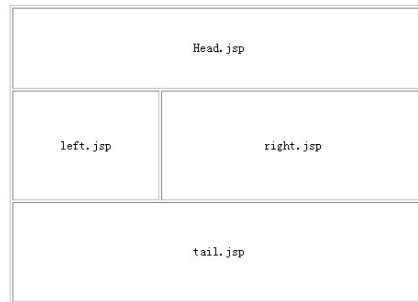


Figure 7.14 Index page framework

The next 4 steps are to be complete, head.jsp, tail.jsp, left.jsp and right.jsp. Head and tail are the easy parts. Left side is a little bit difficult, because I want to insert a flash in this part and add a rank. The most difficult part is the right side; I introduce it at the end of these 4 steps.

I use Dreamweaver draw the framework of head.jsp as you can see in Figure 6.6.2:



Figure 7.15 Head Page Framework

After I add color, picture and text in this framework, I copy the source code to head.jsp and put the head.jsp in the Index.jsp.

The framework of tail.jsp shown in Figure 6.6.3



Figure 7.16 Tail Page Framework

I put some information of myself in this framework, my email address, address, copyright and etc. I also copy the source code from Dreamweaver to the tail.jsp and put the tail.jsp to the index.jsp. So the work of Dreamweaver is to create table and CSS.

The next part is to create the framework of left.jsp is shown in Figure7.17:

Figure 7.17 Left Page Framework

I establish a dynamic rank in this framework and the data of the rank is from the database. In the right side, I create a dynamic click which is decided the rank. In the top of this framework I put some picture in it and add a flash. In order to making flash runs smoothly you have to download focus.swf and creating banner.js. Banner.js is a JavaScript and I use Banner.js to import focus.swf.

The framework of right side is shown in 7.18:

Figure 7.18 Right Page Framework

In the right side, there are some books about happiness; all the books are come from my database. This framework include the price of the books and a shortly introduction of the book. At the top of this part, there is a column which is used to

help people solve their problems. Visitor can fill in his problem and click go, the content which the visitor submit will search in my database and my database will give the best answers.

In the right part you can delete or cancel items from the database; all the items are come from database. At the bottom of this part, you can see the numbers this is paging function. And I use some code to complete the function as you can see in exmaple 7.8:

Example 7.8

```
<%
    int time=0;
    for(int i=0;i<2;i++)
    {
        //print two row 3 item (sometime less than 3)
        %>
        <tr>
        <%
            for(int j=0;j<3;j++)
            {
                GoodsBean gb=new GoodsBean();
                if(time>=al.size())
                {
                    gb.setGoodId(0);
                    gb.setshortint("None");
                    gb.setPhoto("none.jpg");
                    gb.setGoodPrice(0.0f);
                    gb.setGoodName("none");
                }else{
                    gb=(GoodsBean)al.get(time);
                    time++;}
                %>
```

Shortly explain this source code, it means the computer display 6 items at one page.

Each page has tow row, every row display 3 items.

The source codes of paging function are shown in example 7.9:

Example 7.9

```
<td colspan="3" align="center">
  <%
    for(int i=1;i<=pageCount;i++)
    {
      %>
      <a href="ShowGoodsServlet?type=fenye&pageNow=<%=i %>"><%=i %></a>
      <%
    }
  %>
```

This is core codes of paging function. It means when the page has 6 items, the database will put the items to another page. There is a super link to the "ShowGoodsServlet" which is helping us skim to another page. I will display all the "showGoodsServlet at last.

I also give each item a short description at another JSP page and I named it showgoodsdetail.jsp. The framework of this JSP page as you can see in Figure 7.19:

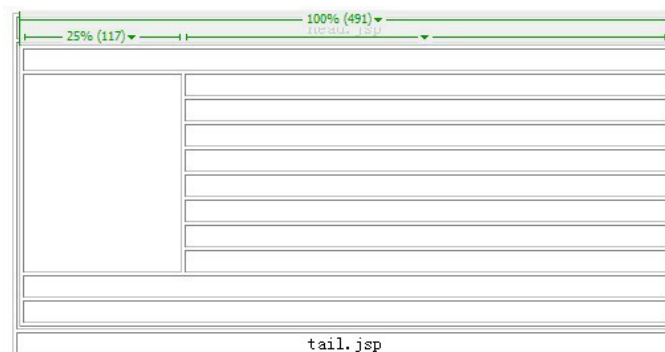


Figure 7.19 The Framework of showgoodsdetail

I put the item's photo in the big picture frame at the left side. In the right side I add some information of this item. It author, price, short introduction.

In showdetail.jsp there are two buttons below the picture; one is “buy” the other is “return to the back”. When visitors click “return to the back” the webpage skip to the index.jsp. If visitors click “buy” the webpage not skip to showMycart.jsp immediately which is used to display all the information that user want to buy including price, amount and goods ID. The showdetail.jsp conveys all the information to ShoppingCIServlet. ShoppingCIServlet cope with these data by using MycartBO which is used to handle logic and return Boolean value. ShoppingCIServlet check the value and then decide which page should skip.

The form of showMycart.jsp is shown in Figure 7.20:

The screenshot shows a web page titled 'Shopping Steps' with a navigation bar containing: 'search your shopping cart' (with a magnifying glass icon), 'User login or register' (with a person icon), 'Fill in shopping information' (with a crescent moon icon), and 'Complete Shopping' (with a smiley face icon). Below the navigation bar is a table with the following data:

ID	Name	Price	Amount
1	2	32	<input type="text"/> delete search

Below the table are two buttons: 'Delete all items' and 'Modify Number'. At the bottom left, there is a text box: 'The totally items price are \$100,click here to countinue purchasing.' At the bottom right, there is a 'Next' button with a right-pointing arrow. The page is labeled 'head.jsp' at the top and 'tail.jsp' at the bottom.

Figure 7.20The Framework of showMycart.jsp

ShowMycart.jsp is one of the complex parts in this website design. It includes goods ID, name and price. You also can delete items and check items detail and modify number. ID, Name and Price are come from database. When you click search, it will give a type value to ShoppingCIServlet. ShoppingCIServlet based on the contents of the type to decide which command should be executed. When you click delete, it will also give a type to ShoppingCIServlet. The work principle of button deletes all items and modifies number are the same as delete and search. At the lower left, there is a total price of all the items. This function also used ShoppingCIServlet.

After you click next, you will enter to shoppingCl2.jsp and this page is used to verification users.

The form of the shoppingCl2.jsp is shown in Figure 7.21:

head.jsp	
USER LOGIN	
username:	<input type="text"/>
password:	<input type="password"/>
<input type="button" value="login"/> <input type="button" value="register"/>	
tail.jsp	

Figure 7.21 The Form of shoppingCl2.jsp

Visitors put their username and password here. This information will be sent to LoginCl to hand. Login page is a very important page for my web site. It can distinguish normal users from super users. For the security issue, I do a lot of work on the Login page. Verification steps are like this. Server firstly receives the username which you put on the login page and then check it in database to find out the reflected password and to check whether the password is the same as your input. If it is correctly, you can enter to shopping3.jsp or you will stay in the login pages.

The form of the shopping3.jsp is shown in Figure 7.22:

head.jsp			
picture			
Buyer Information			
	Username	<input type="text"/>	
	*Turname	<input type="text"/>	
	*Address	<input type="text"/>	
	*Telephone	<input type="text"/>	
	*Email:	<input type="text"/>	
	*Postcode:	<input type="text"/>	
<input type="button" value="Complete Booking"/> <input type="button" value="70% (597) or Information"/>			
My Shopping Cart			
Id	Goods Name	Price	Amount
picture			
tail.jsp			

Figure 7.22 The Form of shopping3.jsp

When you login successful, this page will display both users' information and shopping cart's information. All these information are coming from the database. When you click next you will enter to shopping4.jsp.

The form of shopping4 is shown in Figure 7.23:

head. jsp								
Booking detail								
Booking number	receiver	address	Postcode	phone	total price	Username	Email	More detail
tail. jsp								

Figure 7.23 The Form of shopping4.jsp

To operate this page is one of difficult work in this project. Because not all the information is coming from the database some are coming from the database and some are from the shopping cart. And shopping cart are dynamic change, it depends on the users. In order to accomplish this function I create two forms, one is static and the other is from the shopping cart and then pulls out all the information put it in to another database and draws the need information from it.

The source codes of multi-table search are showin in example 7.10

Example 7.10

```

        ArrayList al=mbo.showMyCart();
        Statement sm=ct.createStatement();
        for(int i=0; i<al.size();i++)
        {
            GoodsBean gb=(GoodsBean)al.get(i);
            sm.addBatch("insertintoorderdetailvalues('"+mbo.getGoodsNumByld(gb.getGoodId()+"'"+
            ,""+orderId+"", ""+gb.getGoodId()+"'");
        }

        sm.executeBatch();
        Stringsql="selectorderId,trueName,address,postcode,phone,totalPrice,username,email from

```

```
users,orders where orderId='"+orderId+"' and users.userId=(select orders.userId from orders
where orderId='"+orderId+"");
```

```
    ps=ct.prepareStatement(sql);
    rs=ps.executeQuery();
    if(rs.next())
    {
        //put rs to orderInfoBean
        oib.setOrderId(rs.getInt(1));
        oib.setTruename(rs.getString(2));
        oib.setAddress(rs.getString(3));
        oib.setPostcode(rs.getString(4));
        oib.setPhone(rs.getString(5));
        oib.setTotalPrice(rs.getFloat(6));
        oib.setUsername(rs.getString(7));
        oib.setEmail(rs.getString(8));
    }
}
```

At the top of right2 page, there is a text field which is used for visitors to submit some bad mood. As you can the picture show in Figure 7.24:



Figure 7.24 Enough text filed

After user click the button “go”, the content will be submit to the EnoughCl.java to handle. The EnoughCl.java use enoughBeanBo.java to check the content from the database. If enoughBeanBo.java can find the content in the database, the enoughBeanBO.java pull out the reflected content and put it into enoughBean.java which is used to save these information and then return a right value to the EnoughCl.java. EnoughCl.java checks the value to decide which page should be skipping.

In other word, I should to create a new webpage to show the result to the visitors.

The head.jsp and tail.jsp are same as before. The middle part is different. The form of the enough pages is shown in Figure 7.25:

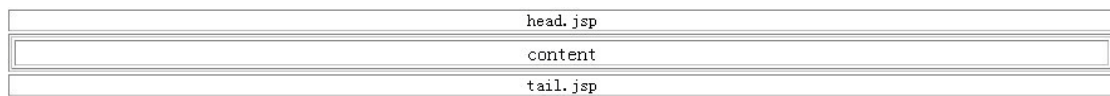


Figure 7.25 The framework of enough page

7.7 The result of the web site

Then click <http://localhost:8080/testShopping/> in the browser. You can see my website in Figure 7.26:



Figure 7.26 The index page

You can put the troubling information in the line after “I am enough” and click the button go. You will receive a method to solve your worries.

When you click books, it will enter to the book page which is the highlights of my project as you can see the book page in Figure 7.27

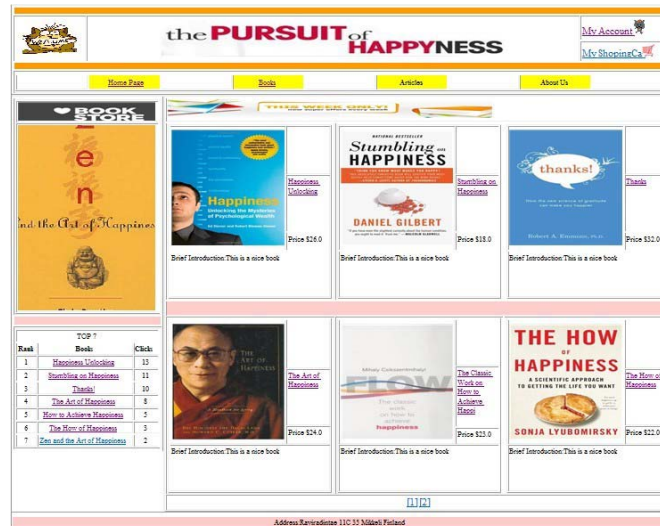


Figure 7.27 The Book page

There are all many books displaying on this page and all these books are base on happiness or how to pursuit happiness. You can click the name of any of these books for more detail information (you can see in Figure 7.28). For example I click the first one which is happiness unlocking.

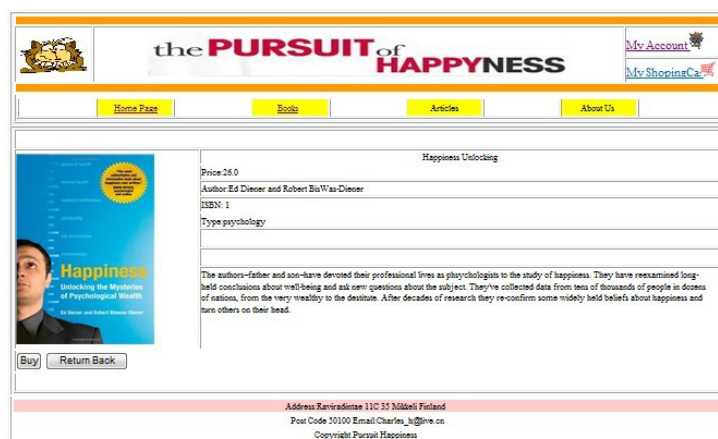


Figure 7.28 The book detail page

I want to emphasize the relation of the book page and book detail page are not one to one correspondence. The book detail is coming from the ShowGoodServlet.java.

When you click the button return back, the page skips to the book page. If you click the button buy the page will skip to the ShowMycart.jsp you can see the page in Figure 7.29



Figure 7.29 The ShowMycart page

There are multiple choices in this page, you can click delete to delete the item; click search for my detail about the book; modify the number of the book or click "here" at the lower left corner return to the book page to buy more books. If you confirm all the booking information you can click next enter to the Login.jsp (you can see it in Figure 7.30)

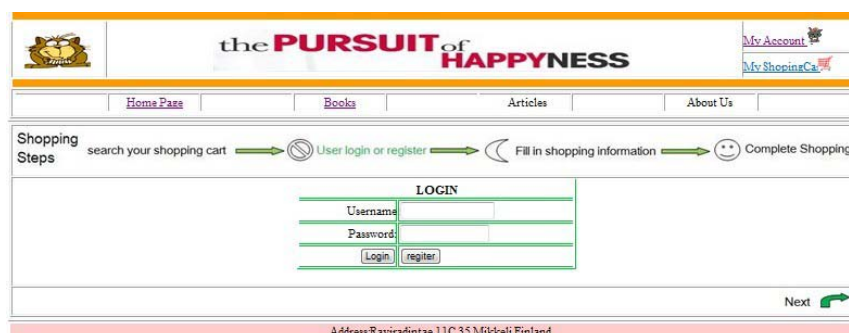


Figure 7.30 Login Page

You can put your user name and password in this page, if the password or the username is correctly, this page would skip to shopping3.jsp (you can see in Figure 7.31) otherwise you would stay in this page.

the PURSUIT of HAPPYNESS

My Account
My ShoppingCar

Home Page Books Articles About Us

Shopping Steps search your shopping cart → User login or register → Fill in shopping information → Complete Shopping

Buyer Information

Username han

*Username hanyoujia

*Address 50100

*Telephone 0417247598

*Email charles_h@live.cn

*Postcode Raviradantie 11c

Complete Booking Modify User Information

My Shopping Cart

Id	Goods Name	Price	Amount
1	Happiness Unlocking	26.0	1

You have been choose 126.0 goods

Return modify shopping cart

Back Next

Address: Raviradantie 11C 35 Mikkelin Finland
Post Code: 50100 Email: Charles_h@live.cn
Copyright Pursuit Happiness

Figure 7.31 Shopping3 Page

If everything is correctly, you can click Next to confirm your booking at the same time the page will skip to shopping4.jsp (you can see in Figure 7.32)

the PURSUIT of HAPPYNESS

My Account
My ShoppingCar

Home Page Books Articles About Us

Shopping Steps search your shopping cart → User login or register → Fill in shopping information → Complete Shopping

Booking detail

Booking number	receiver	address	Postcode	phone	total price	Username	Email	More detail
76	hanyoujia	Raviradantie 11c	50100	0417247598	\$26.0	han	charles_h@live.cn	detail

Your booking is complete, our server will send you a confirm Email later

Address: Raviradantie 11C 35 Mikkelin Finland
Post Code: 50100 Email: Charles_h@live.cn
Copyright Pursuit Happiness

Figure 7.32 shopping4 page

8 CONCLUSION

The aim of my study is to get familiar with J2EE technology on JSP and Servlet and utilize it to establish a website including the function of searching, online shopping and login. Finally I successfully developed the website. To make a conclusion, I will discuss the challenges and problems I met and possible future work.

8.1 Challenges and Solutions

The most difficult part of my work is the shopping cart system. Because all the items and information are dynamic, it is easy for administrator to modify the items and users information but it is hard for web developers to develop. Especially the last shopping carts used many tables in the database but itself it does not exist in the database. The algorithms are hard. Finally, I solve the problem through multi-search in the database.

The second difficult part is the connection problem between Java and MySQL. Because every time when the logic handles data, it has to use the database. It will waste a lot of time in a big website with countless information. To make my website running with high efficiency, I add some source code as I wrote before.

The third problem is the security problem in the login page. I come out the problem by check the username in the database and return the correspondence password to LoginCl.java. LoginCl.java checks the password whether or not is same as the user's input.

8.2 Possible future work

This web site needs some graphic designer to make it looks more colorful. And some web games can be added to the website. After that you can publish the website using Tomcat and buy a DNS for this website.

In my opinion the biggest challenge in the future work is how to establish a connection between my website and Bank System and provide several payment methods for consumers to choose. To overcome this problem, I think the first step is to negotiate with banks, and other internet payment companies like PayPal

Reference

[1] < URL

<http://www.time.com/time/2005/happiness/graphics/poll.html>>

[2]<URL

[http://en.wikipedia.org/wiki/Python_\(programming_language\)](http://en.wikipedia.org/wiki/Python_(programming_language))>

[3]<URL

<http://www.javaeye.com/wiki/topic/395600>>

[4]<URL

<http://en.wikipedia.org/wiki/PhpWiki>>

[5]<URL

<http://image.baidu.com/i?ct=503316480&z=0&tn=baiduimagedetail&word=php&in=5282&cl=2&lm=-1&pn=0&rn=1&di=25236274260&ln=1&fr=&ic=0&s=&se=&sme=1&tab=&width=&height=&face=0&fb=0>>

[6]<URL

http://en.wikipedia.org/wiki/.NET_Compact_Framework>

[7]<URL

<http://baike.baidu.com/view/21087.htm>>

[8]<URL

http://d1.wapttw.cn/wapttw/201002/20100206/wapttw.cn_06170707_ozCYO.gif>

[9]<URL

[http://en.wikipedia.org/wiki/Python_\(programming_language\)](http://en.wikipedia.org/wiki/Python_(programming_language))>

[10]<URL

<http://baike.baidu.com/view/1507.htm> >

[11]<URL

<http://www.scp.co.il/images/newsletter/candleFigure1.jpg>>

[12]<URL

<http://en.wikipedia.org/wiki/JDK>>

[13]<URL

<http://en.wikipedia.org/wiki/MyEclipse>>

[14]<URL

<http://www.aease.org/blog/wp-content/uploads/2010/04/MyEclipse.jpg> >

[15]<URL

<http://en.wikipedia.org/wiki/Happiness>>

[16]<URL

<http://upload.wikimedia.org/wikipedia/commons/thumb/7/79/Face-smile.svg/200px-Face-smile.svg.png>>

[17]<URL

<http://www.time.com/time/magazine/article/0,9171,1015832,00.html>>

[18]<URL

<http://www.wmmenglish.com/Article/web01/test04/2008-10-31/8760.html>>

[19]<URL

<http://baike.baidu.com/view/8039.htm>>

[20]<URL

<http://www.webcargo.com.cn/webcargocn/news/eWebEditor/UploadFile/2005101121616929.jpg>>

[21]<URL

<http://baike.baidu.com/view/45170.htm>>

[22]<URL

<http://www.webcargo.com.cn/webcargocn/news/eWebEditor/UploadFile/2005101115250880.jpg> >

[23]<URL

<http://baike.baidu.com/image/d0526df0d88eda87a40f5291>>

[24]<URL

http://en.wikipedia.org/wiki/Apache_Tomcat>

[25]<URL

<http://en.wikipedia.org/wiki/Servlet>>

[26]<URL

http://en.wikipedia.org/wiki/JavaServer_Pages>

[27]<URL

<http://baike.baidu.com/image/54baacfb624a6c1b4f4aeab2>>

[28]<URL

<http://www.computernews.com.cn/techimages/2004101401/1.gif>>

[29]<URL

<http://en.wikipedia.org/wiki/MySQL>>

[30]<URL

<http://imgsrc.baidu.com/baike/pic/item/d048adde44216b61cdbf1a7a.jpg>>

APPENDICES

```

<%@page contentType="text/html;charset=gb2312" %>
<html>
<h1>my calculator</h1><hr>
<head>
<!-- add javascript code aviod user porvide null-->
<script language="javascript">
  <!--
    function checkNum()
    {
      if(form1.num1.value=="")
      { window.alert("num1 cannot be null");
        return false; }
      if(form1.num2.value=="")
      { window.alert("num2 cannot be null");
        return false;}
      if(Math.round(form1.num1.value)!=form1.num1.value)
      { window.alert("num1 is not a number");
        return false;}
      if(Math.round(form1.num2.value)!=form1.num2.value)
      {window.alert("num2 is not a number");
        return false; }    }
  -->
</script>
</head>
<body>
<form name="form1" action="myResult.jsp">
please input the first number:<input type="text" name="num1"><br>
<select name="flag">
<option value=+>+</option>
<option value=->-</option>
<option value=*>*</option>
<option value=/>/</option>
</select><br>
please input the second number:<input type="text" name="num2"><br>
<input type="submit" value="calculate" onclick="return checkNum();">
</form>
<hr>
</body>
</html>

```

Appendix A-1

ConnDB.java

```
package com.han.model;
```

```
import java.sql.*;
```

```
public class ConnDB {
```

```
    private Connection ct=null;
```

```
    public Connection getConn()
```

```
    {
```

```
        try{
```

```
            Class.forName("com.mysql.jdbc.Driver");
```

```
            ct=DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/handb?user=root&password=123");
```

```
        }
```

```
        catch(Exception e)
```

```
        {
```

```
            e.printStackTrace();
```

```
        }
```

```
        return ct;
```

```
    }
```

```
}
```

Appendix A-2

EnoughBean.java

package com.han.model;**public class** enoughBean

{

private int enoughId; **private** String enoughIntro;**public int** getEnoughId() { **return** enoughId;

}

public void setEnoughId(**int** enoughId) { **this**.enoughId = enoughId;

}

public String getEnoughIntro() { **return** enoughIntro;

}

public void setEnoughIntro(String enoughIntro) { **this**.enoughIntro = enoughIntro;

}

}

Appendix A-3

```

EnoughBeanBO
package com.han.model;
import java.sql.*;
import java.util.*;
public class enoughBeanBO {
    private ResultSet rs=null;
    private Connection ct=null;
    private PreparedStatement ps=null;
    public enoughBean getEnoughBean(String content)
    {
        enoughBean eb=new enoughBean();
        try{
            ct=new ConnDB().getConn();
            ps=ct.prepareStatement("select * from enough where enoughIntro like
'%" + content + "%'");
            rs=ps.executeQuery();
            if(rs.next()){
                eb.setEnoughId(rs.getInt(1));
                eb.setEnoughIntro(rs.getString(2));
                System.out.println(eb.getEnoughId());
            }
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
        finally{
            this.close();
        }
        return eb;}
    public void close()
    {try{if(rs!=null){
        rs.close();
        rs=null;
    }f(ps!=null){
        ps.close();
        ps=null;}
        if(!ct.isClosed()){
            ct.close(); }} catch(Exception e){ e.printStackTrace(); }}

```

Appendix A-4

GoodSBean

```
package com.han.model;
public class GoodsBean {
    private int goodId;
    private String goodName;
    private String goodIntro;
    private float goodPrice;
    private int goodNum;
    private String Author;
    private String photo;
    private String tpye;
    private String shortint;
    public long getGoodId() {
        return goodId;
    }
    public void setGoodId(int goodId) {
        this.goodId = goodId;
    }
    public String getGoodName() {
        return goodName;
    }
    public void setGoodName(String goodName) {
        this.goodName = goodName;
    }
    public String getGoodIntro() {
        return goodIntro;
    }
    public void setGoodIntro(String goodIntro) {
        this.goodIntro = goodIntro;
    }
    public float getGoodPrice() {
        return goodPrice;
    }
    public void setGoodPrice(float goodPrice) {
        this.goodPrice = goodPrice;
    }
    public int getGoodNum() {
        return goodNum;
    }
}
```

```
public void setGoodNum(int goodNum) {  
    this.goodNum = goodNum;  
}  
public String getAuthor() {  
    return Author;  
}  
public void setAuthor(String Author) {  
    this.Author = Author;  
}  
public String getPhoto() {  
    return photo;  
}  
public void setPhoto(String photo) {  
    this.photo = photo;  
}  
public String getTpye() {  
    return tpye;  
}  
public void setTpye(String tpye) {  
    this.tpye = tpye;  
}  
  
public String getshortint() {  
    return shortint;  
}  
public void setshortint(String shortint) {  
    this.shortint = shortint;  
}  
}
```

Appendix A-5

GoodBeanBO

```
package com.han.model;
import java.sql.*;
import java.util.ArrayList;
public class GoodsBeanBO {
    private ResultSet rs=null;
    private Connection ct=null;
    private PreparedStatement ps=null;
    int pageCount=0;
    public int getPageCount(int pageSize)
    {
        int rowCount=0;
        try{
            ct=new ConnDB().getConn();
            ps=ct.prepareStatement("select count(*) from goods");
            rs=ps.executeQuery();
            if(rs.next())
            {
                rowCount=rs.getInt(1);
            }
            if(rowCount%pageSize==0)
            {
                pageCount=rowCount/pageSize;
            }else
            {
                pageCount=rowCount/pageSize+1;
            }
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }finally
        {
            this.close();
        }
        return pageCount;
    }
    public ArrayList<GoodsBean> getGoodsByPage(int pageSize,int pageNow)
    {
```

```

ArrayList<GoodsBean> al=new ArrayList<GoodsBean>();
try{
    ct=new ConnDB().getConn();
    ps=ct.prepareStatement("select * from goods where goodId limit
"+pageSize*(pageNow-1)+",""+pageSize+""");
    rs=ps.executeQuery();
    while(rs.next())
    {
        GoodsBean gb=new GoodsBean();
        gb.setGoodId(rs.getInt(1));
        gb.setGoodName(rs.getString(2));
        gb.setGoodIntro(rs.getString(3));
        gb.setGoodPrice(rs.getInt(4));
        gb.setGoodNum(rs.getInt(5));
        gb.setAuthor(rs.getString(6));
        gb.setPhoto(rs.getString(7));
        gb.setTpye(rs.getString(8));
        gb.setshortint(rs.getString(9));
        al.add(gb);
    }
}catch(Exception e)
{
    e.printStackTrace();

}finally{
    this.close();
}
return al;
}
public GoodsBean getGoodsBean(String id)
{
    GoodsBean gb=new GoodsBean();
    try{
        ct=new ConnDB().getConn();
        ps=ct.prepareStatement("select * from goods where goodId="+id+""");
        rs=ps.executeQuery();
        if(rs.next()){
            gb.setGoodId(rs.getInt(1));
            gb.setGoodName(rs.getString(2));
            gb.setGoodIntro(rs.getString(3));
            gb.setGoodPrice(rs.getInt(4));

```



```
        gb.setGoodNum(rs.getInt(5));
        gb.setAuthor(rs.getString(6));
        gb.setPhoto(rs.getString(7));
        gb.setTpye(rs.getString(8));
        gb.setshortint(rs.getString(9));
    }
}
catch(Exception e)
{
    e.printStackTrace();
}finally{
    this.close();
}
return gb;
}
public void close()
{
    try{
        if(rs!=null){
            rs.close();
            rs=null;
        }
        if(ps!=null)
        {
            ps.close();
            ps=null;
        }
        if(!ct.isClosed())
        {
            ct.close();
        }
    } catch(Exception e)
    {
        e.printStackTrace();
    }
}
}
```

Appendix A-6

MycartBO

```
package com.han.model;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.*;
public class MyCartBO {
    HashMap<String,String> hm=new HashMap<String, String>();
    private ResultSet rs=null;
    private Connection ct=null;
    private PreparedStatement ps=null;
    private float allPrice=0.0f;
    public float returnAllPrice()
    {
        return this.allPrice;
    }
    public String getGoodsNumById(String goodsId)
    {
        return (String)hm.get(goodsId);
    }
    public void addGoods(String goodsid,String goodsNum)
    {
        hm.put(goodsid, goodsNum);
    }
    public void delGoods(String goodsId)
    {
        hm.remove(goodsId);
    }
    public void clear()
    {
        hm.clear();
    }
    public void upGoods(String goodsId,String newNum)
    {
```

```

hm.put(goodsId, newNum);

}

public ArrayList<GoodsBean> showMyCart()
{
    ArrayList<GoodsBean> al=new ArrayList<GoodsBean>();
    try{
        String sql="select * from goods where goodId in";
        Iterator<String> it=hm.keySet().iterator();
        String sub="(";
        while(it.hasNext())
        {

            String goodsId=(String)it.next();
            if(it.hasNext())
                {sub+=goodsId+",";
            }else{
                sub+=goodsId+")";
            }
        }
        sql+=sub;
        ct=new ConnDB().getConn();
        ps=ct.prepareStatement(sql);
        rs=ps.executeQuery();
        this.allPrice=0.0f;
        while(rs.next()){
            GoodsBean gb=new GoodsBean();
            int goodsId=rs.getInt(1);
            gb.setGoodId(goodsId);
            gb.setGoodName(rs.getString(2));
            gb.setGoodIntro(rs.getString(3));
            float unit=rs.getFloat(4);
            gb.setGoodPrice(unit);
            gb.setGoodNum(rs.getInt(5));
            gb.setAuthor(rs.getString(6));
            gb.setPhoto(rs.getString(7));
            gb.setTpye(rs.getString(8));
            gb.setshortint(rs.getString(9));
            this.allPrice=this.allPrice+unit*Integer.parseInt(this.getGoodsNumById(goodsId+""));
            al.add(gb);
        }
    }catch(Exception e)

```

```
{
    e.printStackTrace();
}finally{
    this.close();
}
return al;
}
//close function
public void close()
{
    try{
        if(rs!=null){
            rs.close();
            rs=null;
        }
        if(ps!=null)
        {
            ps.close();
            ps=null;
        }
        if(!ct.isClosed())
        {
            ct.close();
        }
    } catch(Exception e)
    {
        e.printStackTrace();
    }
}
```

Appendix A-7

ORDERBEAN

```
package com.han.model;
public class OrderBean {
    private int orderId;
    private String payMode ;
    private java.util.Date orderDate;
    private byte isPayed;
    private float totalPrice;
    private int usersId;
    public int getOrderId() {
        return orderId;
    }
    public void setOrdersId(int orderId) {
        this.orderId = orderId;
    }
    public String getPayMode() {
        return payMode;
    }
    public void setPayMode(String payMode) {
        this.payMode = payMode;
    }
    public java.util.Date getOrderDate() {
        return orderDate;
    }
    public void setOrderDate(java.util.Date orderDate) {
        this.orderDate = orderDate;
    }
    public byte getIsPayed() {
        return isPayed;
    }
    public void setIsPayed(byte isPayed) {
        this.isPayed = isPayed;
    }
    public double getTotalPrice() {
        return totalPrice;
    }
    }public void setTotalPrice(float totalPrice) {
        this.totalPrice = totalPrice;
    }
    }public int getUsersId() {return usersId;
}public void setUserId(int usersId) {this.usersId = usersId;}}
```



```

values(""+mbo.getGoodsNumByld(gb.getGoodId()+""+""+""+orderId+""+""+gb.getGood
Id()+""));
        }
        //batch handle
        sm.executeBatch();
        String                                sql="select
orderId,trueName,address,postcode,phone,totalPrice,username,email        from
users,orders where orderId="+orderId+" and users.userId=(select orders.userId
from orders where orderId="+orderId+"");
        ps=ct.prepareStatement(sql);
        rs=ps.executeQuery();
        if(rs.next())
        {
            //put rs to orderInfoBean
            oib.setOrderId(rs.getInt(1));
            oib.setTrueName(rs.getString(2));
            oib.setAddress(rs.getString(3));
            oib.setPostcode(rs.getString(4));
            oib.setPhone(rs.getString(5));
            oib.setTotalPrice(rs.getFloat(6));
            oib.setUsername(rs.getString(7));
            oib.setEmail(rs.getString(8));
        }
    }
} catch(Exception e){
    b=false;
    e.printStackTrace();
}
finally{
    this.close();
}
if(b)
{
    return oib;}
else
{
    return null;}
}
public void close()
{
    try{
        if(rs!=null){

```

```
rs.close();
        rs=null;
    }
    if(ps!=null)
    {
        ps.close();
        ps=null;
    }
    if(!ct.isClosed())
    {
        ct.close();
    }
} catch(Exception e)
{
    e.printStackTrace();
}
}
```


Appendix A-9

Orderdetailbean

```
package com.han.model;
public class OrderDetailBean {
private int nums;
    private int orderslid;
    private int goodsId;
    public int getNums() {
        return nums;
    }
    public void setNums(int nums) {
        this.nums = nums;
    }
    public int getOrderslid() {
        return orderslid;
    }
    public void setOrderslid(int orderslid) {
        this.orderslid = orderslid;
    }
    public int getGoodsId() {
        return goodsId;
    }
    public void setGoodsId(int goodsId) {
        this.goodsId = goodsId;
    }
}
```

Appendix A-10

```
OrderInfoBean
package com.han.model;
public class OrderInfoBean {
private int orderId;
private String truename;
private String address;
private String postcode;
private String phone;
private float totalPrice;
private String username;
private String email;
public int getOrderId() {
    return orderId;
}
public void setOrderId(int orderId) {
    this.orderId = orderId;
}
public String getTruename() {
    return truename;
}
public void setTruename(String truename) {
    this.truename = truename;
}
public String getAddress() {
    return address;
}
public void setAddress(String address) {
    this.address = address;
}
public String getPostcode() {
    return postcode;
}
public void setPostcode(String postcode) {
    this.postcode = postcode;
}
public String getPhone() {
    return phone;
}
}
```

```
public void setPhone(String phone) {  
    this.phone = phone;  
}  
public float getTotalPrice() {  
    return totalPrice;  
}  
public void setTotalPrice(float totalPrice) {  
    this.totalPrice = totalPrice;  
}  
public String getUsername() {  
    return username;  
}  
public void setUsername(String username) {  
    this.username = username;  
}  
public String getEmail() {  
    return email;  
}  
public void setEmail(String email) {  
    this.email = email;  
}  
}
```

Appendix A-11

Userbean

```
package com.han.model;
public class UserBean {
    private int userId;
    private String username;
    private String truename;
    private String password;
    private String email;
    private String phone;
    private int grade;
    private String myPic;
    private String address;
    private String postcode;
    public long getUserId() {
        return userId;
    }
    public void setUserId(int userId) {
        this.userId = userId;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getTruename() {
        return truename;
    }
    public void setTruename(String truename) {
        this.truename = truename;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getEmail() {
        return email;
    }
}
```

```
}  
public void setEmail(String email) {  
    this.email = email;  
}  
public String getPhone() {  
    return phone;  
}  
public void setPhone(String phone) {  
    this.phone = phone;  
}  
public int getGrade() {  
    return grade;  
}  
public void setGrade(int grade) {  
    this.grade = grade;  
}  
public String getMyPic() {  
    return myPic;  
}  
public void setMyPic(String myPic) {  
    this.myPic = myPic;  
}  
public String getAddress() {  
    return address;  
}  
public void setAddress(String address) {  
    this.address = address;  
}  
public String getPostcode() {  
    return postcode;  
}  
public void setPostcode(String postcode) {  
    this.postcode = postcode;  
}  
}
```

Appendix A-12

UserBeanBO

```

package com.han.model;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
public class UserBeanBO {
    private ResultSet rs=null;
    private Connection ct=null;
    private PreparedStatement ps=null;
    public UserBean getUserBean(String u)
    {
        UserBean ub=new UserBean();
        try{
            ct=new ConnDB().getConn();
            ps=ct.prepareStatement("select * from users where username='"+u+"'
limit 0,1");
            rs=ps.executeQuery();
            if(rs.next())
            {
                ub.setUserId(rs.getInt(1));
                ub.setUsername(rs.getString(2));
                ub.setTruename(rs.getString(3));
                ub.setPassword(rs.getString(4));
                ub.setEmail(rs.getString(5));
                ub.setPhone(rs.getString(6));
                ub.setGrade(rs.getInt(7));
                ub.setPostcode(rs.getString(9));
                ub.setAddress(rs.getString(10));
            }
        }catch(Exception e){
            e.printStackTrace();
        }
        finally{
            this.close();
        }
        return ub;
    }
    public boolean checkUser(String u,String p)

```

```

{
    boolean b=false;
    try{
        ct=new ConnDB().getConn();
        ps=ct.prepareStatement("select    password    from    users    where
username='"+u+"' limit 0,1");
        rs=ps.executeQuery();
        if(rs.next())
        {
            String dbPasswd=rs.getString(1);
            if(dbPasswd.equals(p))
            {
                b=true;
            }
        }
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
    finally{
        this.close();
    }
    return b;
}
public void close()
{
    try{
        if(rs!=null){

            rs.close();
            rs=null;
        }
        if(ps!=null)
        {
            ps.close();
            ps=null;

        }
        if(!ct.isClosed())
        {ct.close(); }} catch(Exception e){e.printStackTrace();}}

```

Appendix B-1

EnoughCl.java

```

package com.han.servlet;
import com.han.model.*;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class EnoughCl extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        enoughBeanBO ebb=new enoughBeanBO();
        String content=request.getParameter("content");
        enoughBean eb=ebb.getEnoughBean(content);

        if(eb!=null)
        {
            request.setAttribute("content", eb);
            request.getRequestDispatcher("enough.jsp").forward(request,
response);
        }
        else
        {
            request.getRequestDispatcher("index.jsp").forward(request, response);
        }

    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {this.doGet(request, response);}

```


Appendix B-2

LoginCl

```

package com.han.servlet;
import com.han.model.*;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class LoginCl extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String u=request.getParameter("username");
        String p=request.getParameter("password");
        UserBeanBO ubb=new UserBeanBO();
        if(ubb.checkUser(u, p)){
            UserBean ub=ubb.getUserBean(u);
            request.getSession().setAttribute("userInfo", ub);
            MyCartBO
mcb=(MyCartBO)request.getSession().getAttribute("mycart");
            ArrayList al=mcb.showMyCart();
            request.setAttribute("mycartInfo", al);
            request.getRequestDispatcher("shopping3.jsp").forward(request,
response);
        }
        else
        {
            request.getRequestDispatcher("shopping2.jsp").forward(request,
response);
        }
    }
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {this.doGet(request, response);}
}

```

Appendix B-3

Orderclservlet

```

package com.han.servlet;
import com.han.model.*;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class OrderCIServlet extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        OrderBeanBO obb=new OrderBeanBO();
        MyCartBO mcb=(MyCartBO)request.getSession().getAttribute("mycart");
        long userId=((UserBean)request.getSession().getAttribute("userInfo")).getUserId();
        OrderInfoBean oib=obb.addOrder(mcb, userId+ "");
        if(oib!=null)
        {
            request.setAttribute("sss", oib);
            request.getRequestDispatcher("shopping4.jsp").forward(request,
response);
        }
        else
        {
            request.getRequestDispatcher("shopping3.jsp").forward(request,
response);
        }
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
this.doGet(request, response);
    }
}

```

Appendix B-4

ShoppingCl2

```
package com.han.servlet;
import com.han.model.*;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ShoppingCl2 extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        UserBean ub=(UserBean)request.getAttribute("userInfo");
        if(ub==null)
        {
            request.getRequestDispatcher("shopping2.jsp").forward(request,
response);
        }
        else{
            request.getRequestDispatcher("shopping3.jsp").forward(request,
response);
        }
    }
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        this.doGet(request, response);
    }
}
```

Appendix B-5

ShoppingCIServlet

```

package com.han.servlet;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.han.model.MyCartBO;
public class ShoppingCIServlet extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        @SuppressWarnings("unused")
        PrintWriter out= response.getWriter();
        String type=request.getParameter("type");
        MyCartBO mbo=(MyCartBO )request.getSession().getAttribute("mycart");
        if(mbo==null)
        {
            mbo=new MyCartBO();
            request.getSession().setAttribute("mycart", mbo);
        }
        if(type.equals("addGoods"))
        {
            String goodsId=request.getParameter("goodsId");
            mbo.addGoods(goodsId,"1");
        }
        else if(type.equals("delGoods"))

        {
            String goodsId=request.getParameter("goodsId");
            mbo.delGoods(goodsId);

        }
        }else if(type.equals("delAll"))
        {

```

```
mbo.clear();

    }else if(type.equals("show"))
    {

    }else if(type.equals("updateGoods"))
    {
        String goodsId[]=request.getParameterValues("goodsId");
        String newNums[]=request.getParameterValues("newNums");
        for(int i=0;i<goodsId.length;i++)
        {
            System.out.println("id="+goodsId[i]+"amount="+newNums[i]);
            //modify
            mbo.upGoods(goodsId[i], newNums[i]);
        }
    }
    ArrayList al=mbo.showMyCart();
    request.setAttribute("mycartinfo", al);
    request.getRequestDispatcher("showMycart.jsp").forward(request,
response);
}
public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    this.doGet(request, response);
}
}
```

Appendix B-6

```

ShowGoodsServlet
package com.han.servlet;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.han.model.GoodsBean;
import com.han.model.GoodsBeanBO;
public class ShowGoodsServlet extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out= response.getWriter();
    String type=request.getParameter("type");
    if(type.equals("showDetail"))
    {
        String goodsId=request.getParameter("id");
        GoodsBeanBO gbb=new GoodsBeanBO();
        GoodsBean gb=gbb.getGoodsBean(goodsId);
        request.setAttribute("goodsInfo", gb);
        request.getRequestDispatcher("showDetail.jsp").forward(request,
response);
    }
    else if(type.equals("fenye"))
    {
        String pageNow=request.getParameter("pageNow");
        request.setAttribute("abc", pageNow);
        request.getRequestDispatcher("book.jsp").forward(request, response);
    }
}
public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

this.doGet(request, response);
}

}

```

Appendix C-1

Book.jsp

```

<%@ page language="java" import="java.util.*" pageEncoding="ISO-8859-1"%>
<%
String path = request.getContextPath();
String                               basePath                               =
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+p
ath+"/";
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <base href="<%=basePath%>">
    <title>My JSP 'index.jsp' starting page</title>
    <meta http-equiv="pragma" content="no-cache">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
    <meta http-equiv="description" content="This is my page">
    <!-- css import -->
    <link rel="stylesheet" type="text/css" href="css/mycss.css">
  </head>
  <body >
<table width="80%" border="1" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td height="77" colspan="2" align="center" valign="middle">
      <jsp:include page="head.jsp"></jsp:include>
    </td></tr> <tr>
    <td width="21%" height="300" align="center" valign="top">
      <jsp:include page="left.jsp"></jsp:include></td>
    <td width="69%" align="center" valign="middle">
      <jsp:include page="right.jsp"></jsp:include></td>
    </tr>
  <tr>
    <td height="71" colspan="2" align="center" valign="middle">
      <jsp:include page="tail.jsp"></jsp:include></td>
    </tr>
</table>
  </body>
</html>

```

Appendix C-2

```

Enough.jsp
<%@ page language="java" import="java.util.*,com.han.model.*"
pageEncoding="ISO-8859-1"%>
<%
String path = request.getContextPath();
String basePath =
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+p
ath+"/";
OrderInfoBean oib=(OrderInfoBean)request.getAttribute("sss");
enoughBean eb=(enoughBean)request.getAttribute("content");
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <base href="<%=basePath%>">
    <title>My JSP 'enough.jsp' starting page</title>
    <meta http-equiv="pragma" content="no-cache">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
    <meta http-equiv="description" content="This is my page">
    <link rel="stylesheet" type="text/css" href="css/mycss.css">
  </head>
  <body>
    <center>
      <table width="80%" border="1">
        <tr>
          <td align="center"><jsp:include page="head.jsp"></jsp:include></td>
        </tr> <tr>
          <td align="center"><table width="100%" border="1">
            <tr class="abc">
              <td align="center"><%=eb.getEnoughIntro() %></td>
            </tr> </table></td>
          </tr> <tr>
            <td align="center"><jsp:include page="tail.jsp"></jsp:include></td>
          </tr>
        </table>
      </center></body>
    </html>

```


Appendix C-3

Head.jsp

```

<table width="100%" height="144" border="1">
  <tr>
    <td height="12" colspan="3" bgcolor="#FF9900"></td>
  </tr>
  <tr>
    <td width="11%" height="71" rowspan="2" align="center"></td>
    <td width="75%" rowspan="2" align="center"></td>
    <td width="14%" height="37"><span class="STYLE1"><a
href="ShoppingCl2">My Account </a>  </span></td>
  </tr>
  <tr>
    <td><span class="STYLE1"><a href="ShoppingClServlet?type=show">My
ShopingCar</a></span></td>
  </tr><tr>
    <td colspan="3" height="12" bgcolor="#FF9900"></td>
  </tr><tr>
    <td colspan="3"><table width="100%" border="1">
      <tr>
        <td width="11%">&nbsp;</td>
        <td width="11%" align="center" class="bh"><a href="index.jsp">Home
Page</a> </td>
        <td width="11%" align="center">&nbsp;</td>
        <td width="11%" align="center" class="bh"><a
href="book.jsp">Books</a></td>
        <td width="11%" align="center">&nbsp;</td>
        <td width="11%" align="center" class="bh"><span
class="bh">Articles</span></td>
        <td width="11%">&nbsp;</td>
        <td width="11%" align="center" class="bh"><span class="bh">About Us
</span></td>
        <td width="12%">&nbsp;</td>
      </tr>
    </table></td></tr></table>

```

Appendix C-4

```

index.jsp
<%@ page language="java" import="java.util.*" pageEncoding="ISO-8859-1"%>
<%
String path = request.getContextPath();
String basePath =
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+p
ath+"/";
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <base href="<%=basePath%>">
    <title>My JSP 'right2.jsp' starting page</title>
    <meta http-equiv="pragma" content="no-cache">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
    <meta http-equiv="description" content="This is my page">
    <link rel="stylesheet" type="text/css" href="css/mycss.css">
  </head>
  <body topmargin="0">
    <center>
<table width="80%" border="1">
  <tr class="abc">
    <td colspan="2" align="center">
      <jsp:include page="head.jsp"></jsp:include></td>
  </tr>
  <tr>
    <td width="21%" align="center" valign="top" class="abc">
      <jsp:include page="left.jsp"></jsp:include></td>
    <td width="100%" align="center" valign="top">
      <form action="EnoughCI" method="post"><table width="100%"
align="center" border="1" class="abc">
        <tr>
          <td colspan="2" align="center" height="27"><strong>I am
enough</strong>
          <input name="content" type="text" size="30" />
          <input type="submit" name="Submit" value="go"/></td>
        </tr>
      </table>
    </td>
  </tr>

```

```
<td colspan="2" align="center" ></td>
```

```
</tr>
```

```
<tr>
```

```
<td width="30%" align="center" valign="top" bgcolor="#0066FF"></td>
```

```
<td width="70%" align="left" valign="top" bgcolor="#0066FF"><p
align="left"><p align="center"> <strong>What was the project all
about?</strong></p>
```

This mass participation project aimed to have a large number of people spending a few days carrying out an exercise designed to boost their happiness. Because emotions are contagious it was hoped that their increased happiness would pass to those around them and help cheer up the world!

The project took place between Monday 3rd - Friday 7th August 2009, and was conducted by psychologist Richard Wiseman (University of Hertfordshire and author of 59 Seconds: Think a Little, Change a Lot). What happened?

Over 26,000 people signed up for the project. We will submit a full report describing the project to a scientific journal soon, but here are some initial results.

```
</td>
```

```
</tr>
```

```
<tr>
```

```
</tr>
```

```
<tr>
```

```
<td colspan="2" align="center">
```

```
<object width="480" height="385"><param name="movie"
value="http://www.youtube.com/v/G5fLMDId994?fs=1&hl=zh_CN"></param><
param name="allowFullScreen" value="true"></param><param
name="allowscriptaccess" value="always"></param><embed
src="http://www.youtube.com/v/G5fLMDId994?fs=1&hl=zh_CN"
type="application/x-shockwave-flash" allowscriptaccess="always"
allowfullscreen="true" width="360" height="280"></embed></object>
```

```
</td>
```

```
</tr>
```

```
</table></form></td>
```

```
</tr>
```

```
<tr class="abc">
```

```
<td colspan="2" align="center">
```

```
<jsp:include page="tail.jsp"></jsp:include></td>
```

```
</tr>
```

```
</table>
```

```
</center> </body></html>
```

Appendix C-5

Left.jsp

```

<table width="100%" border="1">
  <tr>
    <td align="center"><table width="100%" border="1" class="abc">
      <tr>
        <td align="center"></td>
      </tr>
      <tr>
        <td align="center" ><script language="javascript" src="banner.js"></script>
</td>
      </tr>
    </table></td>
  </tr>
  <tr>
    <td align="center" height="15" bgcolor="#FFCCCC"></td>
  </tr>
  <tr>
    <td align="center"><table width="100%" border="1" class="abc">
      <tr>
        <td colspan="3" align="center">TOP 7</td>
      </tr>
      <tr>
        <td align="center"><strong>Rank</strong></td>
        <td align="center"><strong>Books </strong></td>
        <td align="center"><strong>Clicks</strong></td>
      </tr>
      <tr>
        <td align="center">1</td>
        <td align="center"><a
href="ShowGoodsServlet?type=showDetail&id=1">Happiness Unlocking</a></td>
        <td align="center">13</td>
      </tr>
      <tr>
        <td align="center">2</td>
        <td align="center"><a
href="ShowGoodsServlet?type=showDetail&id=2">Stumbling on Happiness</a></td>
        <td align="center">11</td>
      </tr>
    </table>
  </tr>

```

```

<tr>
  <td align="center">3</td>
  <td align="center"><a
href="ShowGoodsServlet?type=showDetail&id=3">Thanks!</a></td>
  <td align="center">10</td>
</tr>
<tr>
  <td align="center">4</td>
  <td align="center"><a
href="ShowGoodsServlet?type=showDetail&id=4">The Art of Happiness</a></td>
  <td align="center">8</td>
</tr>
<tr>
  <td align="center">5</td>
  <td align="center"><a
href="ShowGoodsServlet?type=showDetail&id=5">How to Achieve
Happiness</a></td>
  <td align="center">5</td>
</tr>
<tr>
  <td align="center">6</td>
  <td align="center"><a
href="ShowGoodsServlet?type=showDetail&id=6">The How of Happiness</a></td>
  <td align="center">3</td>
</tr>
<tr>
  <td align="center">7</td>
  <td align="center"><a
href="ShowGoodsServlet?type=showDetail&id=7">Zen and the Art of
Happiness</a></td>
  <td align="center">2</td>
</tr>
</table></td>
</tr>
</table>

```

Appendix C-6

```

Right.jsp
<%@ page language="java" import="java.util.*,com.han.model.*"
pageEncoding="ISO-8859-1"%>
<%
String path = request.getContextPath();
String basePath =
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+p
ath+"/";
GoodsBeanBO gbb=new GoodsBeanBO();
String s_pageNow=(String)request.getAttribute("abc");
int pageNow=1;
if(s_pageNow!=null)
{
    pageNow=Integer.parseInt(s_pageNow);
}
ArrayList al=gbb.getGoodsByPage(6,pageNow);
int pageCount=gbb.getPageCount(6);
%>
<table width="100%" border="1" class="STYLE1">
    <tr>
        <td colspan="3" align="left">
        </td>
    </tr>
</table>
<%
    int time=0;
    for(int i=0;i<2;i++)
    {
        %>
        <tr>
        <%
            for(int j=0;j<3;j++)
            {
                GoodsBean gb=new GoodsBean();
                if(time>=al.size())
                {
                    gb.setGoodId(0);
                    gb.setshortint("None");
                    gb.setPhoto("none.jpg");
                    gb.setGoodPrice(0.0f);
                }
            }
        %>
    }
%>

```

```

gb.setGoodName("none");
    }else{
        gb=(GoodsBean)al.get(time);
        time++;}
    %>
    <td width="33%" height="280" align="center"><table width="99%"
height="280" border="1" class="abc">
    <tr>
        <td width="53%" rowspan="3" align="center"></td>
        <td width="47%" height="59">&nbsp;</td>
    </tr><tr>
        <td height="69" align="left" valign="top"><a
href="ShowGoodsServlet?type=showDetail&id=<%=gb.getGoodId() %>"><%=gb.getG
oodName() %></a> </td>
    </tr><tr>
        <td height="21" align="left" valign="top">Price $<%=gb.getGoodPrice() %>
</td>
    </tr>
    <tr>
        <td height="70" colspan="2" align="left" valign="top">Brief
Introduction:<%=gb.getshortint() %> </td>
    </tr>
</table></td>
    <% } %></tr><%
    if(i==0)
    {
    %>
        <tr>
            <td colspan="3" align="center" bgcolor="#FFCCCC">&nbsp;</td>
        </tr>
        <%
            } }%>
    <tr> <td colspan="3" align="center">
    <%
        for(int i=1;i<=pageCount;i++)
        { %>

        <a href="ShowGoodsServlet?type=fenye&pageNow=<%=i %>"><%=i %></a>
    <%
        }%> </td> </tr></table>

```

Appendix C-7

Shopping2.jsp

```

<%@ page language="java" import="java.util.*" pageEncoding="ISO-8859-1"%>
<%
String path = request.getContextPath();
String                               basePath                               =
request.getScheme()+"://" + request.getServerName()+":"+request.getServerPort()+p
ath+"/";
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <base href="<%=basePath%>">
    <title>My JSP 'shopping2.jsp' starting page</title>
    <meta http-equiv="pragma" content="no-cache">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
    <meta http-equiv="description" content="This is my page">
    <link rel="stylesheet" type="text/css" href="css/my.css">
  </head>
  <body topmargin="0">
    <center>
      <table width="80%" border="1">
        <tr>
          <td align="center">
            <jsp:include page="head.jsp"></jsp:include></td>
        </tr>
        <tr>
          <td align="center"><table width="100%" border="1">
            <tr>
              <td align="center"></td>
            </tr>
            <tr>
              <td align="center">
                <form action="LoginCl" method="post">

```



```

<table width="33%" border="1" bordercolor="#00CC33" class="abc">
  <tr>
    <td colspan="2" align="center"><strong>LOGIN</strong></td>
  </tr>
  <tr align="center">
    <td width="36%" align="right">Username:</td>
    <td width="64%" align="left"><input name="username" type="text"
size="15" /></td>
  </tr>
  <tr align="center">
    <td align="right">Password:</td>
    <td align="left"><input name="password" type="password" size="15"
/></td>
  </tr>
  <tr align="center">
    <td align="right"><input type="submit" name="Submit" value="Login"
/></td>
    <td align="left"><input type="submit" name="Submit2"
value="regiter" /></td>
  </tr>
</table></form></td>
</tr>
<tr>
  <td align="right"></td>
</tr>
</table></td>
</tr>
<tr>
  <td align="center"><jsp:include page="tail.jsp"></jsp:include></td>
</tr>
</table>
</center>
</body>
</html>

```

Appendix C-8

Shopping3.jsp

```

<%@ page language="java" import="java.util.*,com.han.model.*"
pageEncoding="ISO-8859-1"%>
<%
String path = request.getContextPath();
String basePath =
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+p
ath+"/";
//get user information
UserBean ub=(UserBean)session.getAttribute("userInfo");
//get shopping car information
ArrayList al=(ArrayList)request.getAttribute("mycartInfo");
//get shopping car
MyCartBO mcb=(MyCartBO)session.getAttribute("mycart");
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <base href="<%=basePath%>">
    <title>My JSP 'shopping3.jsp' starting page</title>
    <meta http-equiv="pragma" content="no-cache">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
    <meta http-equiv="description" content="This is my page">
    <link rel="stylesheet" type="text/css" href="css/mycss.css">
  </head>
  <body topmargin="0">
    <center>
      <table width="80%" border="1" class="abc">
        <tr>
          <td align="center">
            <jsp:include page="head.jsp"></jsp:include></td>
          </tr>
          <tr>
            <td align="center"><table width="100%" border="1">
              <tr>
                <td align="center"></td>
              </tr>
            </table>
            </td>
          </tr>
        </table>
    </center>
  </body>
</html>

```

```

<tr>
  <td align="center"><table width="70%" border="1">
    <tr>
      <td colspan="2" align="center">Buyer Information </td>
    </tr>
    <tr>
      <td align="right">&u>Username</td>
      <td><input type="text" name="textfield"
value="<%=ub.getUsername() %>"/></td>
    </tr>
    <tr>
      <td align="right">*Turname</td>
      <td><input type="text" name="textfield2"
value="<%=ub.getTruename() %>"/></td>
    </tr>
    <tr>
      <td align="right">*Address</td>
      <td><input type="text" name="textfield3"
value="<%=ub.getAddress() %>"/></td>
    </tr>
    <tr>
      <td align="right">*Telephone</td>
      <td><input type="text" name="textfield4"
value="<%=ub.getPhone() %>"/></td>
    </tr>
    <tr>
      <td align="right">*Email:</td>
      <td><input type="text" name="textfield5"
value="<%=ub.getEmail() %>"/></td>
    </tr>
    <tr>
      <td align="right">*Postcode:</td>
      <td><input type="text" name="textfield6"
value="<%=ub.getPostcode() %>"/></td>
    </tr>
    <tr>
      <td align="right"><input type="submit" name="Submit"
value="Complete Booking" /></td>
      <td><input type="submit" name="Submit2" value="Modify User
Information" /></td>
    </tr>
  </table></td> </tr>

```

```

<tr>
  <td align="center"><table width="70%" border="1" class="abc">
    <tr>
      <td colspan="4" align="center">My Shopping Cart </td>
    </tr>
    <tr>
      <td align="center">Id</td>
      <td align="center">Goods Name </td>
      <td align="center">Price</td>
      <td align="center">Amount</td>
    </tr>
    <%
      for(int i=0; i<al.size();i++)
    {
      GoodsBean gb=(GoodsBean)al.get(i);
    %>
    <tr>
      <td align="center"><%=gb.getGoodId() %></td>
      <td align="center"><%=gb.getGoodName() %></td>
      <td align="center"><%=gb.getGoodPrice() %></td>
      <td
align="center"><%=mcb.getGoodsNumById(gb.getGoodId()+"" ) %></td>
    </tr> <% } %><tr>
      <td colspan="4" align="left">You have been choose
$<%=mcb.returnAllPrice() %> goods. </td>
    </tr><tr>
      <td colspan="4" align="center"><input type="submit"
name="Submit3" value="Return modify shopping cart" /></td>
    </tr> </table></td></tr> <tr>
      <td align="right"><a href="OrderClServlet" ></a></td>
    </tr>
  </table></td>
</tr><tr>
  <td align="center"><jsp:include page="tail.jsp"></jsp:include><br></td>
</tr>
</table>
</center>
</body>
</html>

```

Appendix C-9

Shopping4.jsp

```

<%@ page language="java" import="java.util.*,com.han.model.*"
pageEncoding="ISO-8859-1"%>
<%
String path = request.getContextPath();
String basePath =
request.getScheme()+ "://" + request.getServerName() + ":" + request.getServerPort() + p
ath + "/";
OrderInfoBean oib=(OrderInfoBean)request.getAttribute("sss");
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <base href="<%=basePath%>">
    <title>My JSP 'shopping4.jsp' starting page</title>
    <meta http-equiv="pragma" content="no-cache">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
    <meta http-equiv="description" content="This is my page">
    <link rel="stylesheet" type="text/css" href="css/mycss.css">
  </head>
  <body topmargin="0">
<center>
<table width="80%" border="1">
  <tr>
    <td align="center">
      <jsp:include page="head.jsp"></jsp:include></td>
    </tr>
  <tr class="abc">
    <td align="center"><table width="100%" border="1">
      <tr>
        <td colspan="9" align="center"></td>
      </tr>
      <tr>
        <td colspan="9" align="center">Booking detail </td>
      </tr>
      <tr>
        <td width="10%" align="center">Booking number </td>

```

```

<td width="9%" align="center">receiver</td>
  <td width="17%" align="center">address</td>
  <td width="9%" align="center">Postcode</td>
  <td width="11%" align="center">phone</td>
  <td width="11%" align="center">total price </td>
  <td width="10%" align="center">Username</td>
  <td width="16%" align="center">Email</td>
  <td width="7%" align="center">More detail </td>
</tr>
<tr>
  <td align="center"><%=oib.getOrderId()%></td>
  <td align="center"><%=oib.getTruename()%></td>
  <td align="center"><%=oib.getAddress() %> </td>
  <td align="center"><%=oib.getPostcode()%></td>
  <td align="center"><%=oib.getPhone()%></td>
  <td align="center">$<%=oib.getTotalPrice()%></td>
  <td align="center"><%=oib.getUsername()%></td>
  <td align="center"><%=oib.getEmail()%></td>
  <td align="center"><a href="#">detail</a></td>
</tr>
<tr>
  <td colspan="9" align="center">Your booking is complete,our server will
send you a confirm Email later </td>
</tr>
</table></td>
</tr>
<tr>
  <td align="center">
    <jsp:include page="tail.jsp"></jsp:include></td>
</tr>
</table>
</center>
</body>
</html>

```

Appendix C-10

Showdetail.jsp

```

<%@ page language="java" import="java.util.*,com.han.model.*"
pageEncoding="ISO-8859-1"%>
<%
String path = request.getContextPath();
String basePath =
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+p
ath+"/";
GoodsBean gb=(GoodsBean)request.getAttribute("goodsInfo");
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <base href="<%=basePath%>">
    <title>My JSP 'showdetail.jsp' starting page</title>
    <meta http-equiv="pragma" content="no-cache">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
    <meta http-equiv="description" content="This is my page">
    <link rel="stylesheet" type="text/css" href="css/mycss.css">
  <script type="text/javascript">

  <!--
    function returnHall()
    {
      window.open("index.jsp", "_self");
    }
    function addGoods(goodsId)
    {
      window.open("ShoppingClServlet?type=addGoods&goodsId="+goodsId,"_self");
    }
  -->
</script>
  </head>
  <body>
  <center>
  <table width="71%" border="1">
    <tr>
      <td align="center">

```

```

<jsp:include page="head.jsp"></jsp:include></td>
</tr><tr>
  <td><table width="100%" border="1">
    <tr>
      <td height="18" colspan="2">&nbsp;</td>
    </tr><tr>
      <td width="26%" rowspan="8"></td>
      <td
          width="74%"
          align="center"
class="abc"><%=gb.getGoodName()%></td>
    </tr><tr>
      <td class="abc">Price:<%=gb.getGoodPrice() %> </td>
    </tr><tr>
      <td class="abc">Author:<%=gb.getAuthor() %> </td>
    </tr><tr>
      <td class="abc">ISBN: <%=gb.getGoodId() %> </td>
    </tr> <tr>
      <td class="abc">Type:<%=gb.getTpye()%></td>
    </tr> <tr>
      <td>&nbsp;</td>
    </tr> <tr>
      <td>&nbsp;</td>
    </tr><tr>
      <td
          height="107"
          align="left"
          valign="top"
class="abc"><%=gb.getGoodIntro() %>
      </td></tr>
      <tr><td
          height="26"
          colspan="2"><input
          type="button"
onclick="addGoods(<%=gb.getGoodId()%>)" name="Submit" value="Buy" />
          <input
          type="button"
          name="Submit2"
          onclick="returnHall();"
value="Return Back" /></td>
      </tr><tr>
      <td height="19" colspan="2">&nbsp;</td>
    </tr></table></td></tr><tr>
  <td align="center">
    <jsp:include page="tail.jsp"></jsp:include></td>
  </tr></table></center>
</body>
</html>

```


ShowMycart

```

<%@ page language="java" import="java.util.*,com.han.model.*"
pageEncoding="ISO-8859-1"%>
<%
String path = request.getContextPath();
String base Path=
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+p
ath+"/";
ArrayList al=(ArrayList)request.getAttribute("mycartinfo");
MyCartBO mbo=(MyCartBO)session.getAttribute("mycart");
%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <base href="<%=basePath%>">
    <title>My JSP 'showMycart.jsp' starting page</title>
    <meta http-equiv="pragma" content="no-cache">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
    <meta http-equiv="description" content="This is my page">
    <link rel="stylesheet" type="text/css" href="css/mycss.css">
    <script type="text/javascript">
      <!--
        function deAll()
      {
        window.open("ShoppingClServlet?type=delAll","_self");
      }
      -->
    </script>
  </head>
  <body>
    <center>
      <table width="80%" border="1">
        <tr>
          <td align="center">
            <jsp:include page="head.jsp"></jsp:include></td>
          </tr>
        <tr>

```

```

<td align="center">
  <form action="ShoppingClServlet?type=updateGoods" method="post">
    <table width="100%" border="1" class="abc">
      <tr>
        <td colspan="6" align="center"></td>
      </tr>
      <tr align="center">
        <td width="24%">ID</td>
        <td width="25%">Name</td>
        <td width="25%">Price</td>
        <td colspan="3">Amount</td>
      </tr>
<%
  for(int i=0;i<al.size();i++)
  {
//draw al from goodsbean
GoodsBean gb=(GoodsBean)al.get(i);

  %>
    <tr>
      <td align="center"><%=gb.getGoodId()%></td>
      <td align="center"><%=gb.getGoodName() %></td>
      <td align="center">${<%=gb.getGoodPrice() %></td>
      <td width="9%" align="center"><input type="hidden" name="goodId"
value="<%=gb.getGoodId() %>"><input name="newNums" type="text" size="7"
value="<%=mbo.getGoodsNumById(gb.getGoodId()+"" ) %>" /></td>
      <td width="8%" align="center"><a
href="ShoppingClServlet?type=delGoods&goodId=<%=gb.getGoodId()%>">Delete</
a></td>
      <td width="9%" align="center"><a
href="ShowGoodsServlet?type=showDetail&id=<%=gb.getGoodId()%>">Search</a><
/td>
    </tr>
  <%
  }%>
    <tr>
      <td align="center">&nbsp;</td>
      <td align="center"><input onclick="deAll();" type="button"
name="Submit" value="Delete all items" /></td>
      <td align="center"><input type="submit" name="Submit2" value="Modify
Number" /></td>

```

```

<td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td colspan="6" align="center">&nbsp;</td>
  </tr>
</table>
</form>
<tr>
  <td colspan="6" align="center"><table class="abc" width="100%"
border="1">
  <tr>
    <td width="46%">The totally items price are
<%=mbo.returnAllPrice()%>,click<a href=book.jsp> here </a>to continue purchasing.
</td>
    <td width="54%" align="right"><a href="ShoppingCl2"></a></td>
  </tr>
</table></td>
</tr>
</table></td>
</tr>
<tr>
  <td align="center">
    <jsp:include page="tail.jsp"></jsp:include>
  </td>
</tr>
</table>
</center>
</body>
</html>

```

Appendix C-11

Tail.jsp

```
<table width="100%" border="0" class="abc">
  <tr>
    <td align="center" bgcolor="#FFCCCC">Address:Raviradintae 11C 35 Mikkeli
Finland </td>
  </tr>
  <tr>
    <td align="center">Post Code 50100 Email:Charles_h@live.cn </td>
  </tr>
  <tr>
    <td align="center">Copyright Pursuit Happiness </td>
  </tr>
</table>
```