

California State University, San Bernardino

CSUSB ScholarWorks

Theses Digitization Project

John M. Pfau Library

2004

PDF Receipts: A micro-based-payment web service

Yu-Feng Ho

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Databases and Information Systems Commons](#)

Recommended Citation

Ho, Yu-Feng, "PDF Receipts: A micro-based-payment web service" (2004). *Theses Digitization Project*. 2714.

<https://scholarworks.lib.csusb.edu/etd-project/2714>

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

PDF RECEIPTS: A MICRO-PAYMENT-BASED WEB SERVICE

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Yu-Feng Ho
June 2004

PDF RECEIPTS: A MICRO-PAYMENT-BASED WEB SERVICE

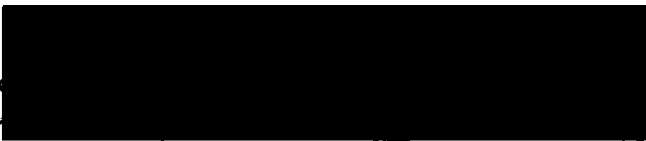
A Project
Presented to the
Faculty of
California State University,
San Bernardino

by

Yu-Feng Ho

June 2004

Approved by:




Dr. David Turner, Chair, Computer Science

5/13/2004

Date



Dr. Ernesto Gomez



Dr. Keith Schubert

ABSTRACT

What are micro-payment-based Web services? Simply, they are Web services that charge a fee for invoking their public operations. Micropayments is a business concept whose goal is to generate revenue by offering pay-per-view Web pages, Web links, or Web services for small amounts of money called "microcents." This project investigates the use of micro-payments for the automated purchase of services available within a marketplace of Web services. The approach taken by this project is to construct prototypes that illustrate how such a Web service marketplace might work. Specifically, a Web service provider and Web service consumer were constructed, and an LCP currency issuing system was used to provide a micro-payment mechanism. The Web service provider sells a PDF receipt generation service to ecommerce sites. The Web service consumer is an online bookstore, which provides receipts in PDF format to its customers. When a bookstore customer requests a receipt in PDF format, the bookstore purchases the receipt from the receipt service. To make this purchase, the bookstore transfers payment to the receipt service in the form of LCP currency. For this purpose, this project relies on an LCP currency issuer built by Steven Hsiao and extended by David Turner.

This process is very convenient for those e-commerce websites to maintain and develop their own critical features without re-implementing common features available in a Web service marketplace.

ACKNOWLEDGMENTS

I thank the faculty of the Computer Science department Department for giving me an opportunity to pursue my M.S. in Computer Science at California State University, San Bernardino. I express my sincere appreciation to my graduate advisor, Dr. David Turner who offered me this project and directed me through this entire effort. I also thank my other committee members, Dr. Keith Schubert and Dr. Ernesto Gomez for their valuable input.

I also want to thank all of the professors in the Computer Science department of CSUSB. I have learned lots of knowledge from the courses they conducted. Without that basic knowledge, it is impossible for me to develop my project even in the beginning.

Finally, I am very grateful to my parents. Without their support, I do not even have a chance to come to United States of America. I am apperiate about my classmates and friends here helping me to deal with many academic and living problems. The support of the National Science Foundation under the award 9810708 is gratefully acknowledged.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER ONE: INTRODUCTION	1
1.1 Purpose of this Project	2
1.2 Scope of Project	4
1.2.1 Phase I	4
1.2.2 Phase II	4
1.2.3 Phase III	4
1.3 Organization of Chapters	5
CHAPTER TWO: ARCHITECTURE	6
2.1 Software Interfaces	9
CHAPTER THREE: DATABASE DESIGN	
3.1 Data Analysis	10
3.2 Database Schema Conceptual Model - ER Diagram	13
3.3 Data Type and Details	15
CHAPTER FOUR: SYSTEM DESIGN	19
CHAPTER FIVE: DEFINITIONS	22
CHAPTER SIX: SECURITY	26
CHAPTER SEVEN: SYSTEM PROCESS	
7.1 Preparation	28
7.2 Flowcharts for the Processing	32

7.3 Sequential Diagram	36
7.4 System Diagram and Brief Process	37
CHAPTER EIGHT: SYSTEM SCENARIO	
8.1 Main Success Scenario	41
8.2 Fail Scenario in the Bookstore	43
8.3 Fail Scenario in the Receipt Service	44
CHAPTER NINE: LIGHTWEIGHT CURRENCY PROTOCOL AND AXIS APPLICATION	
9.1 Lightweight Currency Protocol Application	46
9.2 AXIS Application	47
CHAPTER TEN: USER MANUAL	
10.1 Manual for the Bookstore	48
10.1.1 Functionality Common to All Users	48
10.1.2 Customer Functionality	49
10.1.3 For the Administrator	60
10.2 Manual for the Receipt Service	71
10.2.1 Common Functionality for All Users	71
10.2.2 For the Normal Users	73
10.2.3 For the Administrator	81
CHAPTER ELEVEN: CONCLUSION AND FUTURE DIRECTIONS	
11.1 Conclusion	88
11.2 Future Directions	88
REFERENCES	89

LIST OF TABLES

Table 1.	Structure of Table Admin in the Bookstore	15
Table 2.	Structure of Table Book in the Bookstore	15
Table 3.	Structure of Table Guestlist in the Bookstore	16
Table 4.	Structure of Table Replylist in the Bookstore	16
Table 5.	Structure of Table Info in the Bookstore	17
Table 6.	Structure of Table Rs_user in the Receipt Service	17
Table 7.	Structure of Table Rs_price in the Receipt Service	18
Table 8.	Structure of Table Transaction in the Receipt Service	18

LIST OF FIGURES

Figure 1.	The Relationships of Components in the Architecture	7
Figure 2.	E-R Diagram for the Bookstore	13
Figure 3.	E-R Diagram for the Receipt Service	14
Figure 4.	Use Case Model for Bookstore Administrator	19
Figure 5.	Use Case Model for Bookstore Customer	20
Figure 6.	Use Case Model for Receipt Service Customer	21
Figure 7.	Use Case Model for Receipt Service Administrator	21
Figure 8.	Create an Account in Currency Issuer	29
Figure 9.	The Content in the Rs.cert	31
Figure 10.	The First Part of Detail Processing	32
Figure 11.	The Second Part of Detail Processing	33
Figure 12.	The Third Part of Detail Processing	34
Figure 13.	The Fourth Part of Detail Processing	35
Figure 14.	System Sequential Diagram	36
Figure 15.	Whole System Diagram	37
Figure 16.	Starting Page in the Bookstore	48
Figure 17.	Main Page in the Bookstore	49
Figure 18.	Cart Page	50
Figure 19.	Empty Shopping Cart	51
Figure 20.	SSL Notice	51
Figure 21.	Form Page	52
Figure 22.	Result Page	53
Figure 23.	Text-mode Receipt Page	54

Figure 24. Receipt in PDF Format	55
Figure 25. Search Page	56
Figure 26. Search Result Page	56
Figure 27. Guestbook Page	57
Figure 28. Reply Page	58
Figure 29. Page Responding to Dirty Words	59
Figure 30. Contact Page	60
Figure 31. Login Page	61
Figure 32. Admin Page	62
Figure 33. Transaction Page	63
Figure 34. Chart Page	64
Figure 35. Balance Page	65
Figure 36. Add Book Page	66
Figure 37. Remove Book Page	67
Figure 38. Modify Book Page	68
Figure 39. Modify Detail Page	69
Figure 40. Upload Page	70
Figure 41. Confirm Page	71
Figure 42. Login Page for the Receipt Service	72
Figure 43. Register Page	73
Figure 44. User Page	74
Figure 45. Information Page	75
Figure 46. Modify Information Page	76
Figure 47. Change Password Page	77
Figure 48. Logo Page	78
Figure 49. Upload Logo Page	79
Figure 50. Confirm Page	80

Figure 51. Cert Page	81
Figure 52. Admin Page	82
Figure 53. Transaction Page	83
Figure 54. Chart Page	84
Figure 55. Balance Page	85
Figure 56. Price Page	86
Figure 57. Change Price Page	87

CHAPTER ONE

INTRODUCTION

PDF Receipts is a Web service which provides receipts in PDF format asked for by clients, which are e-commerce websites asking for the service. This service is not free, so those clients must pay the specified amount of "microcents," posted by the receipt service (RS). Our bookstore (BS) was created as an example of those clients. The bookstore is a web bookstore, which sells physical books to online customers. Those customers are the visitors to the Web Pages of the bookstore. The bookstore charges the customer for his or her chosen books that the customer places in a shopping cart. Customers of the bookstore pay by credit card. Thus, the bookstore must collect credit card information from the customer at the time of purchase. After payment is made, the customers can choose to get a receipt in PDF format. When the bookstore requests a PDF receipt from the receipt service, it invokes a purchase request operation on the receipt service's public web service API. The receipt service provides a price in microcents and a transaction ID. The bookstore must then communicate with the currency issuer to transfer the necessary microcents to the receipt service's account.

After verifying payment, the receipt service will produce the PDF receipt, and send it back to the bookstore. Finally, the customer gets his or her requested receipt.

There are transfer and verification fees involved in the processing. Those fees are asked by currency issuer when the bookstore invokes the transfer operation, and when the receipt service invokes the payment verification operation.

1.1 Purpose of this Project

The purpose of this project is to design, build and implement an instance of a specified online payment system, which is based on micro-payments. There are three components for this system: a bookstore, a receipt service and a currency issuer. The bookstore is an instance of an e-commerce enterprise that sells something online. In order to simplify the design of its Web pages, and to speed the process of the customers' purchase, those stores can follow the bookstore's model to concentrate on developing its essential features related to displaying and searching merchandise, shopping carts, customer payment, product suggestions and so on, and leave the execution of other functionality to outside service

providers. So the bookstore can concentrate its efforts on providing its narrow business function.

For non-specialized functions, the bookstore can rely on low-cost information services provided through a Web service marketplace. We chose to illustrate this by a Web service that provides PDF formatting functionality for receipts. Thus, when the bookstore needs a receipt in PDF format, it does not generate such a receipt by itself, but rather purchases the receipt from a receipt formatting service. Because a website such as the bookstore may require many such supporting services, the cost of accessing any one service should be relatively small. For reason, use of a traditional payment mechanism is too costly. In its place, the web service consumers and providers transact using a secure payment protocol that has low transaction costs. To illustrate this process, we use the Lightweight Currency Protocol (LCP) for the payment mechanism. LCP is especially suitable to this application, because of the simplicity by which it can be incorporated into the consumer and provider systems. LCP is conceptually simple, yet fully secure.

In this way, those stores can develop their own features more easilier and in the future there will be more and more unique services provided through the Web

service marketplace. A protocol such as LCP will become more important as its use increases in the Web services marketplace.

1.2 Scope of Project

The project was separated into three phases:

1.2.1 Phase I

The first phase was to construct the bookstore, which is an e-commerce bookstore built with JavaServer Pages (JSP), Java Servlets (Servlet), a PostgreSQL database, various Java classes, and some files configuring the Web application framework, and key identifier files. These components will be described in detail later.

1.2.2 Phase II

The second phase was to build a receipt service, which is a Web service provider offering PDF receipts. The receipt service is built with JavaServer Pages (JSP), Java Servlets (Servlet), a PostgreSQL database, Java classes and a HTML file. They will be described in detail later.

1.2.3 Phase III

The third phase was to implement the interactions between the bookstore, receipt service and currency issuer. The interaction between the bookstore and the receipt service is done using RPC-style SOAP messages. The open

source framework Apache Axis was used to implement the SOAP messaging mechanism.

The interaction between the receipt service and the currency issuer, and between the bookstore and the currency issuer, was done using the LCP protocol. To accomplish this, this project involved the development of LCP java libraries to perform LCP protocol functions.

There are some policies and mechanisms to follow among those interactions and affect the result of whether the customers can get receipts. All of these will be described in detail later.

1.3 Organization of Chapters

Chapter Two introduces the technologies used in the project. Chapter Three discusses the details of system designs. Chapter Four focuses on the implementation details. Chapter Five focuses on testing the process of getting receipts. Chapter Six presents the conclusion and future work.

CHAPTER TWO

ARCHITECTURE

This project implements a Web system to provide an environment for the e-commerce bookstore to access the Web services of another e-commerce website, namely a receipt service that provides receipts in PDF format. There are four machines running in this system. Their relationship is illustrated in Figure 1. The customer buys one or more books in the Bookstore by adding them to a shopping cart and then checking out. After checking out, the customer can choose to get a receipt in PDF format. Once the button for getting a receipt is clicked, the bookstore will invoke a `purchaseReceipt` operation on the receipt service. The receipt service returns a price and a transaction identifier. The bookstore then invokes the `transfer funds` operation on the currency service to send money with the transaction identifier to the receipt service. The currency issuer processes the request by updating its internal state to reflect the movement of funds, and then sends a deposit notification to the bookstore that includes the transaction identifier. The Bookstore invokes the `getReceipt` operation on the receipt service, passing the transaction identifier to it. The receipt service

returns the receipt in PDF format encoded using Base64. The Bookstore decodes the PDF receipt, and sends it back to the customer's browser.

Communication between the browser and the bookstore is done using HTTP (for transport of typical web objects). Communication between the bookstore and the receipt service is done using SOAP/HTTPS (for web service invocations). Communication between the bookstore and the currency issuer, and between the receipt service and the issuer, is done using LCP.

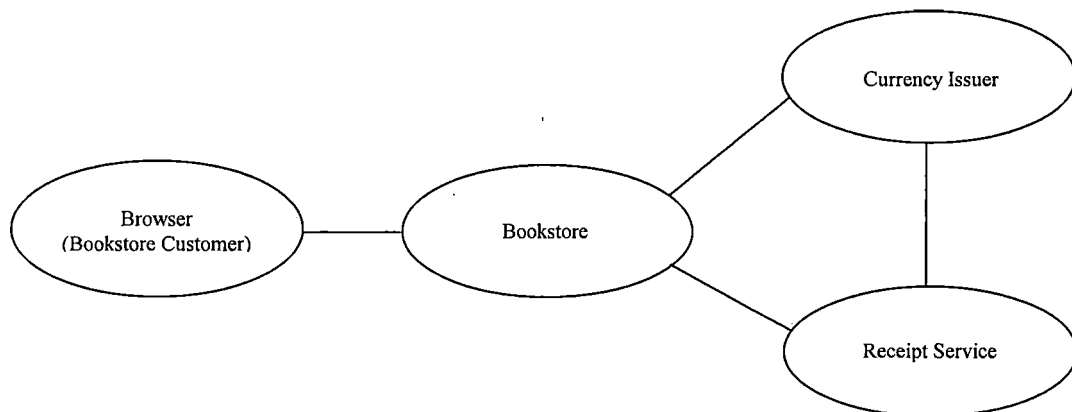


Figure 1. The Relationships of Components in the Architecture

The components used to build this system were chosen with the following criteria: (i) the components should be shareware, i.e., available freely for non-commercial purposes, (ii) be part of a standard, i.e., do not depend on a specific operating system and hence are easily portable across systems with ease, (iii) database server

independent, so that new and different versions of the server can be plugged in easily.

The user interface components are built by using HTML 6.0 forms, frames, Macromedia Flash and Javascript. Flash is used in the bookstore main page to attract customers to get into the bookstore by animated effects. Javascript is used in three places: the animated menu, the text-mode receipt printing feature, and choosing MIDI music to play. The applications are written using JavaServer Pages (JSP) and Java Servlets and are run as Web applications in Apache Tomcat, which is a servlet container. JSP was used because it allows the user of JavaServer Pages Standard Tag Library (JSTL), which encapsulates, as simple tags, core functionality common to many JSP applications and simplifies the contents of JSP Pages. Also, Tomcat can be installed on both Microsoft Windows and Linux systems. The most important part is that JSP, Servlets and Tomcat are all composed of JAVA byte code, so that the system can be run by a JAVA Virtual Machine (JVM). Owing to the JVM's characteristic, this system can be developed and executed regardless of the computer system, and is very suitable to process data received from HTML forms. The reason that Java Servlets was used is that it has the advantages of portability and simplicity. By using Java Servlets, the

Servlet can be executed in any other web server which is the special property of Java Servlets, "write once, serve anywhere." Moreover, Java provides a convenient function, Java Database Connector (JDBC), to connect to databases.

The database used for this system is PostgreSQL. PostgreSQL is a real multi-user database and is royalty-free open source software. To use it, simply activate PostgreSQL in Linux. Also, the availability of the JDBC driver for PostgreSQL is another important reason to choose it. Moreover, the same code could be used to link with another database by changing proper JDBC driver, thereby making it database independent.

2.1 Software Interfaces

- Internet browser: Netscape or Internet Explorer.
- Operating system: Windows 98/Me/2000/XP or Unix/Linux.
- Database: PostgreSQL.
- Compiler: JAVA 2 Platform Standard Edition Software Development Kit 1.4.2.
- Language: HTML / JAVA / JavaScript / JSP / Flash.
- Database connector: JDBC.
- JSP Container/Web server: Jakarta Tomcat.

CHAPTER THREE
DATABASE DESIGN

3.1 Data Analysis

The data for designing and implementing the schema of the database depends on the properties of the bookstore and the receipt service. The data needed by the bookstore is the data of books, the administrator, the buying statistics and the guestbook record. Hence, there are four main tables in the bookstore's database: book, admin, info and guestlist. In the table book, there are seven attributes: ISBN, name, author, price, amount, pic_size and image. The first four attributes self-explanatory, and "amount" stands for the number of books in the inventory of bookstore. The book table also includes the image presenting a photo of the book's front cover, and pic_size is the image size in number of bytes. In the admin table, there are three attributes: userid, password and balance. The first two attributes stand for the administrator's login credentials, and the balance attribute stands for the current balance of LCP funds held by the bookstore. In the info table, there are seven attributes: serial, ISBN, bookname, author, price, printed and order_date. The attribute serial is a special type in the database for a

sequence will be automatically created, so that a proper default value will be assigned to the primary key of each row. The second to fifth attributes are self-explanatory. The attribute "printed" stands for whether the customer asks for the receipt for the corresponding book. The attribute "order_date" records the date when the purchase is made. In the guestlist table, there are six attributes: serial, name, email, comment, send_time and send_date. The attribute serial is similar to the serial column in the book table. The attributes name, email and comment are the information inputted from customers. "Send_time" records the time when the customer sent the information in hour-minute-second mode. "Send_date" records the date when the customer sent the information. The data needed by the receipt service is the data of users, the data related to the price, the records of transactions. Hence, there are three tables: rs_user, rs_price and transaction. In the table rs_user, there are eight attributes: userid, password, logo_size, image, cname, email, phone and address. "Userid" and "password" are the login credentials. This table includes the image stored presenting the corresponding user's logo, which will be used to produce the receipt, and the logo_size stored presenting the logo size. "Cname," "email," "phone" and "address" record the

user's information, in which "cname" standing for the company name will be used in producing the receipt. In the table "rs_price," there are five attributes, "serial," "transId," "price," "balance" and "cert." "Serial" acts as that in the table "book." "transId" is initialized with 0, used by the function producing receipts and updated by that function as well. "Price" stands for the cost for producing a receipt. "Balance" stands for the receipt service's balance initialized with 0 and updated by the function of producing the receipt. "Cert" records the contents of rs.cert composed of the receipt service's public key, and it is needed by the page displaying the public key to the client users. In the table "transaction," there are five attributes, "serial," "transId," "cname," "printed" and "service_date." "Serial" acts as that in the table "book." "TransId" gets the data from "transId" in table "rs_price," increase the value by 1 and stores this value while the function of producing a receipt is working. "Cname" acts as that in table "rs_user." "Printed" stands for whether the whole transaction is successfully done from getting a request to sending the receipt. "Service_date" records the date when the transaction is made.

3.2 Database Schema Conceptual Model - ER Diagram

The ER Diagram for the bookstore is shown in Figure 2. The ER Diagram for the receipt service is shown in Figure 3.

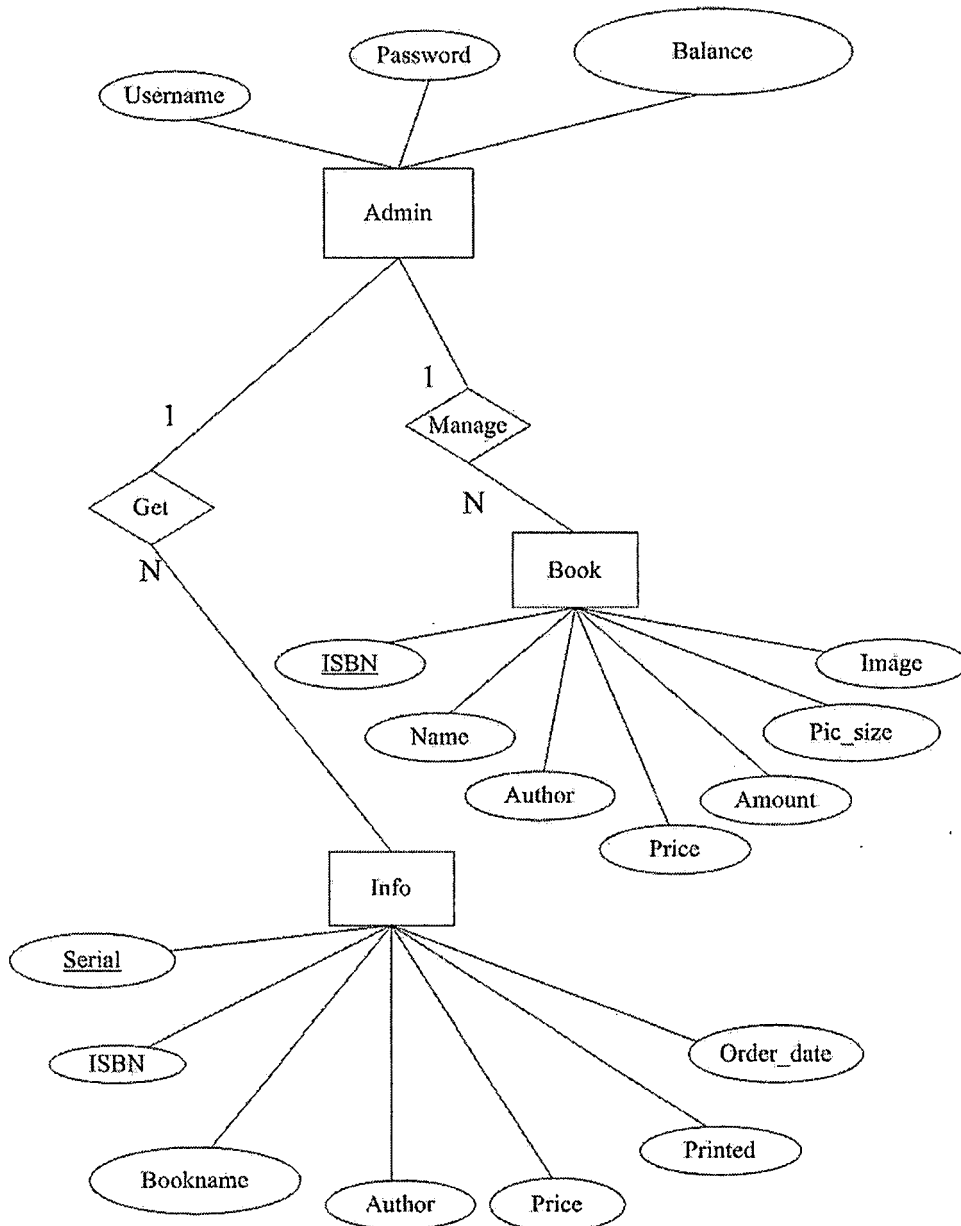


Figure 2. E-R Diagram for the Bookstore

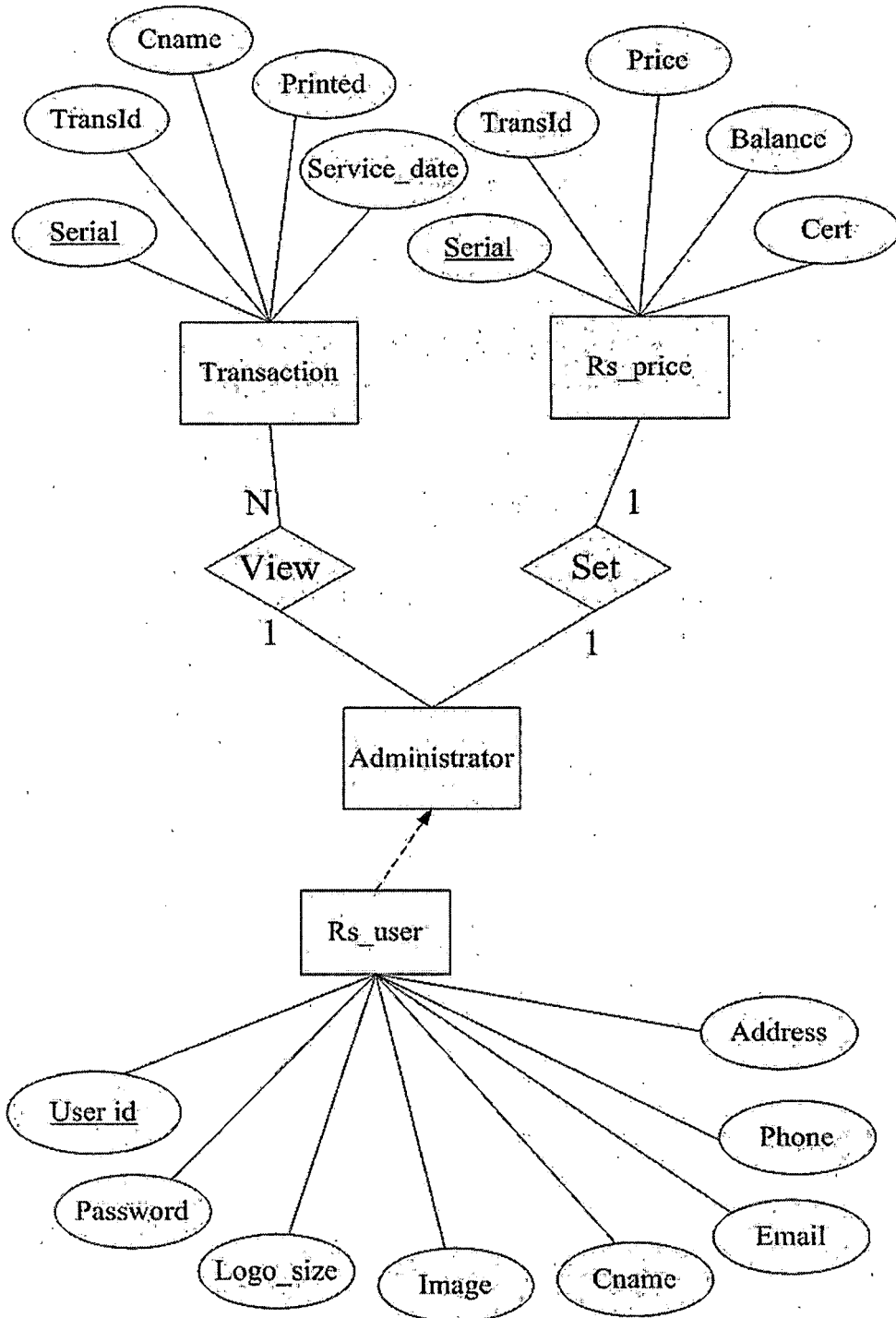


Figure 3. E-R Diagram for the Receipt Service

3.3 Data Type and Details

The logical model establishes the following detailed design in PostgreSQL database. The following tables describe data type, length, primary key, null or non-null keys, and extra information, such as auto_increment.

Table 1. Structure of Table Admin in the Bookstore

Field	type	null	key	default	extra
userid	varchar(255)		PRI	'ho'	
password	varchar(255)			'ho'	
balance	double precision			500.0	

Table 2. Structure of Table Book in the Bookstore

field	type	null	key	default	extra
ISBN	Varchar(255)		PRI		
name	Varchar(255)				
author	Varchar(255)				
price	double precision				
amount	integer				
pic_size	integer	y			
image	bytea	y			

Table 3. Structure of Table Guestlist in the Bookstore

field	type	null	key	default	extra
serial	serial		PRI		auto
name	Varchar(255)				
email	Varchar(255)	y			
comment	Varchar(255)				
send_time	time				
send_date	date				

Table 4. Structure of Table Replylist in the Bookstore

field	type	null	key	default	extra
serial	serial		PRI		auto
source_id	integer				
name	Varchar(255)				
email	Varchar(255)	y			
comment	Varchar(255)				
send_time	time				
send_date	date				

Table 5. Structure of Table Info in the Bookstore

field	type	null	key	default	extra
serial	serial		PRI		auto
ISBN	Varchar(255)				
bookname					
author					
price	Varchar(255)				
printed	Varchar(255)				
order_date	date				

Table 6. Structure of Table Rs_user in the Receipt Service

field	type	null	key	default	extra
userid	Varchar(255)		PRI		
password	Varchar(255)				
logo_size	integer	y			
image	bytea	y			
cname	Varchar(255)				
email	Varchar(255)	y			
phone	Varchar(255)	y			
address	Varchar(255)	y			

Table 7. Structure of Table Rs_price in the Receipt

Service

field	type	null	key	default	extra
serial	serial		PRI		auto
transId	Varchar(255)			'0'	
price	double precision	y		1.0	
balance	double precision	y			
cert	varchar(999)			rs.cert	

Table 8. Structure of Table Transaction in the Receipt

Service

field	type	null	key	default	extra
serial	serial		PRI		auto
transId	varchar(255)				
cname	varchar(255)				
printed	boolean				
service_date	date				

CHAPTER FOUR

SYSTEM DESIGN

There are two types of bookstore users: customers and administrators. Customers visit the bookstore, build a shopping cart of books, and purchase the books by entering credit card information. Customers do not need to log in. Administrators view customer purchases, which include customer names, purchased items and whether a receipt was required. Administrators need to log in. The bookstore administrator use case diagram is shown in Figure 4 The bookstore customer use case diagram is shown in Figure 5.

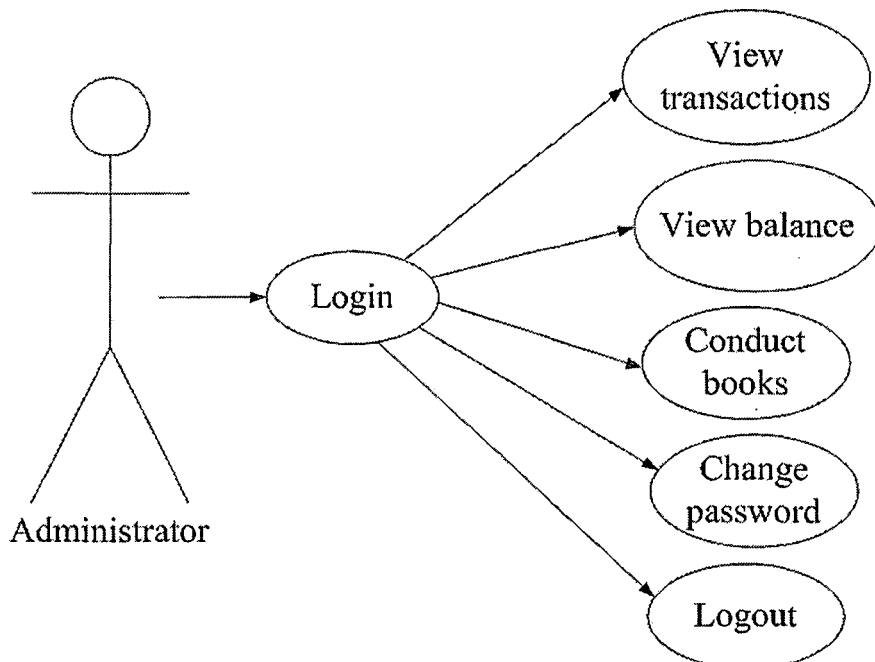


Figure 4. Use Case Model for Bookstore Administrator

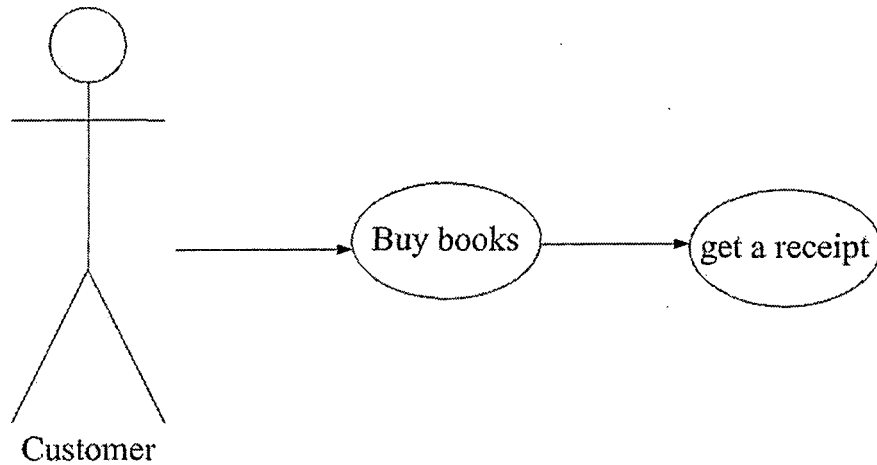


Figure 5. Use Case Model for Bookstore Customer

There are two types of receipt service users: customers and administrators. In this case, the bookstore administrator functions in the role of receipt service customer. The customer registers with the receipt service, which includes setting a password, userid and uploading a company logo to appear in the receipts. The customer can log in after registration to change userid, password or logo. The receipt service administrator views a list of transactions, and views the receipt service's balance of funds. Its customer's use case diagram is shown in Figure 6. Its administrator's use case diagram is shown in Figure 7.

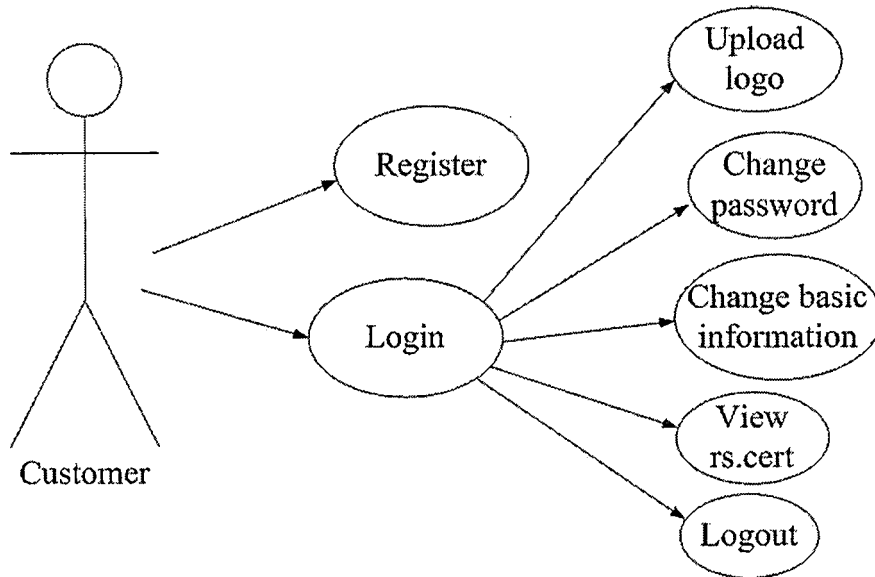


Figure 6. Use Case Model for Receipt Service Customer

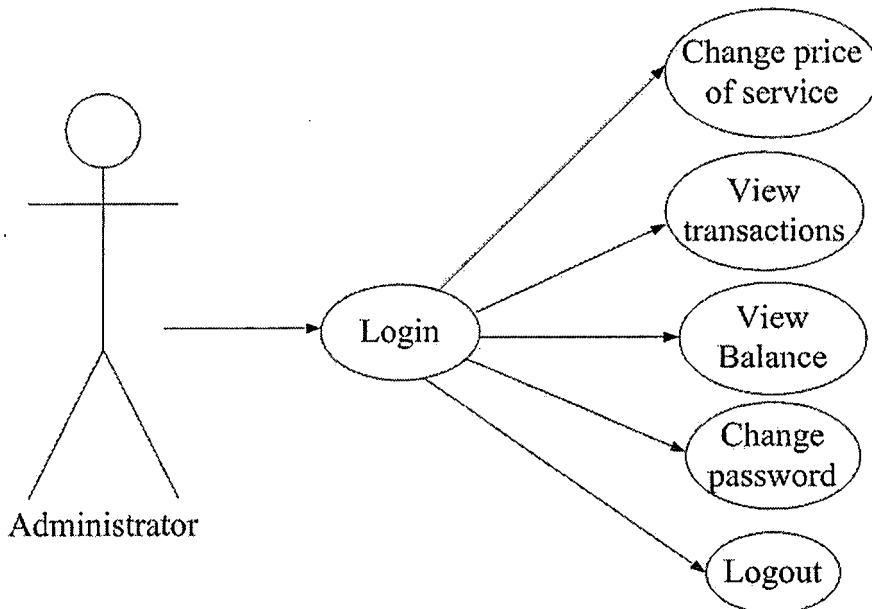


Figure 7. Use Case Model for Receipt Service Administrator

CHAPTER FIVE

DEFINITIONS

LCP - The Lightweight Currency Protocol enables entities to create currencies that can be used as a secure medium of exchange for low-value business transactions on the Internet.

HTML - Hyper-Text Markup Language is a Standard Generalized Markup Language (SGML) application conforming to International Standard ISO 8879.

XML - Extensible Markup Language is an initiative from the World Wide Web Consortium (W3C) defining an "extremely simple" dialect of SGML suitable for use on the World-Wide Web.

HTTP - Hypertext Transfer Protocol is a client-server Transmission Control Protocol over Internet Protocol (TCP/IP) used on the World-Wide Web for the exchange of HTML documents. It conventionally uses port 80.

Java Servlet - Java Servlet is a Java program that runs as part of a network service, typically an HTTP server and responds to requests from clients. The most common use for a servlet is to extend a web server by generating web content dynamically. For example, a client may need information from a database; a

servlet can be written that receives the request, gets and processes the data as needed by the client and then returns the result to the client.

JSP - Java Server Page is A freely available specification for extending the Java Servlet API to generate dynamic web pages on a web server. JSP assists developers in creating HTML or XML pages that combine static (fixed) page templates with dynamic content. Separating the user interface from content generation allows page designers to change the page layout without having to rewrite program code. JSP was designed to be simpler than pure servlets or CGI scripting.

RS - Receipt Service is this product's name

SQL - Structure Query Language is an industry-standard language for creating, updating and, querying relational database management systems. The first SQL standard, in 1986, provided basic language constructs for defining and manipulating tables of data; a revision in 1989 added language extensions for referential integrity and generalised integrity constraints. Another revision in 1992 provided facilities for schema manipulation and data administration, as well as substantial enhancements

for data definition and data manipulation.

Development is currently underway to enhance SQL into a computationally complete language for the definition and management of persistent, complex objects.

SSL - Secure Sockets Layer is a protocol designed by Netscape Communications Corporation to provide encrypted communications on the Internet. SSL is layered beneath application protocols such as HTTP, Simple Mail Transfer Protocol (SMTP), Telnet (an Internet standard protocol for remote login. Runs on top of TCP/IP), File Transfer Protocol (FTP), Gopher (a distributed document retrieval system), and Network News Transfer Protocol (NNTP) and is layered above the connection protocol TCP/IP. It is used by HyperText Transmission Protocol, Secure (HTTPS) access method.

Tomcat - Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed by Sun under the Java Community Process. Tomcat is developed in an open and participatory environment and released under the Apache Software

License. Tomcat is intended to be a collaboration of the best-of-breed developers from around the world.

CHAPTER SIX

SECURITY

Three machines are used in this project to run the server systems; Redhat Linux is the operating system installed on each machine. The machines are configured to be highly secure. For server protection, my security measures fall into three categories: security updates, firewall and secure remote access. To make my web services secure, I use SSL/TLS.

During the course of my project, I checked for security updates to critical systems by going to the Redhat website and checking for security updates. I installed the current security updates for the kernel, iptables, openssh and other packages that I installed. When a security update to a package was posted, I downloaded and installed the update.

For the firewall, I set up iptables to block TCP connection requests coming into the system to all ports other than ports 22 and 8080. I drop all UDP packets except those packages coming from the DNS servers and going to port 53. All other packets are accepted.

When accessing my systems remotely, I use SSH version 2.

The web services are secured through the use of SSL/TLS. All web service invocations are done over SSL. Client authentication is needed when the bookstore invokes a receipt generation service operation on the receipt service, otherwise an attacker can generate bogus receipts with the bookstore's logo. However, this security precaution is not very important, because an attacker can easily forge receipts using a logo taken from a bookstore receipt. Nevertheless, the bookstore authenticates using a password when it invokes a web service operation. The bookstore needs to install the receipt service's certificate in its trust store, so that the bookstore can establish authenticated secure SSL sessions with the receipt service. The LCP protocol is followed, which is secure through its use of TLS.

CHAPTER SEVEN
SYSTEM PROCESS

7.1 Preparation

Before starting the system, the necessary certificates must have been created in both of the bookstore and the receipt service. The following are the steps:

Step 1: Create keystore in bookstore

```
keytool -genkey \  
        -alias bs \  
        -keyalg RSA \  
        -keysize 1024 \  
        -dname "cn=yho.ias.csusb.edu" \  
        -keystore bs.keystore \  
        -keypass changeit \  
        -storepass changeit
```

Step 2: Create bs.cert in bookstore

```
Keytool -export \  
        -alias bs \  
        -file bs.cert \  
        -rfc \  
        -keystore bs.keystore \  
        -storepass changeit
```


Step 3: Create bookstore account in LCP by inputting balance and the content of bs.cert (shown as Figure 8).

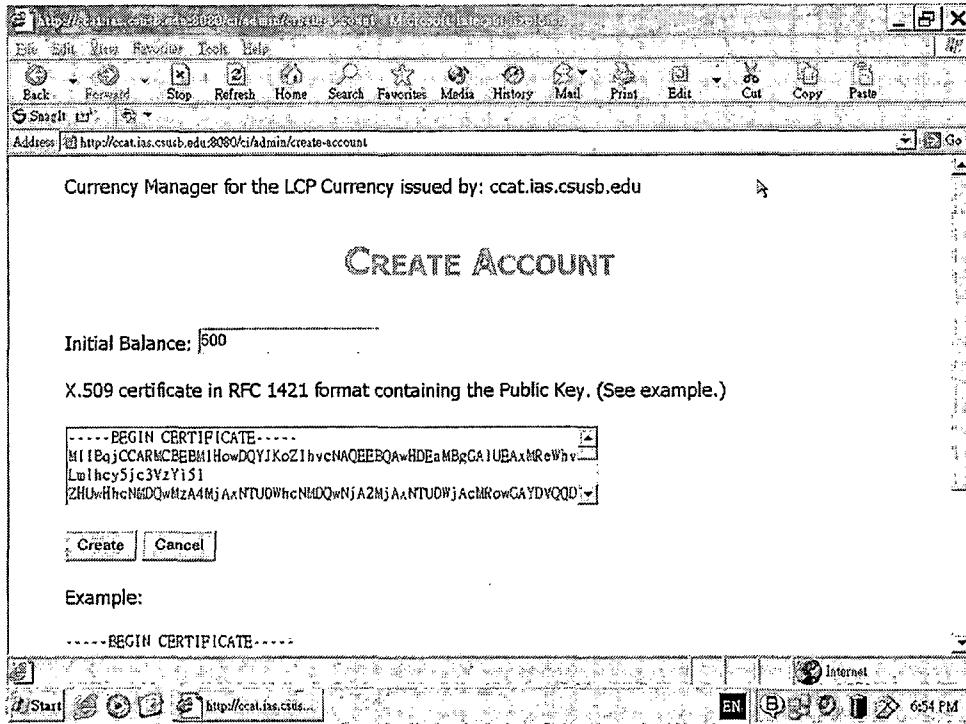


Figure 8. Create an Account in Currency Issuer

Step 4: Create ccat.cert in both rs and bookstore.

Copy the following lines (from -----BEGIN CERTIFICATE-----
to -----END CERTIFICATE-----)

-----BEGIN CERTIFICATE-----

MIIBrDCCARUCBEA+kJYwDQYJKoZIhvcNAQEEBQAwHTEbMBkGA1UEAxMSY2

NhdC5pYXMuY3N1c2Iu

ZWR1MB4XDTA0MDIyNzAwMzQzMFOXDTA1MDIyMTAwMzQzMFOwHTEbMBkGA1

UEAxMSY2NhdC5pYXMu

Y3N1c2IuZWR1MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDCO2MMMM

sjVYXdF9W+ML6Q7DWE
Kws8hl+ggVzvvh2vOpiq4AMsE9NnP2EiIMpx5oolpyXNyM1zuxPpP4dbR
5eg5LLNOUm6mlqau8H
Gyc81568tc5Xawf0FL2vKla/1AW1h8MsWhPsAactacWCfHERQkzGNjdaop
XDbnVcit5bCwIDAQAB
MA0GCSqGSIb3DQEBAUAA4GBAB8DG+HQsRSWb9W920sOjV/DWQ8NHrIYII
NsIF3iLjcBn3CqTylC
17RqqgahXRbb4cUGoRjwLXCoZwuPgnahb6Q6oALL/5rp6tLukRwwHWYh38
zsoyJfxXz+1EeX0rwy
WsX7F5CUuYrJ5hgvxHlV1fhQI+Yxl4hqLeuGRzQzS0Sm

-----END CERTIFICATE-----

Step 5: Create keystore in receipt service

```
Keytool -genkey \  
        -alias rs \  
        -keyalg RSA \  
        -keysize 1024 \  
        -dname "cn=yho.ias.csusb.edu" \  
        -keystore rs.keystore \  
        -keypass changeit \  
        -storepass changeit
```

Step 6: Create rs.cert in receipt service.

```
Keytool -export \  
        -alias rs \  
        -file rs.cert \  
        -storepass changeit
```

```

-rfc \
-keystore rs.keystore \
-storepass changeit

```

Step 7: Create rs.cert whose content in receipt service for bookstore (shown as Figure 9).

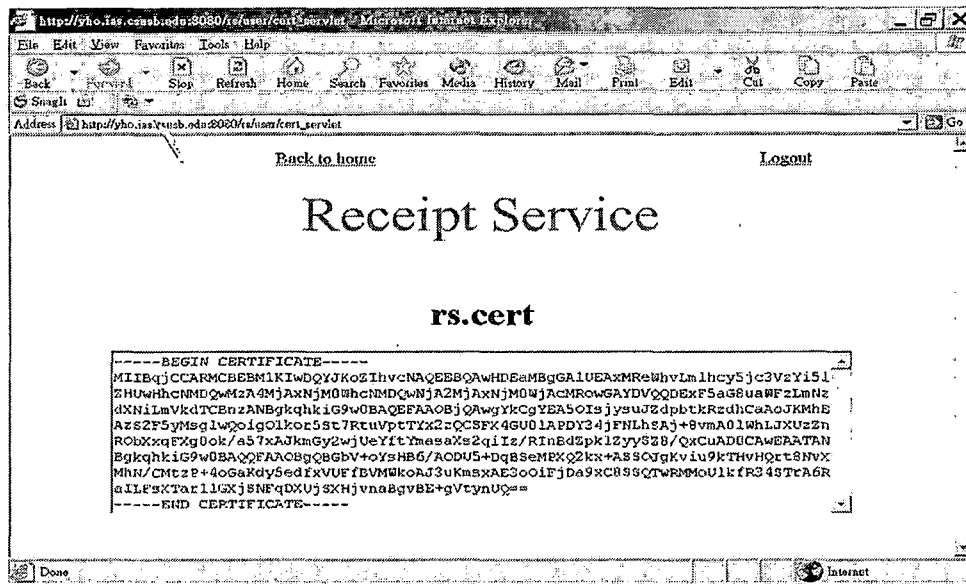


Figure 9. The Content in the Rs.cert

Step 8: Inspection.

Bookstore has

ccat.cert

bs.keystore

rs.cert

bs.cert (not needed in the transaction, just for creating account in Currency Issuer)

Receipt service has

cocat.cert

rs.keystore

rs.cert (not needed in the transaction, just for client to see)

7.2 Flowcharts for the Processing

The following flowcharts are the processing in the whole system, where the numbers are in time-ascending order. The bookstore will be called as BS and the receipt service will be called as RS in the flowcharts.

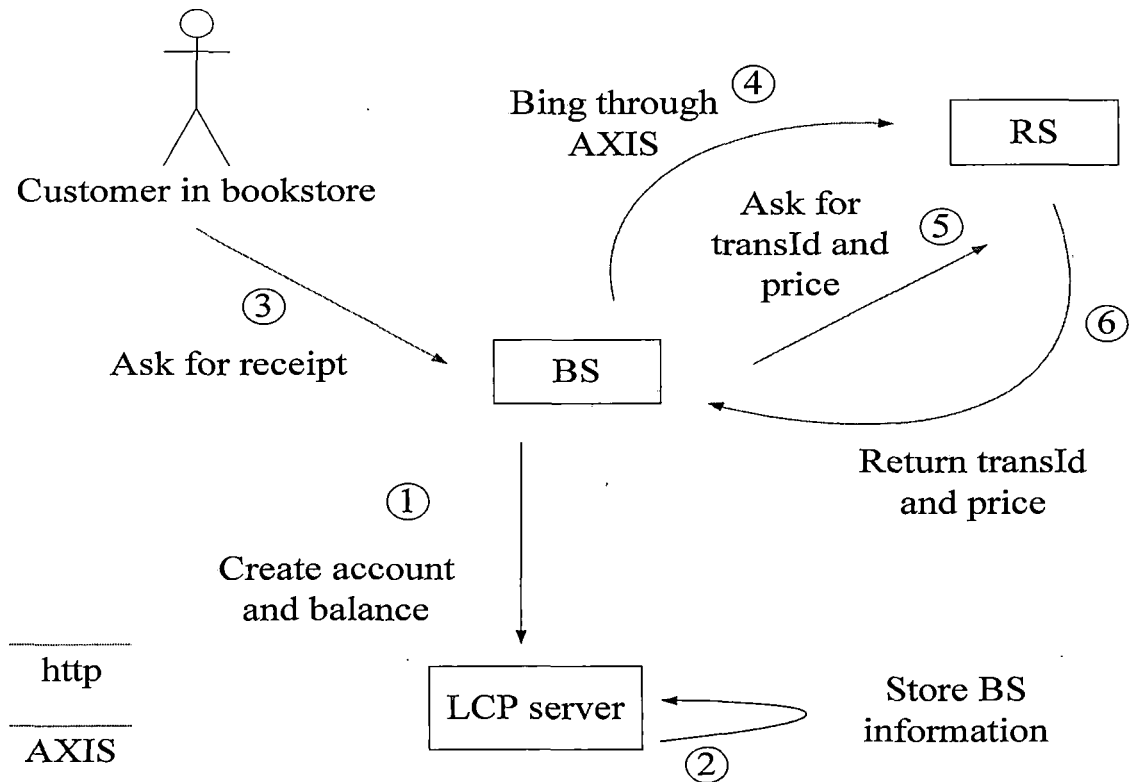


Figure 10. The First Part of Detail Processing

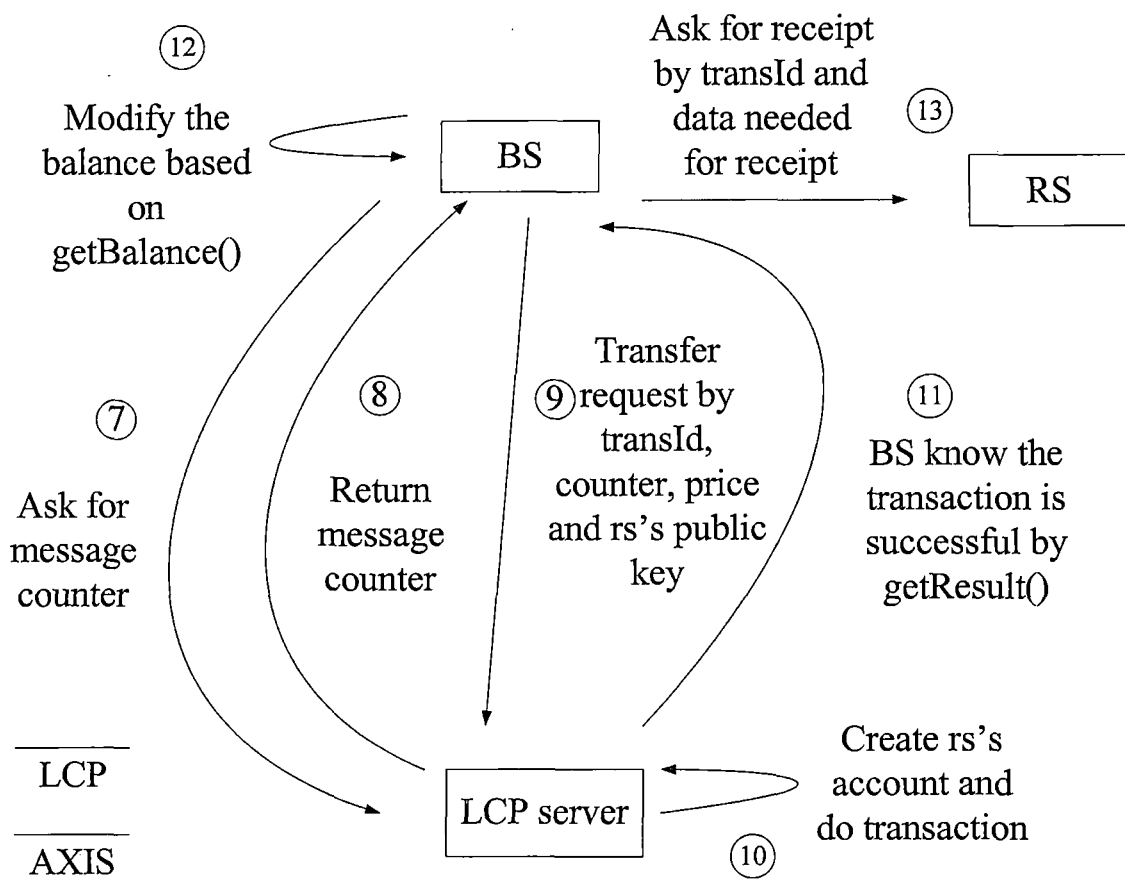
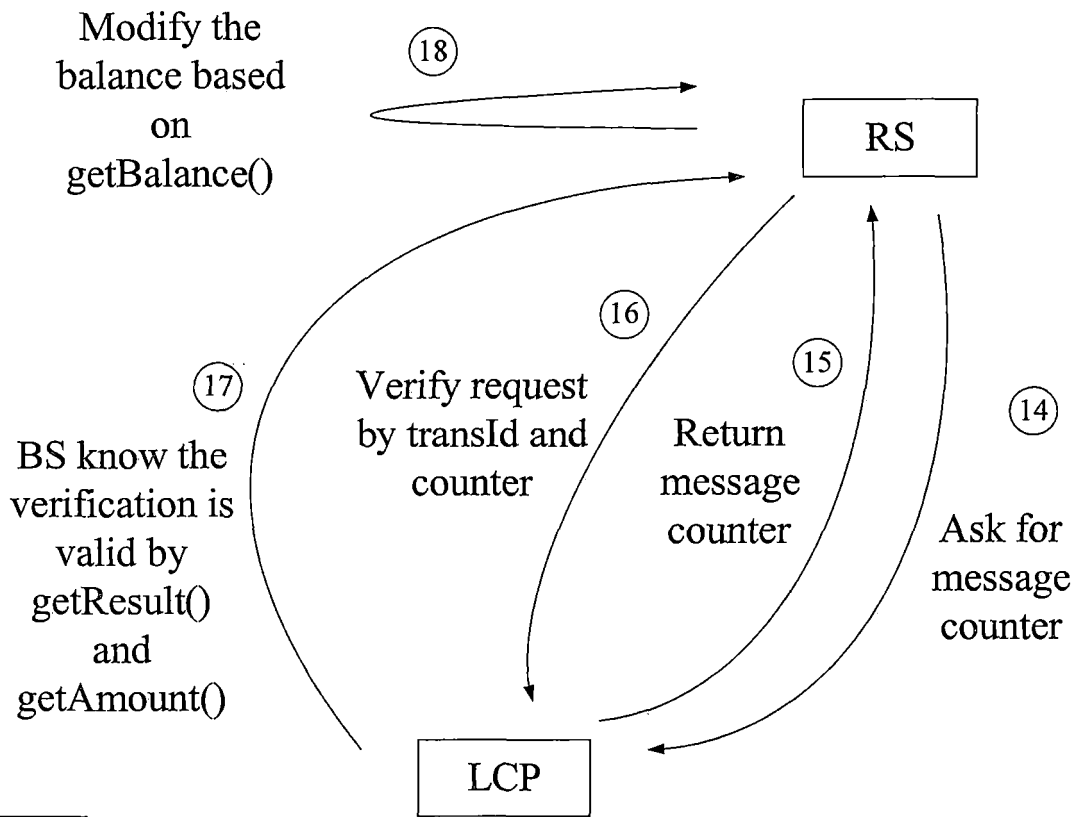


Figure 11. The Second Part of Detail Processing



LCP

Figure 12. The Third Part of Detail Processing

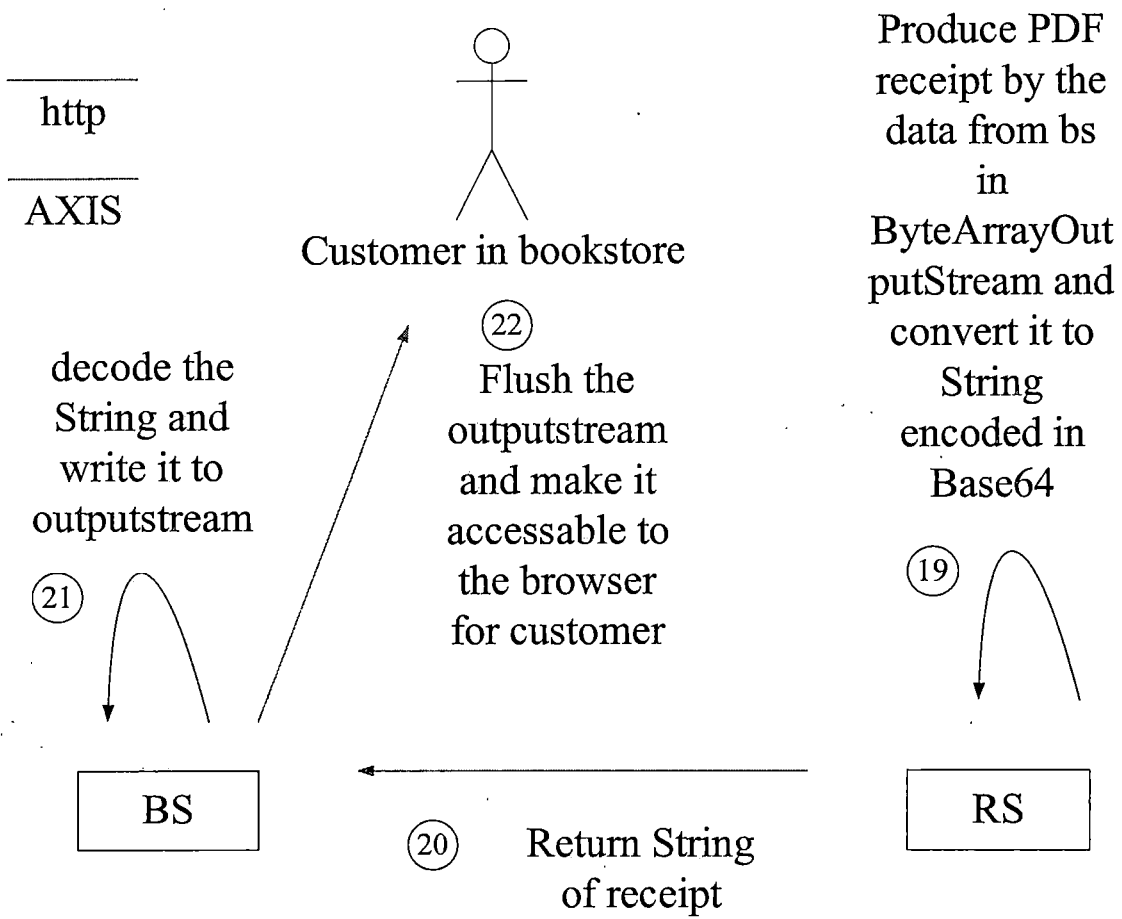


Figure 13. The Fourth Part of Detail Processing

7.3 Sequential Diagram

Figure 14 shows the system sequential diagram.

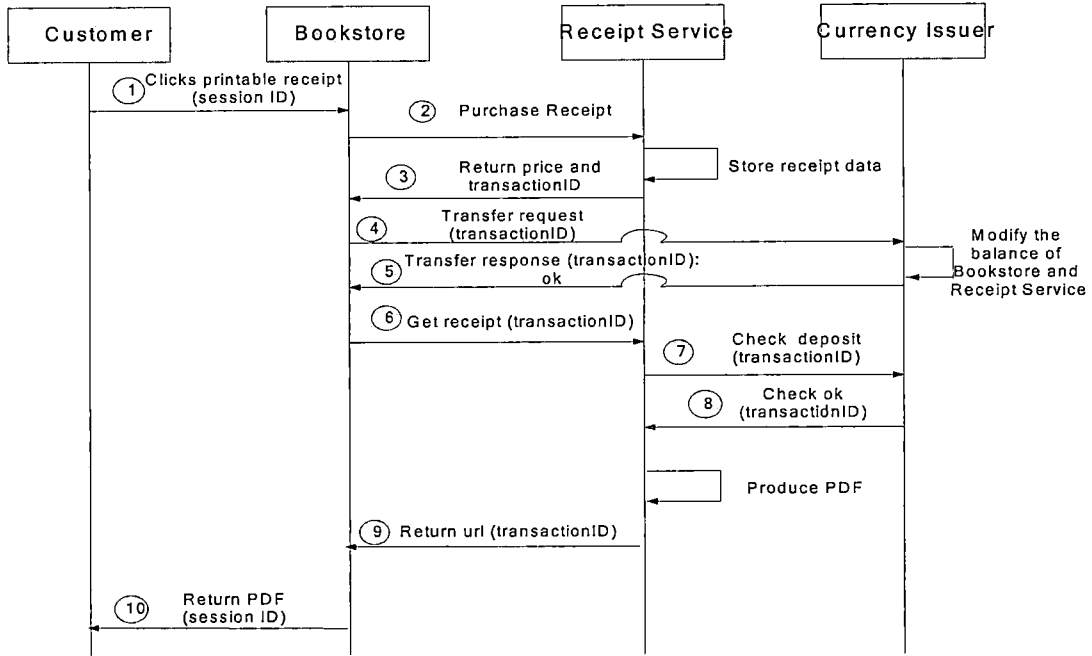


Figure 14. System Sequential Diagram

7.4 System Diagram and Brief Process

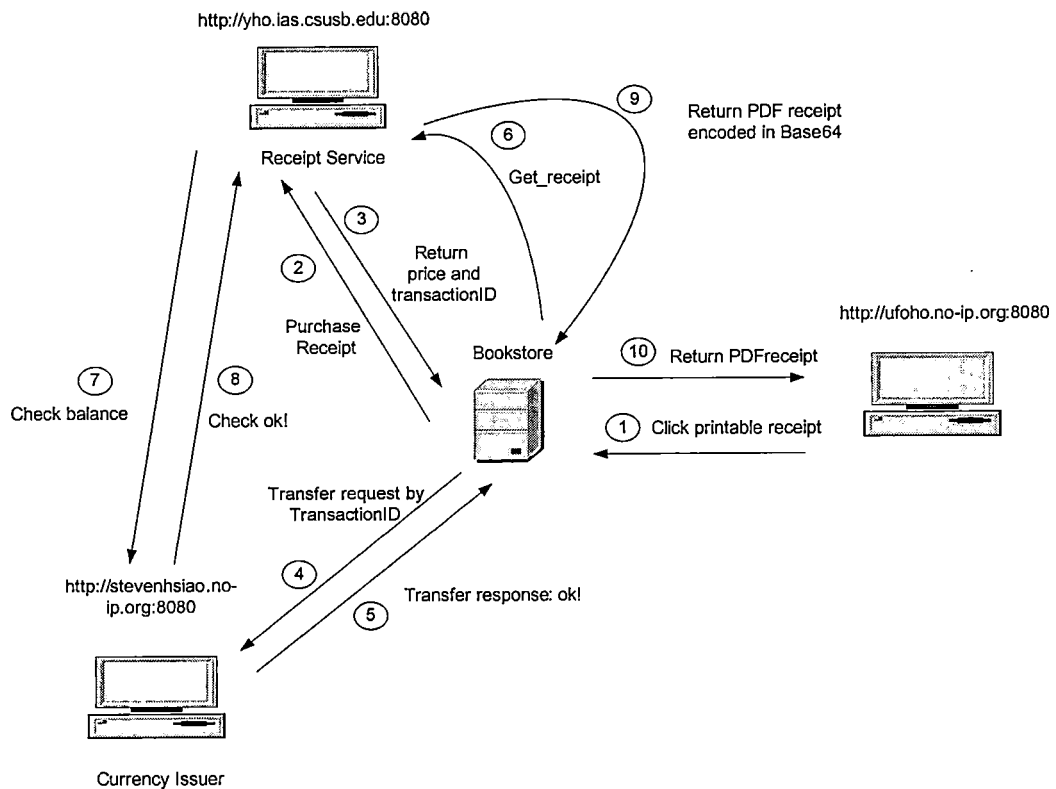


Figure 15. Whole System Diagram

① HTTP Request

From Browser (customer side) to Bookstore

HTTP Method = GET

(Content-type: text/html)

This is a Form data submission to send information for PDF.

② SOAP/HTTPS Request (Self-Signed Certificate)

From Bookstore to Receipt Service

HTTP Method = POST

(Content-type: text/xml)

This is a request to Receipt Service for purchasing the service of getting receipt with necessary data to construct receipt.

③ SOAP/HTTPS Response (Self-Signed Certificate)

From Receipt Service to Bookstore

HTTP Method = POST

(Content-type: text/xml)

This is a response for returning the price and transaction ID to proceed the transaction.

④ LCP Transfer Request

From Bookstore to Currency Issuer

LCP Method = transfer_request

This is a request to Currency Issuer for executing the transaction whose content is to decrease the balance of Bookstore in 10 units and increase the balance of Receipt Service in 10 units.

⑤ LCP Transfer Response: OK!

From Currency Issuer to Bookstore

LCP Method = transfer_response

This is a response for informing Bookstore that the transaction is successful. In other words, the balances

in Bookstore and in Receipt Service are decreased and increased in a certain amount respectively.

⑥ SOAP/HTTPS Request (Self-Signed Certificate)

From Bookstore to Receipt Service

HTTP Method = POST

(Content-type: text/xml)

This is a request to Receipt Service for getting its service including the data for PDF with transaction ID.

⑦ LCP Deposit Check Request

From Receipt Service to Currency Issuer

LCP Method = Verification_request

This is a request to Currency Issuer for checking if the payment from Bookstore is done

⑧ LCP Deposit Check Response: OK

From Currency Issuer to Receipt Service

LCP Method = Verification_response

This is a response for informing Receipt Service that the payment from Bookstore is done.

⑨ SOAP/HTTPS Response (Self-Signed Certificate)

From Receipt Service to Bookstore

HTTP Method = POST

(Content-type: text/xml)

This is a response for sending the PDF receipt encoded in Base64 to Bookstore.

⑩HTTP Response

From Bookstore to Browser (customer side)

Method = GET

(Content-type: application/pdf)

This is a response for the request from browser by sending PDF object back to the browser and make it display.

CHAPTER EIGHT
SYSTEM SCENARIO

8.1 Main Success Scenario

1. The user goes to the Bookstore web site by the browser.
2. The user buys one or more books.
3. The user checks out and asks for the PDF receipt.
4. The Bookstore records the Date, the session ID, the Customer ID (from User ID), his information for receipt and "printed" (initially false) by creating a new record in its receipt table.
5. The Bookstore transfers purchase request and the information for receipt to the Receipt Service.
6. The Receive Service stores the information.
7. The Receive Service generates a transaction ID.
8. The Receive Service creates a record with this transaction ID, and the Bookstore's name from the information, the date and the price for the service fee, and "printed" (initially false) in its Receipt table.
9. The Receive Service returns the price and transaction ID to the Bookstore.

10. The Bookstore stores the transaction ID into the record.
11. The Bookstore transfers request for the transaction to the Currency Issuer with transaction ID.
12. The Currency Issuer finishes the transaction which decreases the balance of the Bookstore and increases the balance of the Receive Service in 10 units respectively.
13. The Currency Issuer transfers ok response to the Bookstore with transaction ID.
14. The Bookstore transfers request of getting receipt to the Receipt Service with transaction ID.
15. The Receipt Service transfers request to the Currency Issuer for checking payment with transaction ID from the Bookstore.
16. The Currency Issuer seeks transaction ID.
17. The Currency Issuer returns ok message to the Receipt Service with transaction ID.
18. The Receipt Service produces the PDF receipt.
19. The Receipt Service returns the PDF receipt encoded in Base64 with transaction ID to the Bookstore.
20. The Bookstore decodes the binary codes in Base64 for the PDF receipt.

21. The Receipt Service sets the "printed" to be "true" of this record in the receipt table.
22. The Receipt Service returns the PDF receipt and transaction ID to the Bookstore.
23. The Bookstore sets the "printed" to be "true" of this record in the receipt table.
24. The Bookstore returns the PDF receipt to the browser by the session ID.

8.2 Fail Scenario in the Bookstore

1. Fail to get SoapBindingStub
Cannot bind to AXIS !
2. binding.get_idprice(). substring(0,
5).equals("Sorry") cannot get the price from rs !
3. get_price()(decoded string from rs) > = 2.0
the price for producing receipt is too high !
4. issuer.getCounter() fail (try once more after 30
minutes) the connection with LCP fail
5. ! GetCounterResponse.getResult().equals("success")
The response is invalid
6. Issuer.transfer fail
the request of transferring failed
7. !TransferResponse.getResult().equals("success")
The response is invalid

8. `User.set_balance(tresp.getBalance())` fail
cannot change the balance
9. `binding.getReceipt` fail
Receipt Service cannot be connected
10. `encoded_pdf.substring(0, 5).equals("Sorry")`
Problem occurred in producing PDF receipt

8.3 Fail Scenario in the Receipt Service

1. `PriceDatabase.get_price()` fail
Database cannot be accessed
2. `issuer.getCounter()` fail (try once more after 30 minutes) the connection with LCP fail
3. `!GetCounterResponse.getResult().equals("success")`
The response is invalid
4. `Issuer.verify` fail
the request of verification failed
5. `!VerifyResponse.getResult().equals("success")`
The response is invalid
6. `PriceDatabase.get_price()` fail
Database cannot be accessed
7. `veresp.getAmount() < receipt_price`
the amount of payment from bookstore is not enough
8. `set_balance(veresp.getBalance())` fail
cannot' change the balance

9. Error occurs in creating ByteArrayOutputStream
cannot produce PDF Receipt
10. transacDatabase.addTransc(transaction) fail
Database cannot be accessed

CHAPTER NINE
LIGHTWEIGHT CURRENCY PROTOCOL AND
AXIS APPLICATION

9.1 Lightweight Currency Protocol Application

The Lightweight Currency Protocol (LCP) is a framework for creating currencies to be used by applications that rely on market forces for the allocation of scarce resources, or rely on incentives for the creation and operation of desirable services that would be otherwise impractical or infeasible through the use of real-world currencies. Lightweight currency was conceived as a means to enable the peer-to-peer resource market - a market for computer systems to sell their unused bandwidth, storage and compute power. End user applications use earned funds to purchase services built on top of this pool of distributed resources. Thus, applications may also earn money by selling derived services. Lightweight Currency is called lightweight, because it avoids the risk, complexity and cost of real-world currencies. It is thus suitable as a medium of exchange for low-value, unreliable resources provided through groups of collaborating, unrelated end systems.

Hence, the bookstore and the receipt service can really benefit from LCP throughout proving that in the project.

9.2 AXIS Application

Apache Axis is an implementation of the SOAP ("Simple Object Access Protocol"). SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses. Hence, in order to invoke the remote service located in the receipt service, the bookstore must go through AXIS. Also, once the receipt is produce in the receipt service, the receipt must be sent to remote website, the bookstore, through AXIS. The important thing to finish this is that the deployment in the receipt service must have been done first and the bookstore access the corresponding WSDL file to get the stub files of the original functions in the receipt service. In this project, the connection in receipt service will keep active after it is established.

CHAPTER TEN

USER MANUAL

10.1 Manual for the Bookstore

10.1.1 Functionality Common to All Users

Figure 16 is the starting page with animated effects to attract customers to get inside. After the user clicks the button "enter," he or she will see the main page of the bookstore. Figure 17 is this main page.

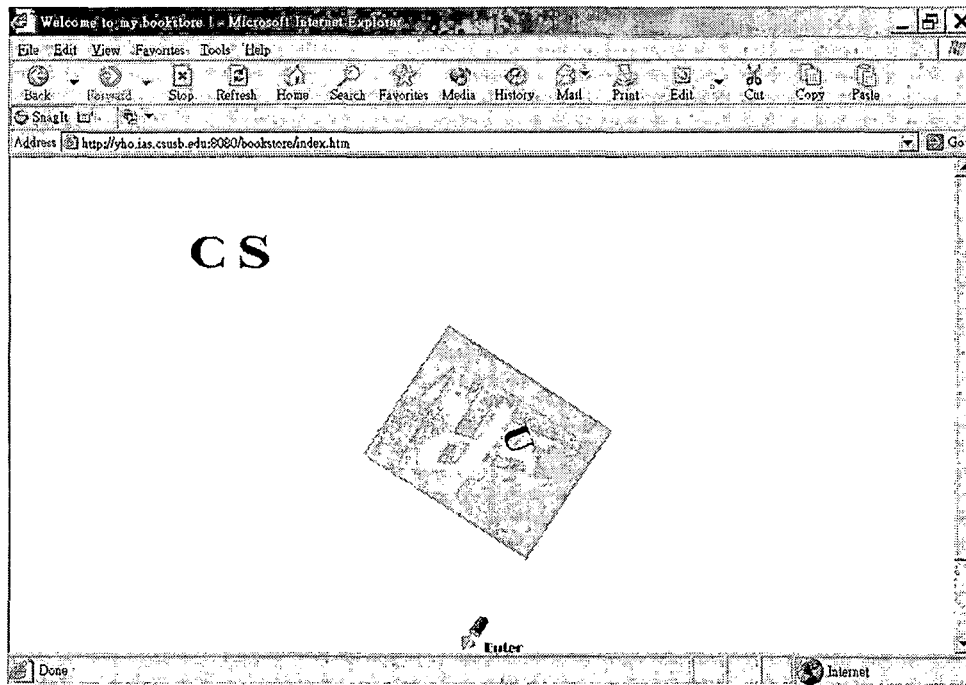


Figure 16. Starting Page in the Bookstore

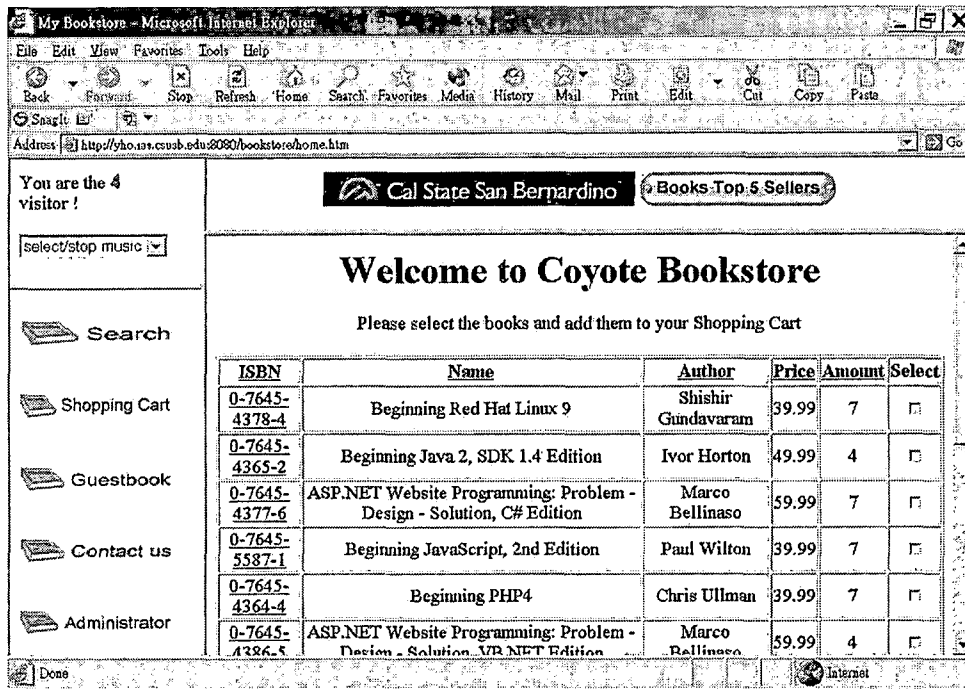


Figure 17. Main Page in the Bookstore

10.1.2 Customer Functionality

In the following pages, there is always a hyperlink, "keep shopping" on the top of each page for the customer to go back to the main page. After the customer chooses one or more books to buy and then submits the button "add to cart," he or she will see the cart page. Another way to get into the cart page is by clicking the button "Shopping cart" in the left menu so that the customer can check the status of the shopping cart at any time. Figure 18 is an instance of that page. If the customer does not like one or more books inside the shopping cart, he or she can remove one of them each time by clicking the hyperlink of

"remove" located after the corresponding book. If the shopping cart is empty, the page will be like Figure 19. Once the customer decides to check out, he or she can click the button of "Check out."

If that button is clicked, a SSL notice, shown in Figure 20, will show up and it means the following page is secured in SSL. The reason for encrypting the page is that the following page is for the customer to input private information which should be protected. If the customer leaves the page, this notice will show up again to inform him or her that the following page is not secured. The data is needed to accomplish the payment of ordered books. This page with a form is shown in Figure 21.

You are the 4 visitor !

select/stop music ▾

Search

Shopping Cart

Guestbook

Contact us

Administrator

Cal State San Bernardino Books Top 5 Sellers

[Keep Shopping](#)

Information of your shopping cart

Please remove disliked books from the list and then submit your order

ISBN	Name	Author	Price	Remove Item
0-7645-4378-4	Beginning Red Hat Linux 9	Shishir Gundavaram	39.99	Remove
Your total cost			39.99	

Figure 18. Cart Page

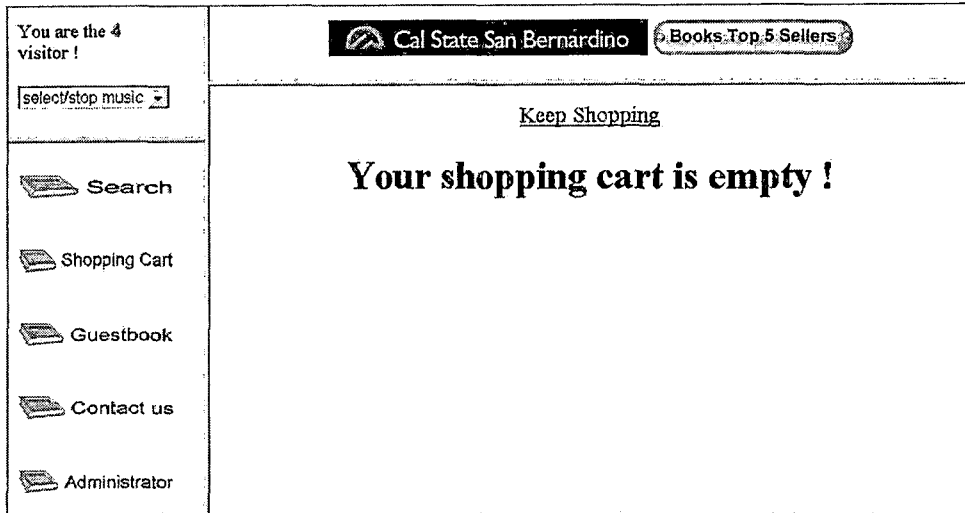


Figure 19. Empty Shopping Cart

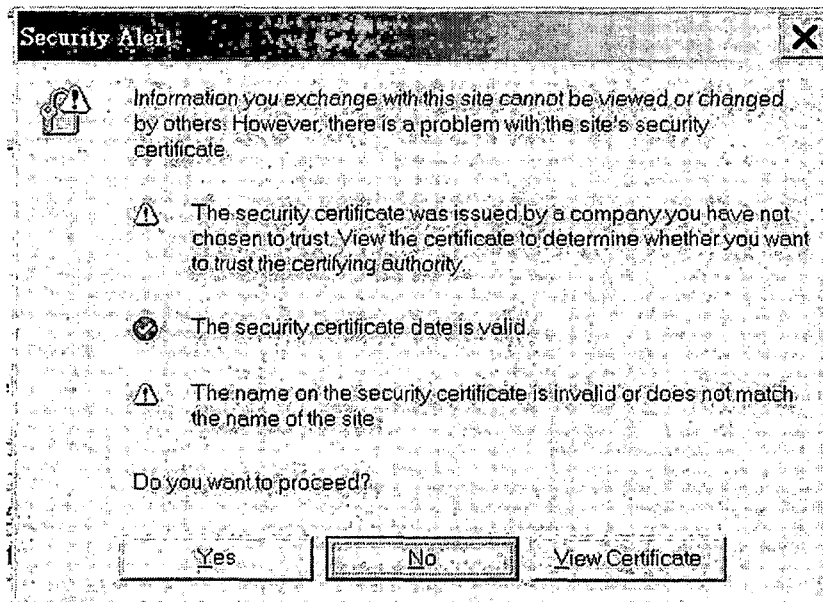


Figure 20. SSL Notice

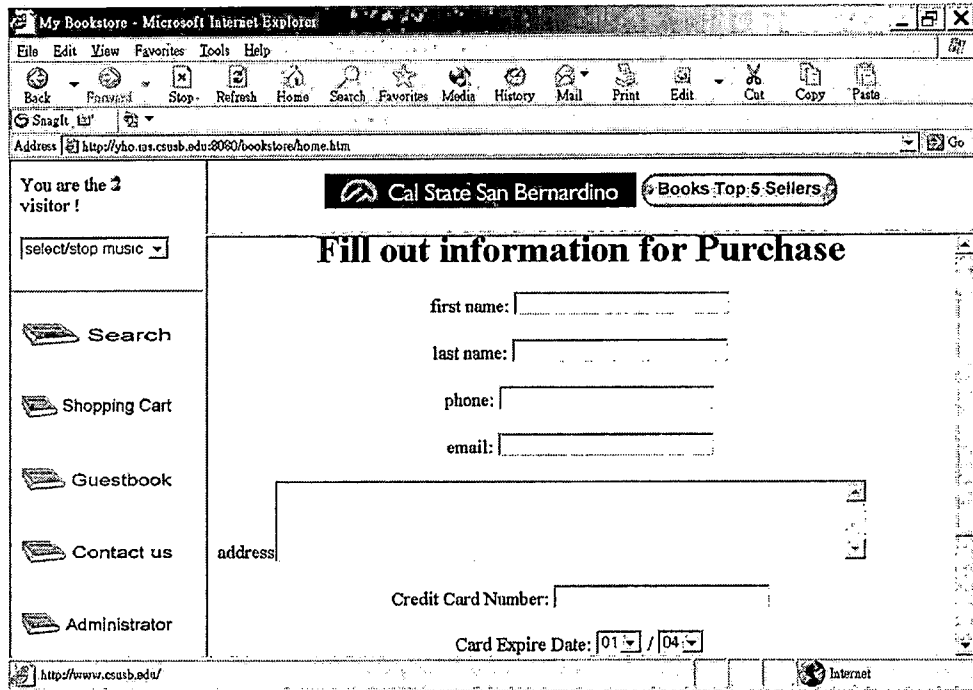


Figure 21. Form Page

If the customer regrets buying those books at this moment, he or she can click the button of "cancel" to go back to the cart page. After the button of "buy" is clicked and the format of inputted data is correct, the customer will see the result page, shown in Figure 22. In this page, the customer can see the information of his or her payment and ask for a receipt. If the customer clicks the button of "No," he or she will see the main page. If the customer clicks "text mode," he or she will see the text receipt page, shown in Figure 23. In this page, the customer can click the button of "Print receipt" to print out this whole page. If the customer clicks the button of

"PDF model," he or she will get a dialog box of asking for downloading a receipt file in PDF. Figure 24 is an instance of the PDF file.

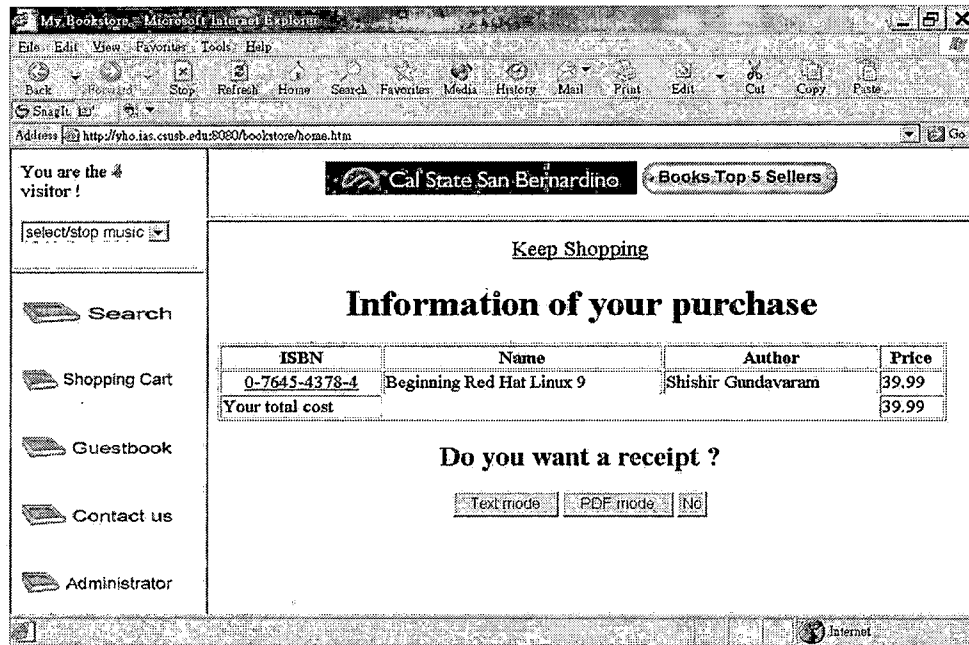


Figure 22. Result Page

My Bookstore - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites Media History Mail Print Edit Cut Copy Paste

Address http://yho.ias.csusb.edu:8080/bookstore/home.htm

You are the 3 visitor!

select/stop music

Search

Shopping Cart

Guestbook

Contact us

Administrator

Cal State San Bernardino Books.Top 5 Sellers

Print Receipt

COYTE
B o o k s

Serial	ISBN	Name	Author	Price
1	0-7645-4378-4	Begining Red Hat Linux 9	Shishir Gundavaram	39.99

Total Price: 39.99

Customer Name: Ho Yu-Feng

Purchase Date: Thu Apr 22 16:29:31 PDT 2004

Customer Phone: 8839792

Customer Email: ufoho@hotmail.com

Done Internet

Figure 23. Text-mode Receipt Page

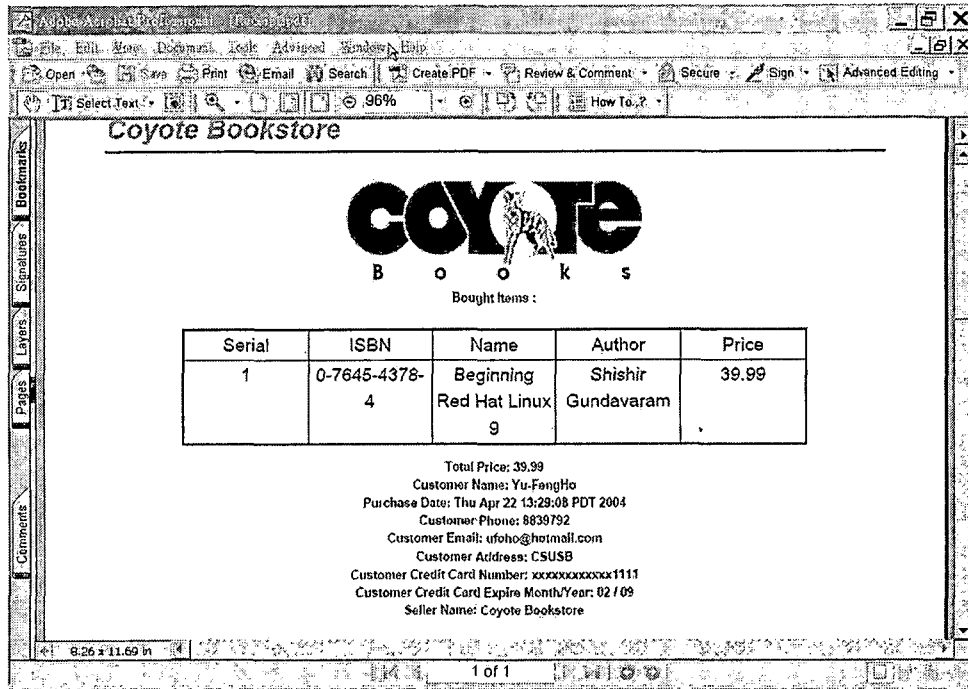


Figure 24. Receipt in PDF Format

The customer can click the button of "Search" in the left menu to search the desired books. Once the button is clicked, the customer will see the search page, shown in Figure 25. There are four options: name, author, lower bound and upper bound to limit the scope of the search. At least one of the options must be inputted with some content for searching. After inputting the content and clicking the button of "search," the customer will see the search result page, shown in Figure 26. The customer can add the books from the books in the result table into the shopping cart.

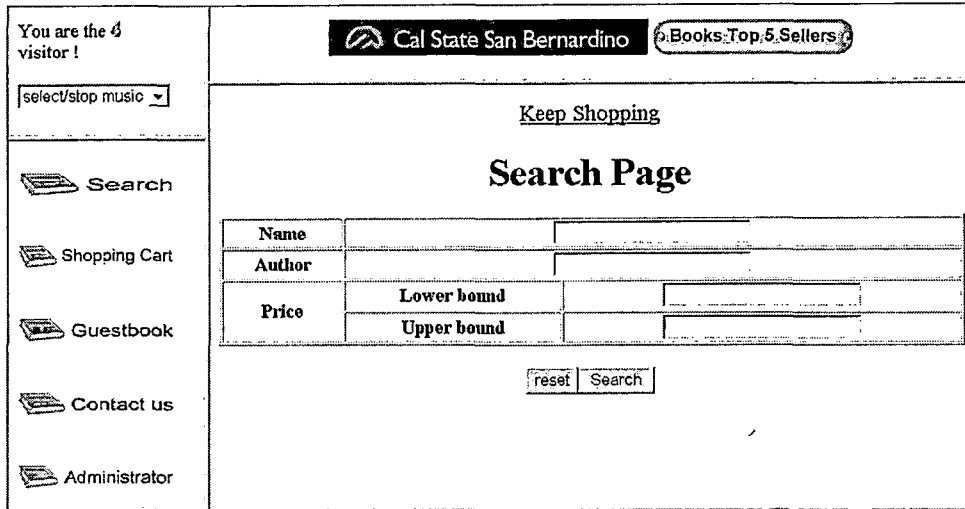


Figure 25. Search Page

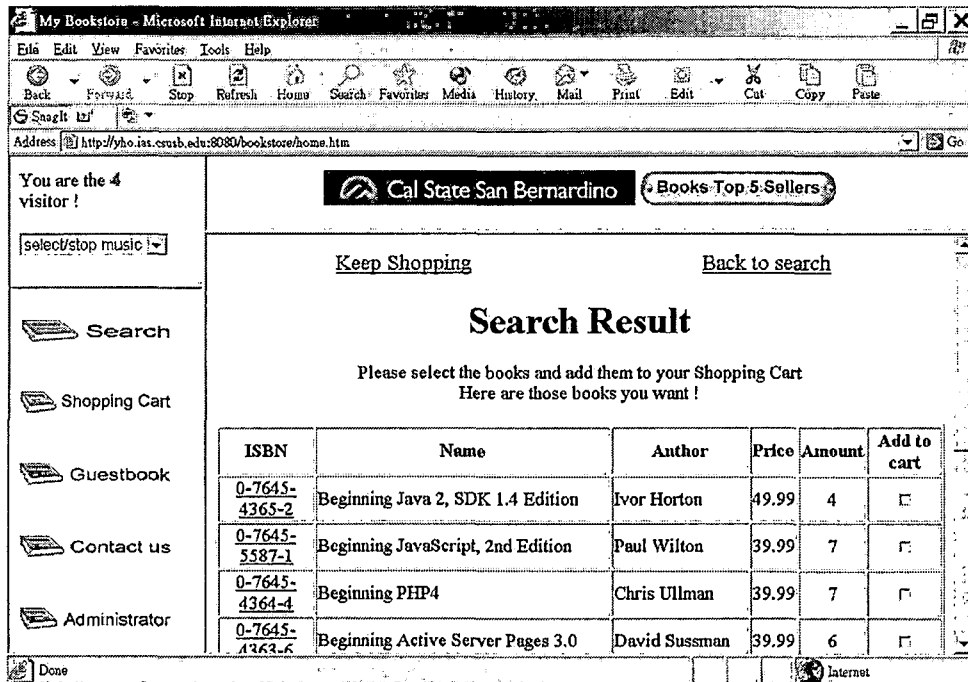


Figure 26. Search Result Page

The customer can click the button of "guestbook" in the left menu for presenting his or her feedback to the bookstore. After clicking, the customer will see the guestbook page, shown in Figure 27. The customer also can reply to others' feedback by clicking the corresponding "reply" hyperlink. After that, the customer will see the reply page, shown in Figure 28. Any dirty words inside the contents customers input are not allowed, so the original page will show with the dirty words replaced with "[dirty word]" if that happens. This situation is shown in Figure 29.

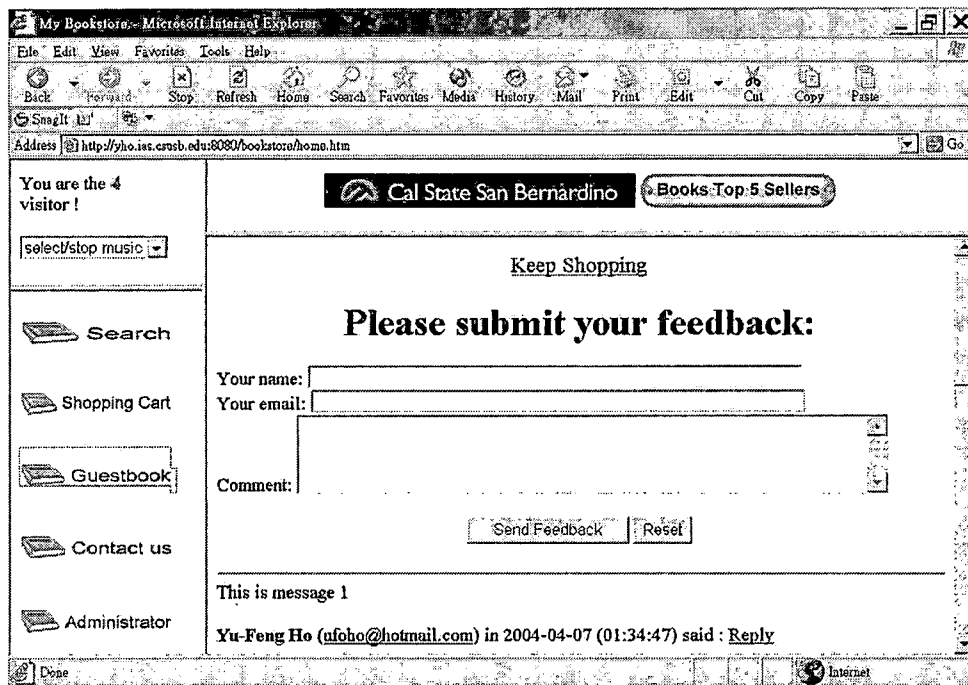


Figure 27. Guestbook Page

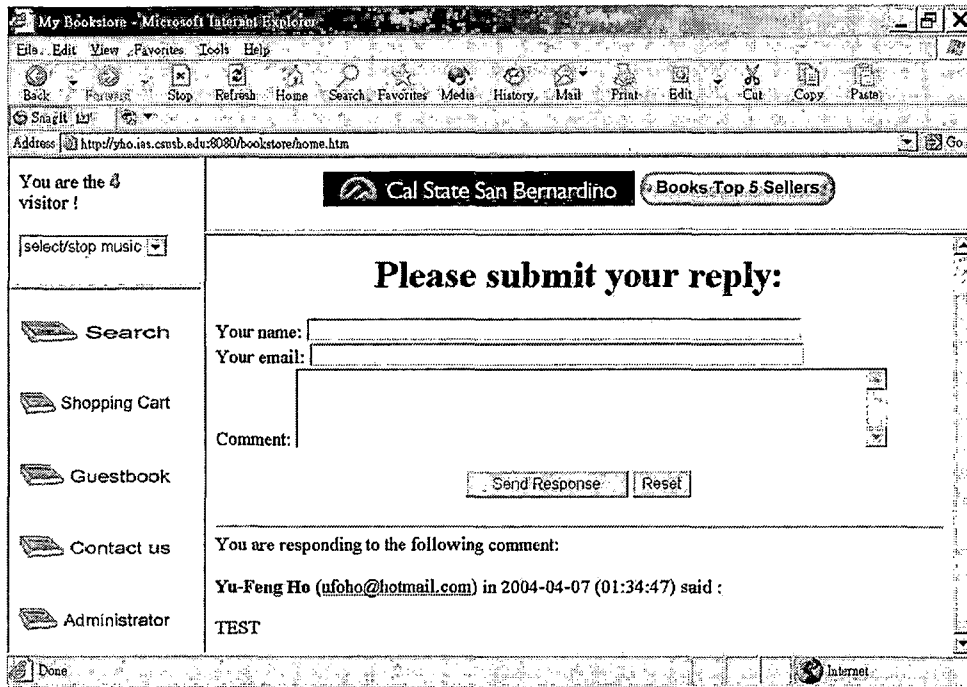


Figure 28. Reply Page

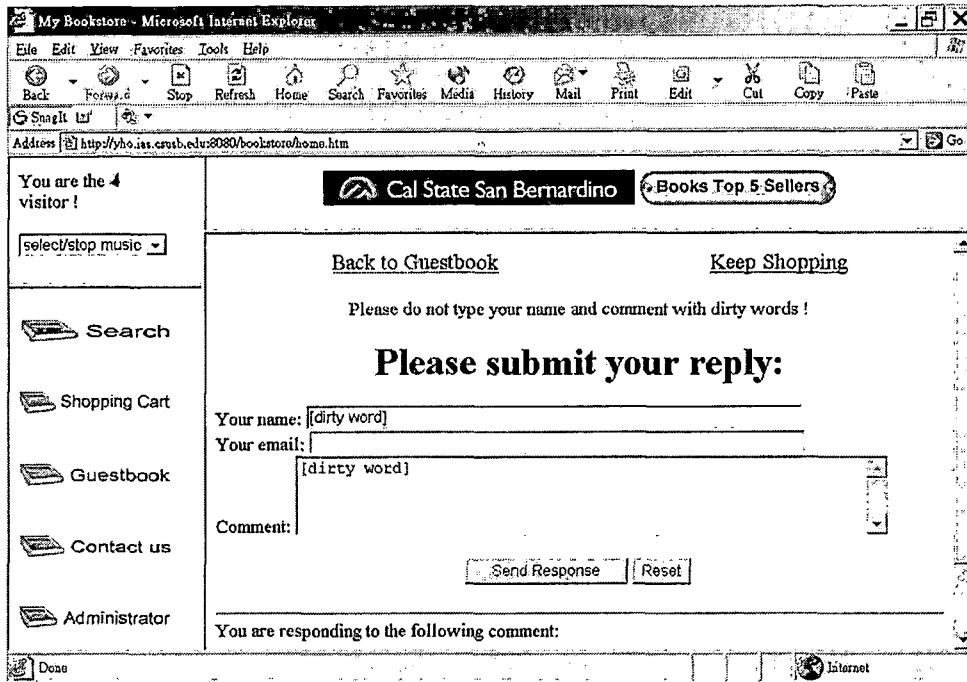


Figure 29. Page Responding to Dirty Words

The customer can submit the question by the bookstore's email system by clicking the button of "contact us" in the left menu. After clicking, the customer will see the contact page, shown in Figure 30. The contents the customer writes will be sent to the administrator's email box.

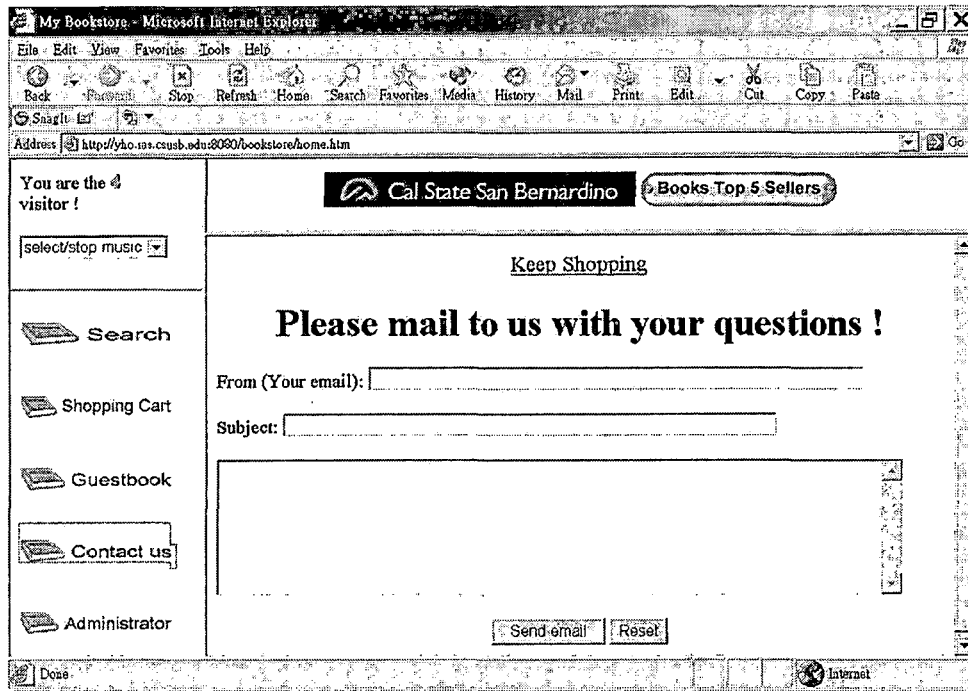


Figure 30. Contact Page

10.1.3 For the Administrator

The bookstore's administrator can administrate the bookstore by getting into the login page, shown in Figure 31, and inputting correct username and password. After logging in successfully, the administrator will see the admin page, shown in Figure 32. The administrator can check the transaction by clicking the "View transaction" hyperlink. Then, the administrator will see the transaction page, shown in Figure 33. In that page, the administrator can see the selling chart by clicking the "View the status of transactions" hyperlink. Then, the

administrator will see the chart in the following page, shown in Figure 34.

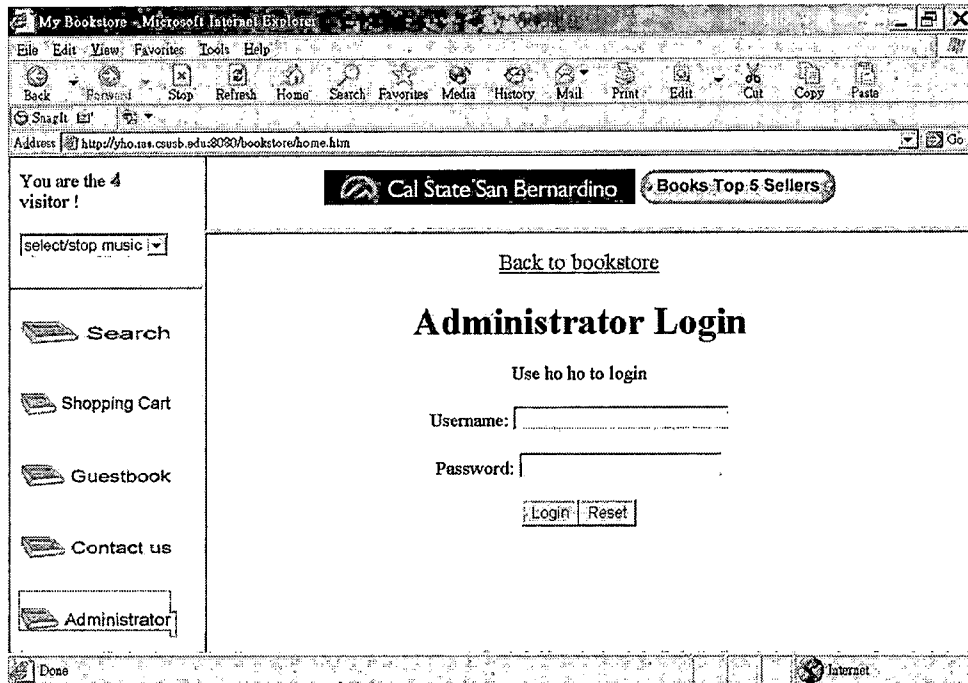


Figure 31. Login Page



Figure 32. Admin Page

My Bookstore - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites Media History Mail Print Edit Cut Copy Paste

Address http://yho.ist.csusb.edu:8080/bookstore/home.htm

You are the 4 visitor!

select/stop music

Search

Shopping Cart

Guestbook

Contact us

Administrator

Cal State San Bernardino Books Top 5 Sellers

[Back to home](#) [view the statistic of transactions](#) [Logout](#)

Transaction Information

Serial	ISBN	Bookname	Author	Price	Receipt	Date
1	0-7645-4365-2	Beginning Java 2, SDK 1.4 Edition	Ivor Horton	49.99	true	2004-03-30
2	0-7645-4363-6	Beginning Active Server Pages 3.0	David Sussman	39.99	true	2004-03-30
3	0-7645-4386-5	ASP.NET Website Programming: Problem - Design - Solution, VB.NET Edition	Marco Bellinaso	59.99	true	2004-03-30
4	0-7645-4364-4	Beginning PHP4	Chris Ullman	39.99	true	2004-03-30
5	0-7645-4377-6	ASP.NET Website Programming: Problem - Design - Solution, C#	Marco Bellinaso	59.99	true	2004-03-30

Internet

Figure 33. Transaction Page

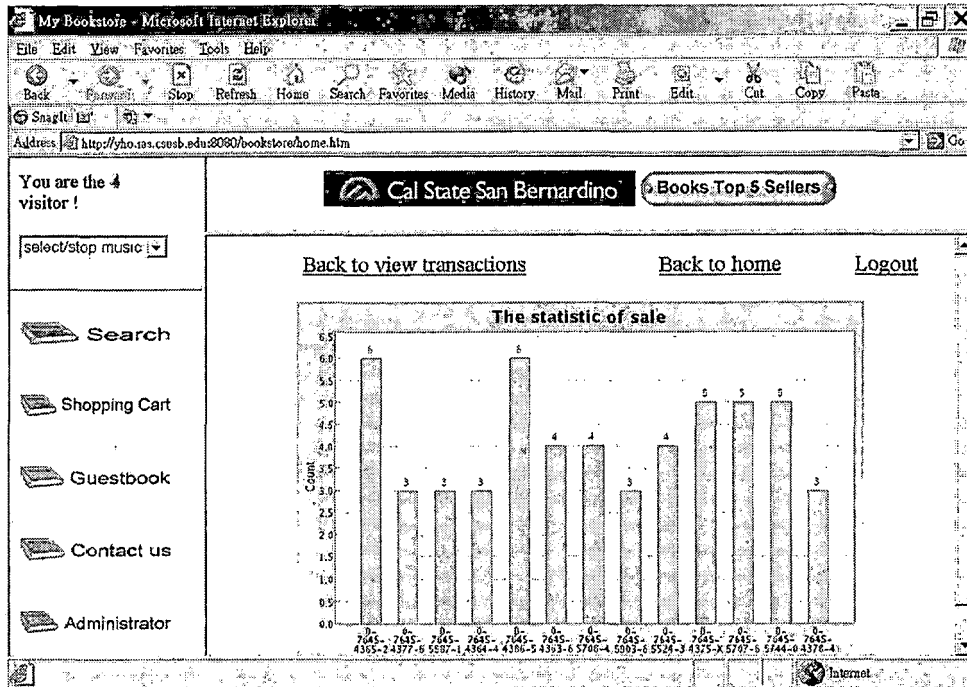


Figure 34. Chart Page

The administrator can see the current balance by clicking the "View balance" hyperlink in the admin page. Then, the administrator will see the balance page, shown in Figure 35. The administrator can add a book each time by clicking the "Add a book" hyperlink in the admin page. Then, the administrator will see the add book page, shown in Figure 36. The administrator can remove one or more books by clicking the "Remove books" hyperlink in the admin page. Then, the administrator will see the remove book page, shown in Figure 37.

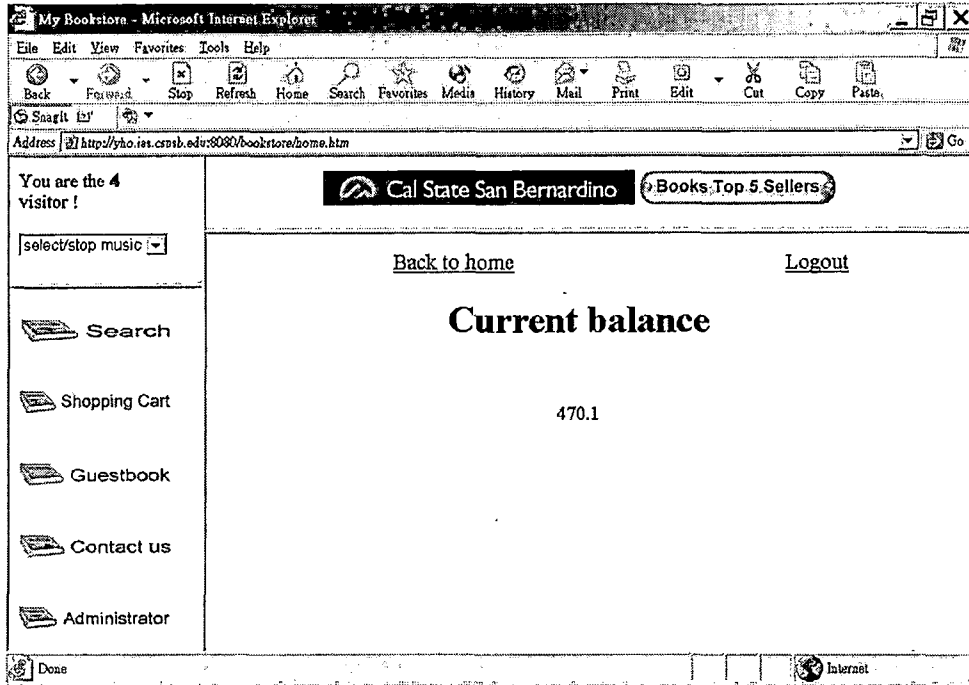


Figure 35. Balance Page

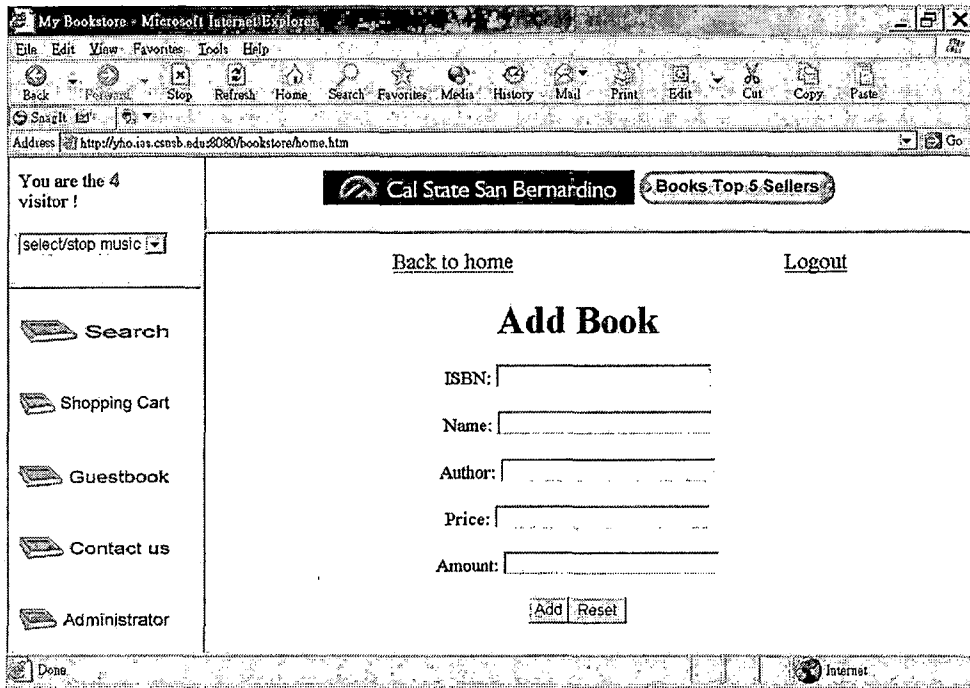


Figure 36. Add Book Page

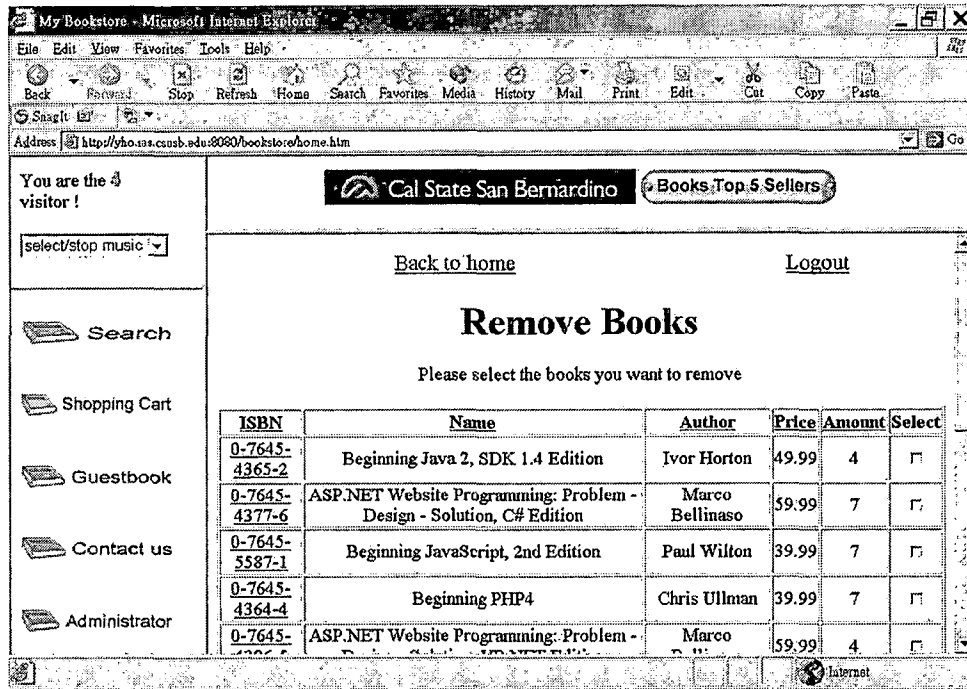


Figure 37. Remove Book Page

The administrator can modify the information of a book each time by clicking the "Modify a book" hyperlink in the admin page. Then, the administrator will see the modify book page, shown in Figure 38. After getting into that page, choosing the book to modify and clicking the button of "Select," the administrator will see the modify detail page, shown in Figure 39. In that page, the administrator can modify the data belonging to that book and replace (or upload) its picture by clicking the "replace picture" hyperlink. If clicking the hyperlink, the administrator will see the upload page, shown in Figure 40, for uploading the image file in jpg format. The

administrator can delete all the books' information by clicking the "Delete all books" in the admin page. Then, the administrator will see the confirm page, shown in Figure 41, to decide whether to do so or not. The administrator can add the default data of books into the database by clicking the "Add the default books" hyperlink in the admin page. The administrator can logout by clicking the "logout" hyperlink in any one of the pages belonging to the administrator.

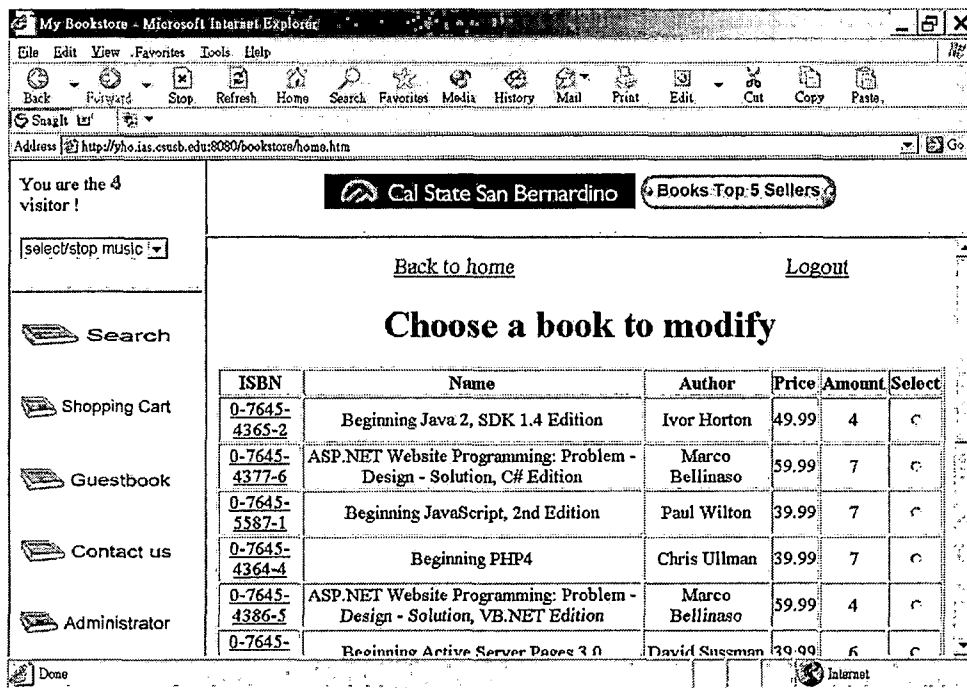


Figure 38. Modify Book Page

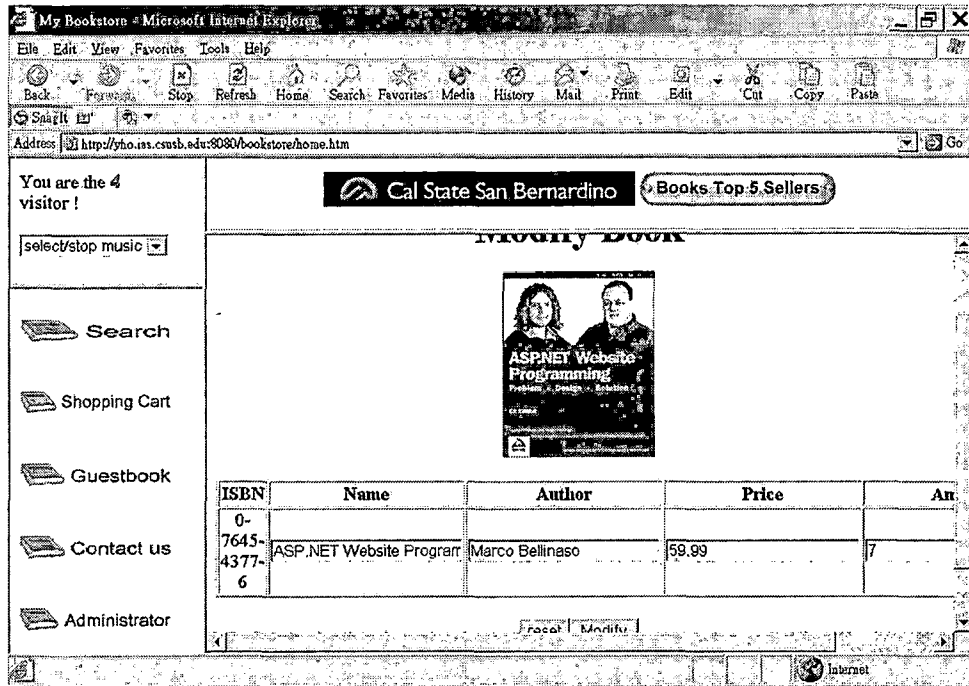


Figure 39. Modify Detail Page

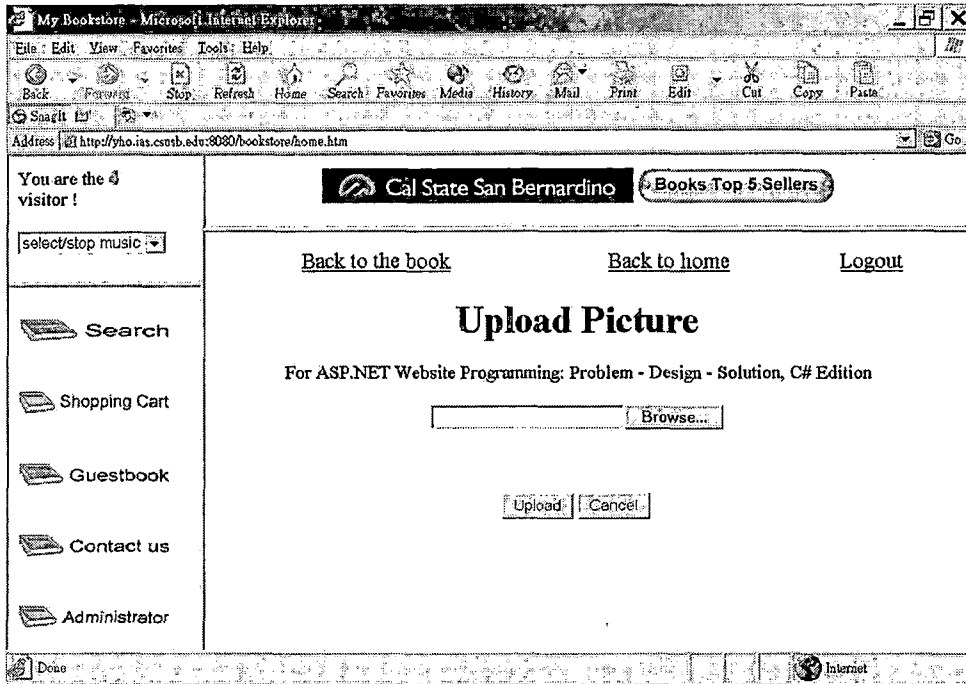


Figure 40. Upload Page

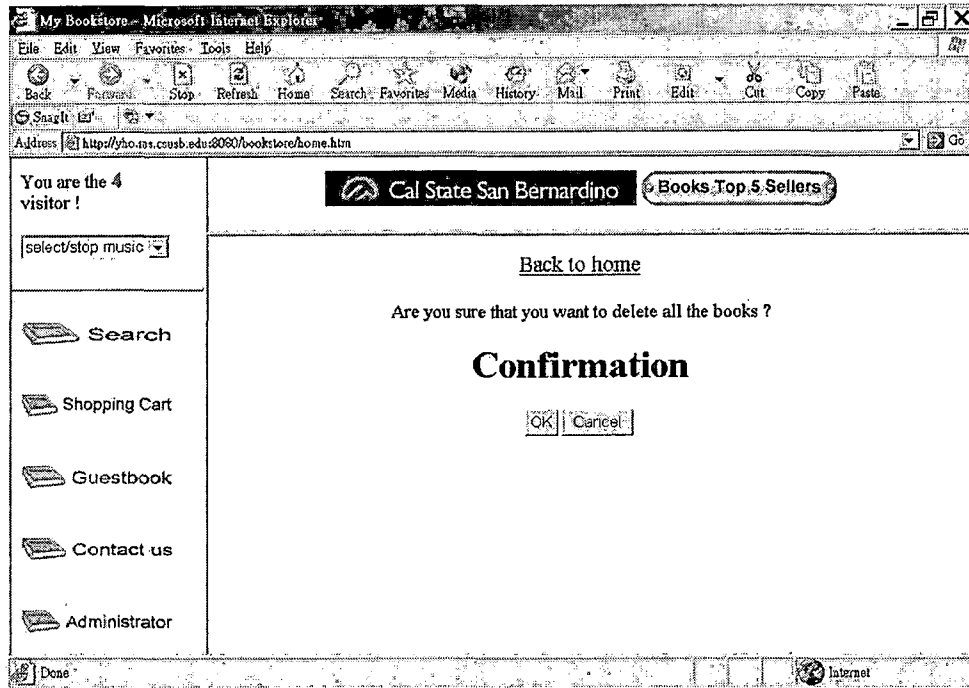


Figure 41. Confirm Page

10.2 Manual for the Receipt Service

10.2.1 Common Functionality for All Users

Figure 42 shows the first page for the receipt service, the login page. The unregistered visitor can register by clicking the "Register" hyperlink to do the registration and then login with his or her valid username and password. If clicking that hyperlink, the visitor will see the register page, shown in Figure 43. After logging in, there is a "logout" hyperlink in all following pages for any kinds of users to log off to the login page.

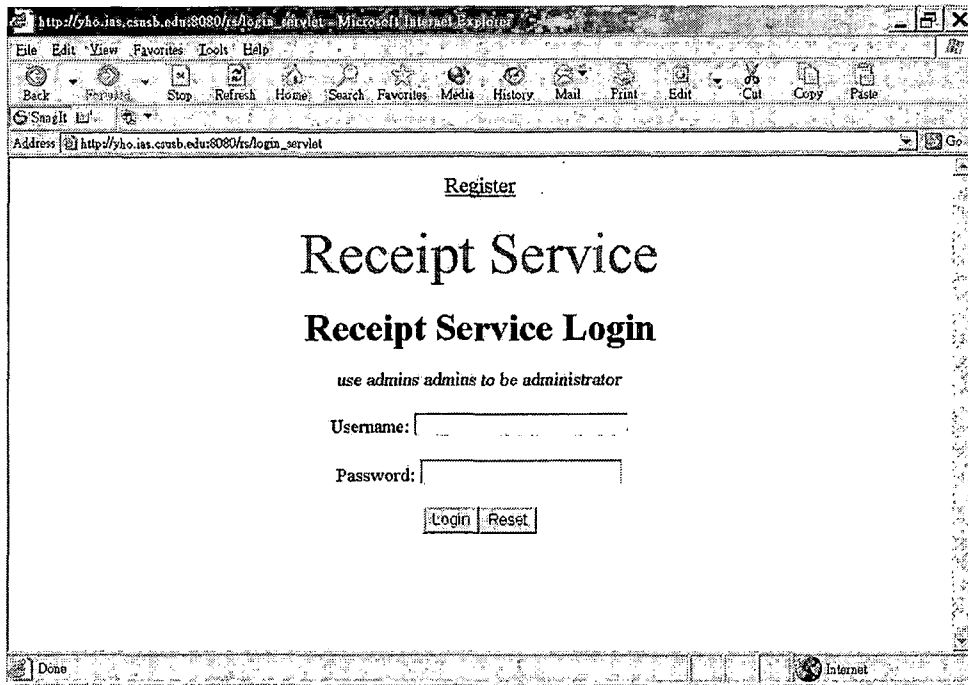


Figure 42. Login Page for the Receipt Service

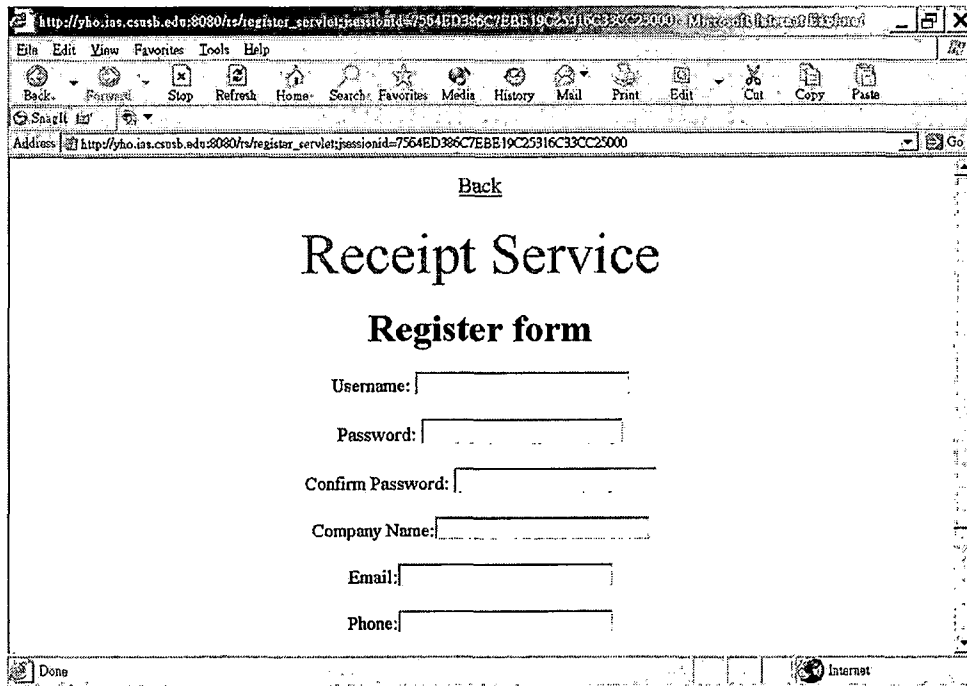


Figure 43. Register Page

10.2.2 For the Normal Users

If the visitor registers successfully, then he or she becomes a member and can login by inputting valid data and clicking the button of "login" in the login page. After logging in, the member can see the user page, shown in Figure 44. The member can see his or her information by clicking the "Your Information" hyperlink in the user page. Then, the member will see the information page, shown in Figure 45. In the information page, the member can modify his or her information by clicking the "Modify information" hyperlink or password by clicking the "Change password" hyperlink. If the member chooses to modify the

information, he or she will see the modify information page, shown in Figure 46, for doing the modification. If the member chooses to change the password, he or she will see the change password page, shown in Figure 47, for changing the password. The new password works from the time when the member logs in next time.



Figure 44. User Page

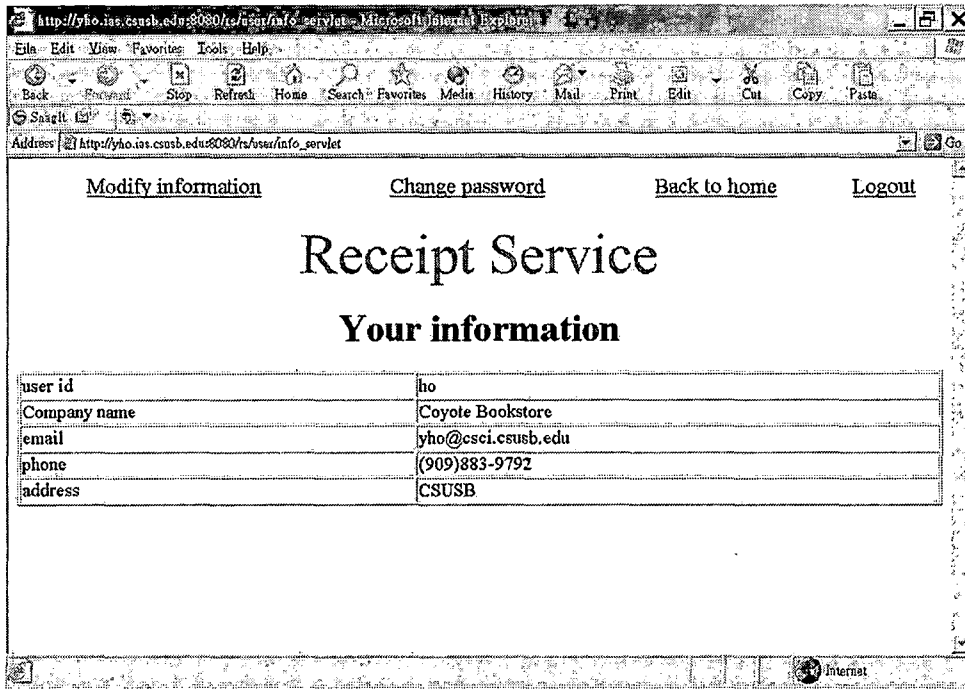


Figure 45. Information Page

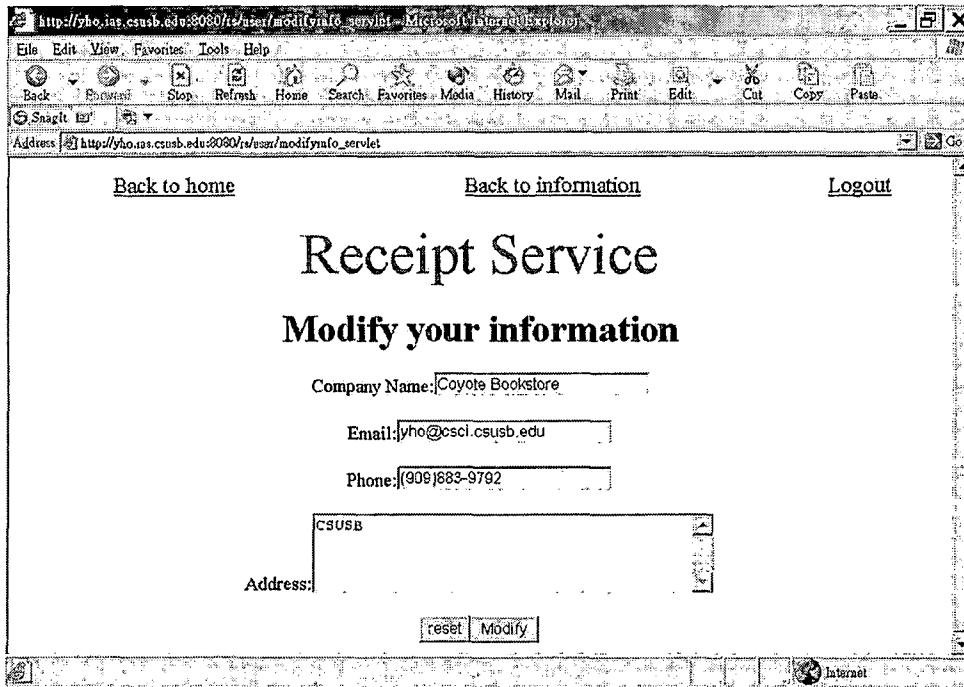


Figure 46. Modify Information Page

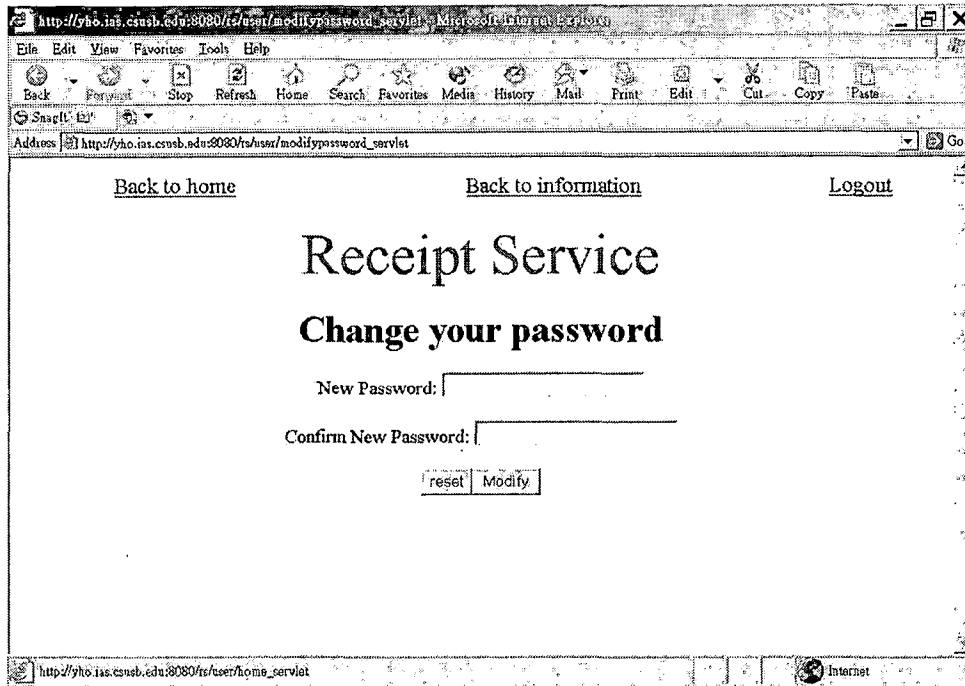


Figure 47. Change Password Page

The member can deal with his or her logo by clicking the "Your Logo" hyperlink in the user page. Then, the member will see the logo page, shown in Figure 48. The member can upload or replace the logo by clicking the "replace" hyperlink in the logo page. Then, the member can see the upload logo page, shown in Figure 49. In that page, the member can upload the image file in jpg format for the logo. The member can delete the logo by clicking the "Delete logo" hyperlink in the logo page if he or she has already uploaded a logo. Then, the member will see the confirm page, shown in Figure 50, for he or she to decide whether to do so or not.

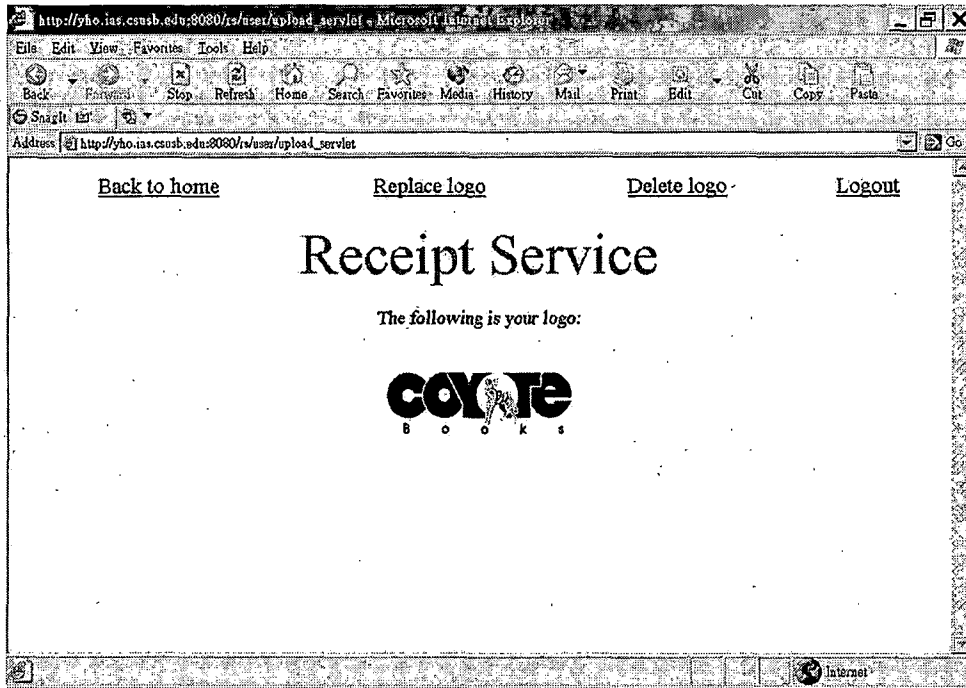


Figure 48. Logo Page

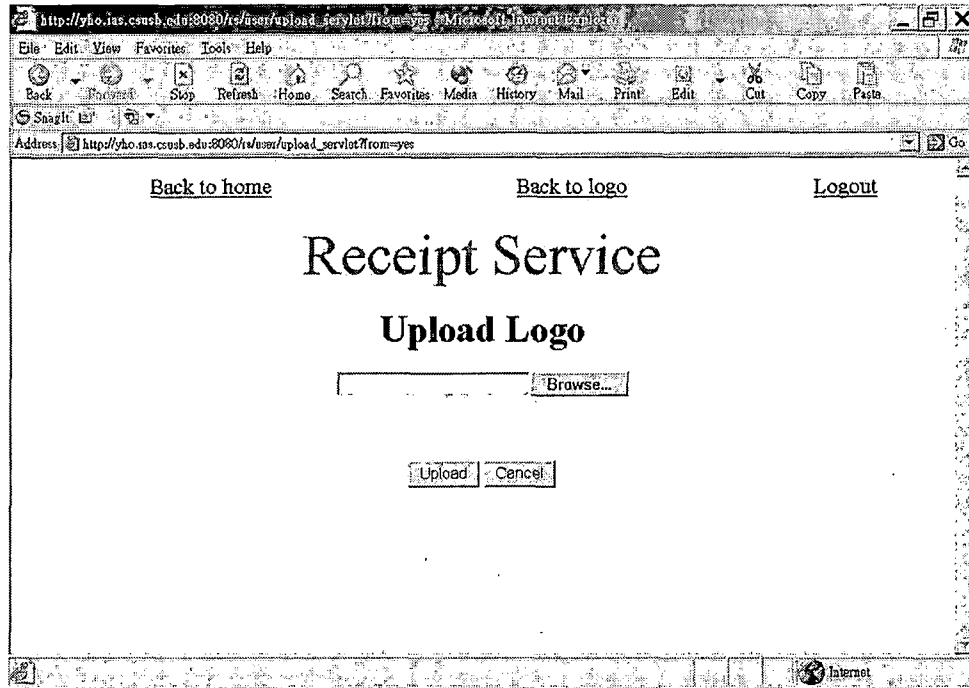


Figure 49. Upload Logo Page

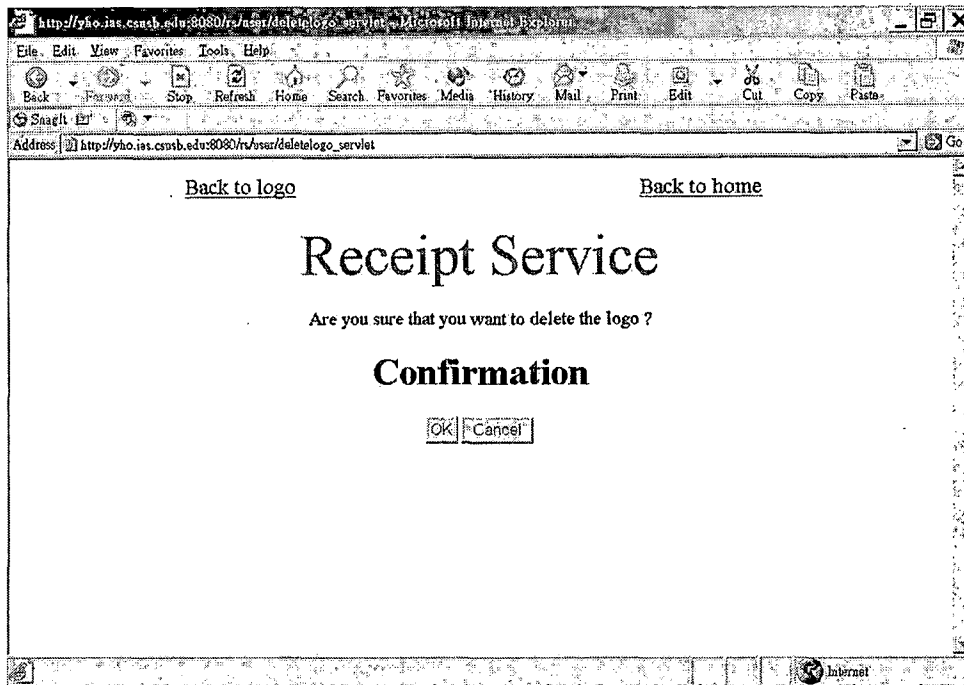


Figure 50. Confirm Page

The member can see the receipt service's certificate file by clicking the "View the rs.cert" hyperlink in the user page. Then, the member will see the cert page, shown in Figure 51. The content of rs.cert is needed by all the client web sites in order to do the transaction with the receipt service. All the clients must copy the content and do the instruction as mentioned in Chapter Seven.

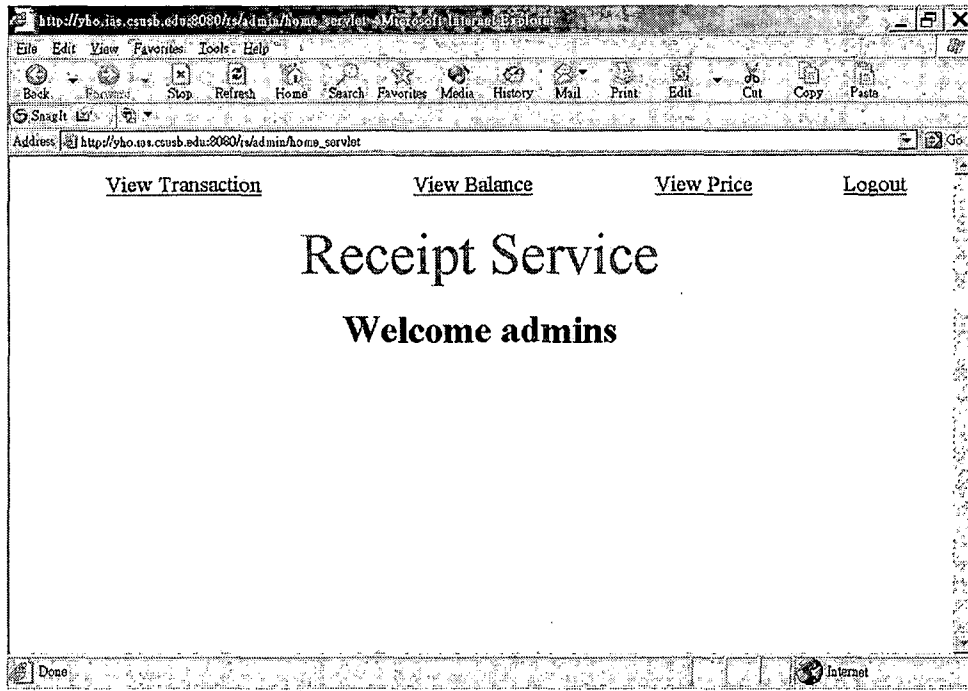


Figure 52. Admin Page

[Back to home](#)
[view chart of transactions](#)
[Logout](#)

Receipt Service

Transaction Information

Transaction Id	Company Name	Served	Date
1	Coyote Bookstore	true	2004-04-06
2	Coyote Bookstore	true	2004-04-08
3	Coyote Bookstore	true	2004-04-16
4	Coyote Bookstore	true	2004-04-20
5	Coyote Bookstore	true	2004-04-20
8	Coyote Bookstore	true	2004-04-20
9	Coyote Bookstore	true	2004-04-22

Figure 53. Transaction Page

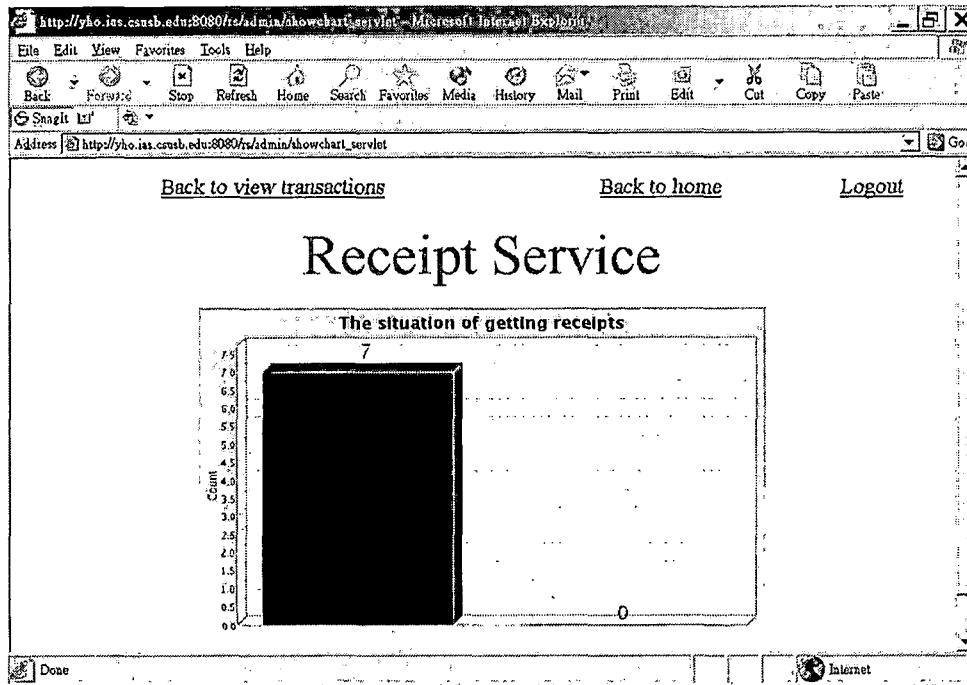


Figure 54. Chart Page

The administrator can see the current balance by clicking the "View Balance" hyperlink in the admin page. Then, the administrator will see the balance page, shown in Figure 55. The administrator can see the price for the service by clicking the "View Price" hyperlink in the admin page. Then, the administrator will see the price page, shown in Figure 56. The administrator can modify the price by clicking the "Change Price" hyperlink in the price page. Then, the administrator can see the change price page, shown in Figure 57, for modifying the price.

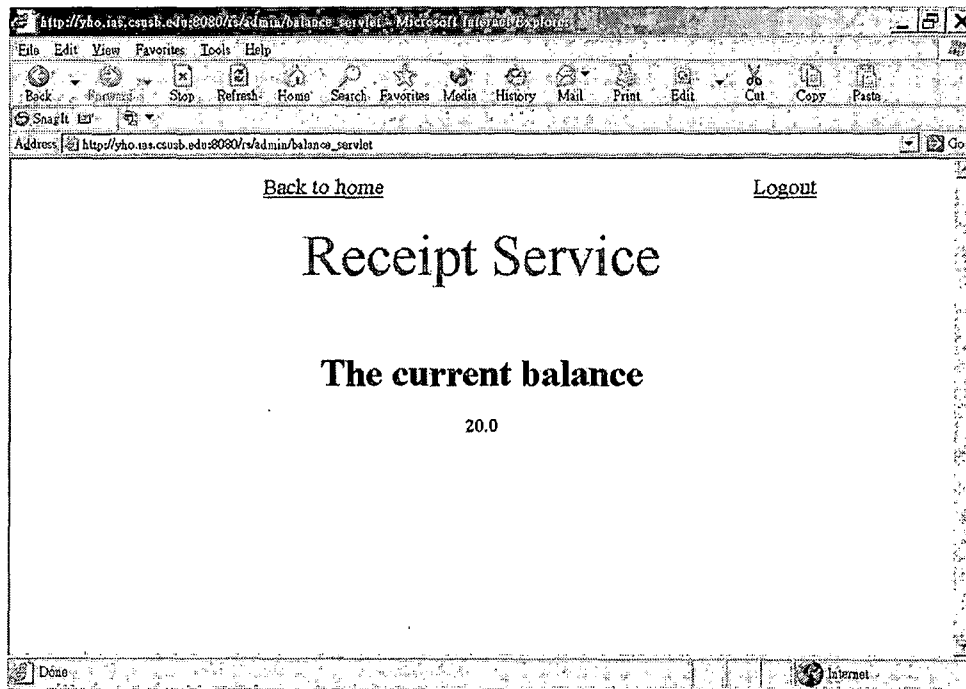


Figure 55. Balance Page

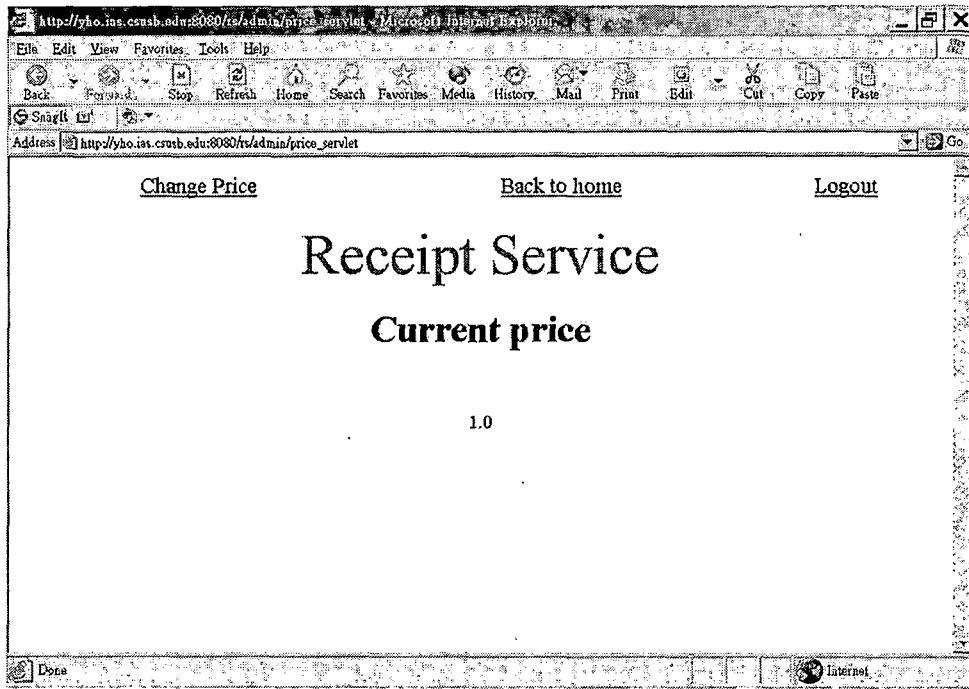


Figure 56. Price Page

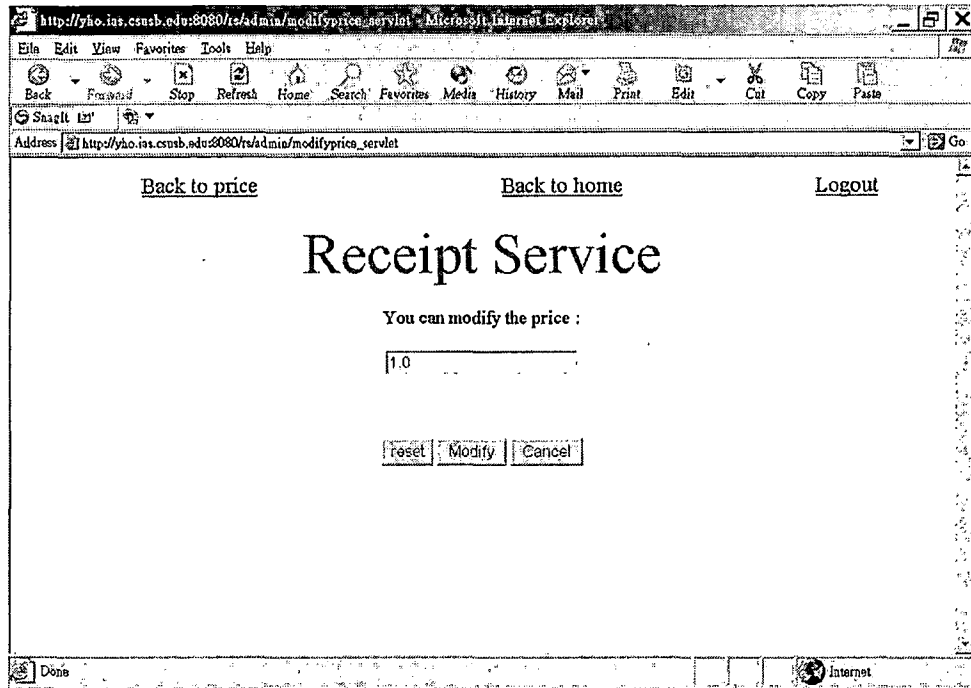


Figure 57. Change Price Page

CHAPTER ELEVEN

CONCLUSION AND FUTURE DIRECTIONS

11.1 Conclusion

This system mainly applies the LCP which by Steven Hsiao and extended by David Turner. We can take this system as an example for applying LCP. After testing many times, the system in LCP transaction is very stable and correct. Once the module of LCP is developed, the clients, including the bookstore and the receipt service, are capable of invoking the communication with LCP issuer very easily. However, AXIS is necessary in this system. It is impossible to finish providing remote service when the two clients in different locations.

11.2 Future Directions

The bookstore should have many features for customers to use. The receipt service can be added with the options providing additional types of receipts for its customers. In other words, adding more arguments inside the function of producing receipts is possible.

REFERENCES

1. David A. Turner. The Lightweight Currency Protocol Draft. September 2003. <http://www.ietf.org/internet-drafts/draft-turner-lcp-01.txt>
2. David A. Turner and Keith W. Ross. A Lightweight Currency Paradigm for the P2P Resource Market. Submitted, Sept 2003.
3. David A. Turner and Daniel Havey. Controlling Spam through Lightweight Currency. Submitted, August 2003.
4. Chih-Wen Chiao. A Secure Lightweight Currency Service Provider, Master's Project in Computer Science Department of Cal State University San Bernardino, U.S.A., December 2003.
5. IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1993).
6. Ivor Horton. Beginning Java 2. Wrox Press Inc, 1st Edition, 2000.
7. PostgreSQL 7.3.2 Documentation. September 2002. <http://www.postgresql.com/docs/>
8. SSL 3.0 Specification. November 1996. <http://wp.netscape.com/eng/ssl3/>
9. Jason Hunter, Java Servlet Programming. O'Reilly & Associates, 2nd Edition, January 15, 2001
10. SSL 3.0 Specification. November 1996. <http://wp.netscape.com/eng/ssl3/>
11. Jayson Falkner, et al. Beginning JSP Web Development. Wrox Press Inc, 1st Edition, August 2001.