2000

An e-commerce site generator

Younik Lee

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

Part of the E-Commerce Commons

Recommended Citation

https://scholarworks.lib.csusb.edu/etd-project/1612

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.
AN E-COMMERCE SITE GENERATOR

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
Younik Lee
December 2000
AN E-COMMERCE SITE GENERATOR

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

by
Younik Lee
December 2000

Approved by:

Dr. George M. Georgiou, Chair, Computer Science
Dr. Owen Murphy, Professor, Computer Science
Dr. Kerstin Voigt, Associate Professor, Computer Science
ABSTRACT

The e-commerce site generator called "CARTSHARE" is a software application that provides Web-based user tools to conduct e-business over the Internet. The purpose of this software is to provide an affordable and accessible environment for small businesses to conduct e-commerce in a simple manner.

CARTSHARE allows users to create multiple online shopping stores from a Web browser interface. By sharing the resources required for e-business, CARTSHARE minimizes the cost of building an e-commerce site.

Sharing resources, such as a server machine, database, merchant account and shopping-cart software eliminates a significant amount of fees and effort to create and maintain an e-business. For this reason, CARTSHARE provides an affordable e-commerce solution to small businesses or individuals, who want to sell products without a high initial overhead. The only resources required to create an online business through CARTSHARE are a web browser and an Internet connection. It does not require any software installation or programming skills.

The CARTSHARE system was designed using the object-oriented concept and was implemented in Java Servlet.
Using Java Servlet, which is a new and efficient server-side application development API, the CARTSHARE software provides platform independence and extensibility for further development. This system also supports two different database management systems (MYSQL [11] and ORACLE [12]) and uses a connection pooling mechanism [5] to increase the performance of database connection.

Using CARTSHARE, users can easily create an online store, maintain a product catalog and control transactions using the Web.
# TABLE OF CONTENTS

ABSTRACT ........................................ iii

LIST OF TABLES ................................. viii

LIST OF FIGURES ................................. ix

CHAPTER 1 : SOFTWARE REQUIREMENT SPECIFICATION

1.1 Introduction ................................. 1

1.2 Product Overview and Summary ................. 4

1.2.1 Perspective of Store Owner ................. 6

1.2.2 Perspective of Customer ................... 7

1.3 Technologies

1.3.1 Java Servlet ................................ 8

1.3.2 Database Connection Pooling ............... 9

1.3.3 Session Tracking ........................... 10

1.4 Development, Operating, and Maintenance Environments

1.4.1 Development Environment ................... 11

1.4.2 Operating Environment ...................... 11

1.4.3 Maintenance Environment .................... 11

1.5 Graphical User Interface (GUI)

1.5.1 General Layout ............................. 12

1.5.2 Home ....................................... 14

1.5.3 Store Builder ............................... 15

1.5.4 Store Manager Login ....................... 16
1.5.5 Store Manager Menu .................. 17
1.5.6 Product Addition .................... 18
1.5.7 Product List ......................... 19
1.5.8 Product Modification ................. 20
1.5.9 Store Logo Update .................... 21
1.5.10 Transaction Search ................... 22
1.5.11 Transaction Report ................. 23
1.5.12 Store Update ........................ 24
1.5.13 New Customer Signup ............... 25
1.5.14 Order History ....................... 26
1.5.15 Customer Service Request .......... 27
1.5.16 Customer Account Update .......... 28
1.5.17 Password Retrieval .................. 29
1.5.18 Shop By Category ................... 30
1.5.19 Shop By Store ....................... 31
1.5.20 Store ................................ 32
1.5.21 Product Information ................. 33
1.5.22 Add to Shopping Cart ............... 34
1.5.23 Shopping Cart ....................... 35
1.5.24 Order Confirmation ................. 36
1.5.25 Product Search Result .............. 37
1.5.26 Change Order Status ............... 38
1.5.27 Write Store Review .................. 39
LIST OF TABLES

Table 1. User Inputs and Corresponding Actions for Store Owner ............... 13
Table 2. User Inputs and Corresponding Actions for Customer .................. 13
Table 3. Database Table: Member ........................................... 48
Table 4. Database Table: Store .............................................. 49
Table 5. Database Table: Owner ............................................. 50
Table 6. Database Table: Product ........................................... 50
Table 7. Database Table: Transaction ...................................... 51
Table 8. Database Table: Tax ............................................... 52
Table 9. Database Table: Review .......................................... 52
Table 10. Unit Testing ....................................................... 56
Table 11. Integration Testing ............................................... 57
Table 12. System Testing ................................................... 58
LIST OF FIGURES

Figure 1. Home ........................................... 15
Figure 2. Store Builder ................................. 16
Figure 3. Store Manager Login ......................... 17
Figure 4. Store Manager Menu ......................... 18
Figure 5. Product Addition ............................. 19
Figure 6. Product List ................................... 20
Figure 7. Product Modification ......................... 21
Figure 8. Store Logo Update ............................ 22
Figure 9. Transaction Search ............................ 23
Figure 10. Transaction Report ......................... 24
Figure 11. Store Update .................................. 25
Figure 12. New Customer Signup ....................... 26
Figure 13. Order History ................................ 27
Figure 14. Customer Service Request .................. 28
Figure 15. Customer Account Update .................. 29
Figure 16. Password Retrieval .......................... 30
Figure 17. Shop By Category ............................ 31
Figure 18. Shop By Store ................................ 32
Figure 19. Store ........................................... 33
Figure 20. Product Information ......................... 34
Figure 21. Add to Shopping Cart ....................... 35
Figure 22. Shopping Cart ................................ 36
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>Order Confirmation</td>
<td>37</td>
</tr>
<tr>
<td>24</td>
<td>Product Search Result</td>
<td>38</td>
</tr>
<tr>
<td>25</td>
<td>Change Order Status</td>
<td>39</td>
</tr>
<tr>
<td>26</td>
<td>Write Store Review</td>
<td>40</td>
</tr>
<tr>
<td>27</td>
<td>Show Store Review</td>
<td>41</td>
</tr>
<tr>
<td>28</td>
<td>Overall Schema</td>
<td>42</td>
</tr>
<tr>
<td>29</td>
<td>State Diagram for Store Manager</td>
<td>43</td>
</tr>
<tr>
<td>30</td>
<td>State Diagram for Store</td>
<td>44</td>
</tr>
<tr>
<td>31</td>
<td>Directory Structure</td>
<td>45</td>
</tr>
<tr>
<td>32</td>
<td>Conceptual Model Diagram-1</td>
<td>46</td>
</tr>
<tr>
<td>33</td>
<td>Conceptual Model Diagram-2</td>
<td>47</td>
</tr>
</tbody>
</table>
CHAPTER 1 : SOFTWARE REQUIREMENT SPECIFICATION

1.1 Introduction

With the advent of the Internet, anyone with access to a computer, or an "Internet appliance", is able to perform tasks electronically that affect many aspects of his/her life. People all over are now being offered the infinite possibilities of a wired world. One of the most promising is "e-business".

"e-business" is the conduct of business over the Internet including not only buying and selling, but also customer service, and collaboration with business partners [4].

E-business promotes a new way of doing business. Via dedicated intranets and extranets, consumers and suppliers can access product specifications, place orders, check order details, track shipments, send and receive confirmations and updates, even provide after-sale service -- everything necessary to complete a transaction.

Finally, e-business is eliminating the need for more "brick-and-mortar" stores, reduces infrastructure costs, and improves inventory efficiency. The potential market for the e-commerce is beyond imagination.
Currently, most major corporations have already expanded their businesses to take advantages of the Internet. Companies are using the Web to provide product information, sell products to customers, buy parts and supplies from other companies, and advertise themselves.

Today, even small companies are rethinking their business strategies in terms of the Internet. However, starting an e-business requires advanced computer knowledge and skills, also the purchase of expensive software and hardware. Even though there are several Websites and software packages that provide a way of starting an e-business, the initial cost of starting such business is too high for most small businesses.

Currently there are several Websites allowing users to build their online stores like Yahoo Store Builder [15], CNET Store [2], EasyCart [6] and others. They provide elegant and easy ways to create online stores. One of the disadvantages of the above tools is the cost of maintaining the store, which is typically paid by a monthly fee regardless of the amount of transactions made from the store, even if no purchase order was placed. They also require users to provide extra resources like their own merchant accounts and online credit card processing.
This means every online store is required to have its own cashier.

With the CARTSHARE system, one cashier takes care of transactions for all the stores in the shopping mall, which minimizes the cost of store maintenance.

CARTSHARE provides the mechanism for creating an online store, which becomes part of an online shopping mall. This means that users can shop for products provided from all the stores within the mall or shop only at the store that they like. CARTSHARE even provides product search, which allows users to find specific products from the catalog of all stores.

CARTSHARE was designed to provide an online storefront solution that adds e-commerce functionality while greatly reducing the overhead.

The main idea of this system is the sharing of resources like the server machine, shopping cart software, merchant account, database and also a common catalog.

CARTSHARE provides an affordable e-commerce solution to many small business owners, who want to sell product through the Internet with minimal setup efforts and maintenance costs.
This project documentation describes the design, development and evaluation of the CARTSHARE software.

1.2 Product Overview and Summary

The software product CARTSHARE is designed to provide a Web-based user interface tool that creates and maintains multiple online stores in a simply way. CARTSHARE creates an online store based on data provided by a would-be storeowner through the user-friendly GUI (Graphical User Interface) of Internet browsers.

The main idea of the CARTSHARE system is the sharing of resources required to carry on e-business. It aims to provide an affordable e-commerce solution to small businesses while sharing resources like the server machine, shopping-cart software and merchant account to minimize the cost of building an online store. Each of these resources is shared by multiple stores within the CARTSHARE system. In addition, each store’s product information is retrieved from a common product catalog, from which user can directly find a product without visiting every store.

CARTSHARE is written in Java, especially in Java Servlet API, which increases the performance of Web applications, compared to traditional CGI script. With a
Java Virtual Machine, Servlet is loaded once when it is called, and it is not loaded again until the Servlet program is modified [3].

CARTSHARE supports multiple database management systems and uses a Connection Pooling mechanism [5] which is critical to database intensive applications like CARTSHARE, to optimize the database transaction processing. It creates a pool of reusable database connections to be embedded by the application component as needed. Once the user-specified number of connections are established, an application doesn’t have to reopen the connection anymore, which reduces the database connection time.

The CARTSHARE Store Builder creates a new online store with the provided user information. A directory with the store name is created, and all related files are placed under this directory for organizational and security purposes.

The CARTSHARE Store Manager provides an easy way to manage the product information stored in the database. It allows a store owner to add, delete and modify product information in the product catalog. It also allows the
store owner to control transactions and modify store information.

The Store Manager is only available to store owners of CARTSHARE via Store ID and password.

CARTSHARE supports file upload from a Web browser. From the HTML form, the product image file can be easily transferred to a designated directory.

The CARTSHARE system has a text configuration file, which includes database information, the store directory, the error log location, etc. This file can be edited using any text editor to configure the CARTSHARE system.

CARTSHARE's own error log functionality allows the system administrator to find internal Web server errors, which are caused by programming exceptions.

1.2.1 Perspective of Store Owner

The prospective store owner can build online store from store builder, which can be reached from the main web page. From the store builder interface, the prospective owner provides store information (store name, store description and store password) and owner information (email, name and address). After submitting the information, the store owner selects the logo image file from owner's local computer and uploads it.
Once the store is created, the store owner logs in to store manager giving the store name and password. Using the store manager, the store owner accesses the product catalog, the transaction report and the store account. To add a product, the store owner provides the product name, price, image file and description. Each product can be modified or deleted from the product catalog, which contains a list of all the products. When a customer places a purchase order, the store owner can create a transaction report or search the transaction by providing starting and ending dates. From the account page, the owner updates the account information (password, description, email and address) and the store logo image.

1.2.2 Perspective of Customer

A customer signs up once for membership in order to purchase products from any store within the CARTSHARE system. Once a user account is created, the customer provides user name (email) and a password to proceed to make purchases at any time. A customer has three starting points to begin shopping (shop by category, shop by store and product search). By selecting one of shopping categories, a customer can see product list within the specified category from all stores. Otherwise, if a
customer likes or trusts in a particular store, a customer limits shopping within the store, which he/she prefers. Finally a customer finds a product using a search engine, which is available on every shopping page. Providing a product keyword, all related products will appeared in the search result.

Once a customer finds the desired product, he/she simply adds it to the shopping cart. The shopping cart is not owned by any particular store. It allows the customer to buy products from several different stores within CARTSHARE at the same time.

After a purchase is made, the order status is updated the order history page and the purchase receipt with the order number is emailed to the customer. Whenever a customer places an order, a store review can be written. Store reviews will help other customers to rate each store within CARTSHARE. A customer can also request to have the forgotten password retrieved by providing his/her email address.

1.3 Technologies

1.3.1 Java Servlet

Servlet, a java class, is a generic server extension that can be loaded dynamically to expand the functionality
of a server. CGI has been the standard for developing interactive Web sites [14]. Servlet is an alternative solution to CGI, providing better performance [10].

Unlike CGI, which creates a separate process for each request, Servlet starts a new thread with each request within one Web server process. With Servlet, a server doesn’t have to create and initialize a new address space for every client’s request [9].

Servlets are server-side applications that run within the Web server, which means that they do not require support for the Java-enabled Web browser to execute Servlets [1].

1.3.2 Database Connection Pooling

Database connection pooling is a mechanism, reduces the overhead time for establishing the database connection from a database intensive application. The database connection time consists of locating a database server, establishing a communication channel and authenticating with the database user’s name and password through the database management system [10].

To reduce the overhead time, a pool of reusable connections is created and managed by a connection broker application, which is available from Java Exchange [5].
The connection broker reuses a collection of pre-established Database connections from the pool to avoid the overhead needed to connect to a database.

1.3.3 Session Tracking

Session tracking is a flexible and lightweight mechanism for maintaining the current state of a single user's sequential requests. The stateless HTTP protocol does not have the capability to recognizing that a sequence of requests is from the same client [1].

To create software like a shopping cart, maintaining a state for each transaction is crucial. The application should keep track of what each user does from the browser. With the built-in Java Session Tracking API, the server maintains the state through the use of cookies or through URL rewriting during the time period [10].

A cookie is a small piece of information sent to the client's browser and can be retrieved later by the server to identify the client. Whenever a browser refuses to accept a cookie, the session tracker uses a URL rewriting method, in which every URL link will be rewritten to include extra information like unique session ID to maintain the state and user's identity.
1.4 Development, Operating, and Maintenance Environments

1.4.1 Development Environment

CARTSHARE uses the following hardware and software on the server. Clients can use a standard Web browser such as Netscape Navigator or Internet Explorer to access it.

1. A PC (Personal Computer) with Redhat Linux 6.1 specifically
   a. 200 Mhz Pentium
   b. 3.4 GB hard disk
   c. 128 MB RAM
   d. Linux Kernel v2.2.14

2. JDK Development Kit 1.2.2

3. TWZ JDBC Driver for MySQL

4. Thin JDBC Driver for Oracle

5. MySQL-3.22.32-1

6. Oracle8i 8.1.5

7. Apache Server 1.3

8. ApacheJserv 1.1-2

1.4.2 Operating Environment

The product shall operate within the environment as specified above in section 1.4.1

1.4.3 Maintenance Environment
The product shall be maintained within the environment as specified above in section 1.4.1

1.5 Graphical User Interface (GUI)

CARTSHARE GUI offers an easy way for creating an online store from a Web browser. This section describes each of the GUI that CARTSHARE provides.

1.5.1 General Layout

The menu for the storeowner is displayed at the Store Manager site. Interaction with the Store Manager is done by simple clicks on hyperlinks. The menu consists of three categories and eight site links. Each page link allows the store owner to move on to other Store Manager utilities of interest. The utilities that each hyperlink allows the store owner to access are in the category of Product, Transaction and Account.

The menu for the shopper, which includes each shop category, store list, user account, shopping cart, order status, customer service and password retrieval, is displayed at store home site and part of above utilities can be found in every shopping site as needed.

The search box is shown on every shopping site and enables users to find a product using keyword(s). The product search is done through the SQL query.
The description of the utilities for the storeowner is found in the following sections.

Table 1. User Inputs and Corresponding Actions for Store Owner

<table>
<thead>
<tr>
<th>User Input</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click the Add a Product link</td>
<td>Display Product Addition page</td>
</tr>
<tr>
<td>Click the Modify a Product link</td>
<td>Display Modify a Product link</td>
</tr>
<tr>
<td>Click the Delete a Product link</td>
<td>Display Delete a Product link</td>
</tr>
<tr>
<td>Click the All Transaction link</td>
<td>Display All Transaction report page</td>
</tr>
<tr>
<td>Click the New Transaction link</td>
<td>Display New Transaction report page</td>
</tr>
<tr>
<td>Click the Search Transaction link</td>
<td>Display Transaction search page</td>
</tr>
<tr>
<td>Click the Store Update link</td>
<td>Display store information page</td>
</tr>
<tr>
<td>Click the Add Store Logo link</td>
<td>Display Add Store Logo page</td>
</tr>
</tbody>
</table>

The description of these utilities for the shopper is found in the following sections.

Table 2. User Inputs and Corresponding Actions for Customer

<table>
<thead>
<tr>
<th>User Input</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click the Shop Category link</td>
<td>Display Product list page in specified category</td>
</tr>
<tr>
<td>Click the See All Store link</td>
<td>Display Store list page</td>
</tr>
<tr>
<td>Click the Shopping Cart link</td>
<td>Display Shopping cart page</td>
</tr>
<tr>
<td>Click the New Member link</td>
<td>Display New Member Signup Page</td>
</tr>
<tr>
<td>Click the User Account link</td>
<td>Display User Account login page</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Click the Order Status link</td>
<td>Display Order Status login page</td>
</tr>
<tr>
<td>Type keyword in Search box and Click submit button</td>
<td>Display product list found with provided keyword</td>
</tr>
<tr>
<td>Click the Customer Service link</td>
<td>Display Customer Service page</td>
</tr>
<tr>
<td>Click the Forgot Password link</td>
<td>Display Password retrieval page</td>
</tr>
</tbody>
</table>

1.5.2 Home

Home is a main web page of the CARTSHARE system, which allows customers to start shopping by store or by category, or find a product with the search box. This page also provides links for storeowner to create a new store or login to store manager.
1.5.3 Store Builder

This page allows a new user to create a new store by providing store and owner information. Before a store is created, the name of store will be verified with existing store names. The name of store should be unique and cannot be changed once created. New user must provide a unique store name and description, as well as personal information such as email address, real name and valid address as owner information.
1.5.4 Store Manager Login

Click on Store Manager link allows the storeowner to see the following page (Figure 3). On this page, storeowner can login to store manager with valid store name and password. After password validation, storeowner can access store information.
Figure 3. Store Manager Login

If you already created your store, please login with your store name and password. You can control your store here.

Store Information

Store Name
Password

1.5.5 Store Manager Menu

This page allows the store owner to control store properties. There are three categories including product, transaction and account. Each category has submenu links to do a specific job.
1.5.6 Product Addition

This page allows the storeowner to add a new product into the store including product category, name of product, price, description and image. A product id will be assigned automatically from the product table.
1.5.7 Product List

This page lists all product information of the store: product id, product name, price and images. From this page, storeowner can see two links for product deletion and product modification.
1.5.8 Product Modification

This page allows storeowner to modify product information including product category, name of product, price, description and image.
1.5.9 Store Logo Update

This page allows the storeowner to modify the store logo image, which was provided when the store was created. A logo image file, which is selected from the local machine, is uploaded using the HTML form.
1.5.10 Transaction Search

This page allows storeowner to select a range of transaction date to be displayed and status of order processing, which can be new, complete or pending.
1.5.11 Transaction Report

This page allows the storeowner to see new purchase orders and complete transactions by changing the status of the orders. By clicking the User Email link, the browser will load a pop up window showing the user’s shipping address. The storeowner can find the information of the product purchased by clicking on the Product ID column. The order status can be managed from the order status column.
1.5.12 Store Update

This page allows the storeowner to update store and owner information. Current information is displayed first and storeowner can modify information with valid password. The store name cannot be changed at any time.
1.5.13 New Customer Signup

This page allows a new customer to signup with user information, which will be stored in database for future use. Once the signup process has completed, customer is not required to provide any further information to purchase a product.

User selected password and credit card information will be stored into database with encryption for security reason.
1.5.14 Order History

This page allows the customer to see the order history and check the status of a specific order.
1.5.15 Customer Service Request

This page allows customer to submit a customer service claim regarding any problem found with a purchased product. The customer must provide email, password, order number and a detailed description of the problem. Provided information will be forwarded to the store owner who sold that product.
1.5.16 Customer Account Update

This page allows the customer to update user information. Current information is displayed first without password and credit card information for security reasons and customer can modify information at any time.
1.5.17 Password Retrieval

On this page, a customer who has forgotten his/her password can receive the password by email.
1.5.18 Shop By Category

By clicking on any of product category from the mall homepage, the browser will load the following page. This page provides a list of products from all stores in the same category that customer select.
1.5.19 Shop By Store

By clicking on "all store" link, the browser will load the following page. This page provides a list of all stores registered in the CARTSHARE System. From this page, the customer browses through all the stores and select one to shop.
1.5.20 Store

By clicking on the name of store, the browser will load the following page. This page provides all categories and randomly selected products available within the specific store.
1.5.21 Product Information

By clicking on any of product link, the browser will load the following page. This page provides detailed product information including description and seller information. The customer can obtain further information about the product via email by clicking the store name link.
1.5.22 Add To Shopping Cart

By clicking on any of the "buy now" link, selected product can be put into the shopping cart. From the shopping cart page, customer can remove or update items in the shopping cart.
1.5.23 Shopping Cart

This page allows customer to verify or update content of shopping cart and then proceed to the check out process with items in the shopping cart. Customer must select a shipping method and provide user id, which is the same as user email address, and password. Customer can purchase items from different stores and Shipping fee will be charged per item.
1.5.24 Order Confirmation

This page allows customer to make sure whether the shipping address, shipping fee, applicable taxes and items to be purchased are correct. By clicking on complete button, Purchase order will be confirmed and customer can see an order receipt. After order was confirmed, customer can find the list of order from the order status page.
1.5.25 Product Search Result

Product search box can be found in every shop page, which allows customer to find a product with product keyword. The following page displays a sample search result.
1.5.26 Change Order Status

This page allows the storeowner to change the status of placed order, which can be either "On Order", "Complete" or "Pending".
## Figure 25. Change Order Status

Use this form to change order status.

<table>
<thead>
<tr>
<th>Order Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On Order</td>
<td>Store owner has received order but hasn’t process the order yet.</td>
</tr>
<tr>
<td>Complete</td>
<td>Store owner has sent the ordered items.</td>
</tr>
<tr>
<td>Pending</td>
<td>Store owner can’t process this order with some reasons.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order Date</th>
<th>Order #</th>
<th>Buyer Email</th>
<th>Seller</th>
<th>Status</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000-07-25</td>
<td>13</td>
<td><a href="mailto:ylee@ovillage.com">ylee@ovillage.com</a></td>
<td>mystore</td>
<td>Pending</td>
<td></td>
</tr>
</tbody>
</table>

1.5.27 Write Store Review

This page allows customers to write an opinion about a store they have dealt with. For privacy, the writer’s identity is not shown to other customers, but is only available to the system administrator. Only registered customers can rate stores and place comments. The range of rating score is -5 to 5.
1.5.28 Show Store Review

Clicking on any store review link, and the following page will be displayed. This page displays store reviews, which were placed by registered customers. This page will help customers make intelligent and better informed purchase online.
1.6 Functional Requirements

The CARTSHARE operational structures and their requirements are described in this section.

1.6.1 Overall Schema
Figure 28. Overall Schema

Customer <--- Internet <--- Store Owner

Web Server

Java Servlet

JDBC Driver

DB Pool

Database
1.6.2 State Diagram

The following is the state diagram of CARTSHARE Store Manager functions.

Figure 29. State Diagram for Store Manager
Figure 30. State Diagram for Store

ONLINE STORE

- View Shopping cart
- Add item to cart
- Delete item from cart
- Member Signup
- Account Update
- Customer Service Request
- Order Status
- Confirm Purchase
- Check Out
- Product Search
1.6.3 Directory Structure

The following is a file and directory structure of CARTSHARE.

Figure 31. Directory Structure

- CARTSHARE Directory
  - Servlet Directory
  - HTML Directory
  - Public Directory
    - Store..1
    - Store..2
    - Store..n
    - DB Connection Log
    - Error Log
1.6.4 Database Design

This section shows the database design structure of CARTSHARE.

1.6.4.1 Conceptual Model Diagram

Figure 32. Conceptual Model Diagram-1

```
Store
   Store ID
   Store Name
   Password
   Description
   Logo File Name

Product
   Product ID
   Product NAME
   Category
   Price
   Image Location
   Description
   Store Name

Review
   Store ID
   Writer
   Rate
   Content

Owner
   Email
   Name
   Address
```
1.6.4.2 Logical Model Table Schema

Each of the following shows a database table, description of attributes and examples.

Table Name: Member

Email: User's email address (text)
Password: User Password for login (password)
Fname: User's first name (text)
Lname: User's last name (text)
Streetl: mailing address (text)
Street2: Extra mailing address (text)
City: City of mailing address (text)
State: State Abbr. of mailing address (text)
Zip: Zipcode of mailing address (text)
Phone: User’s contact phone number (text)
Type: Type of Credit Card (text)
Num: Credit Card Number (text)(Encrypted)
Cdate: Expiration date of Credit Card (date)

Table 3. Database Table: Member

<table>
<thead>
<tr>
<th>Email</th>
<th>Fname</th>
<th>Lname</th>
<th>Street1</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:ylee@csci.csusb.edu">ylee@csci.csusb.edu</a></td>
<td>Younik</td>
<td>Lee</td>
<td>5000 Univ. Parkway</td>
</tr>
<tr>
<td><a href="mailto:tony@csci.csusb.edu">tony@csci.csusb.edu</a></td>
<td>Tony</td>
<td>Kim</td>
<td>5000 Univ. Parkway</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Street2</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1211</td>
<td>San Bernardino</td>
<td>CA</td>
<td>12345-1122</td>
</tr>
<tr>
<td>#2200</td>
<td>Irvine</td>
<td>AZ</td>
<td>21322-0099</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Num</th>
<th>Cdate</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>VISA</td>
<td>R!@WQDFEHGHTYUJU</td>
<td>09-01-2002</td>
<td>1-909-234-1122</td>
</tr>
<tr>
<td>DISCOVER</td>
<td>ADDCEERRRTSDGFGH</td>
<td>10-01-2002</td>
<td>1-949-132-0099</td>
</tr>
</tbody>
</table>
Table Name: Store
ID: Store ID generated by DB(int)
Sname: Store Name provided by User (text)
Password: Store Password provided by User(password)
Description: store description (text)
Logo: File name of store log image (text)

Table 4. Database Table: Store

<table>
<thead>
<tr>
<th>ID</th>
<th>Sname</th>
<th>Password</th>
<th>Description</th>
<th>Logo</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>MyStore</td>
<td>*********</td>
<td>My store is...</td>
<td>Mystore.gif</td>
</tr>
<tr>
<td>101</td>
<td>yourStore</td>
<td>*********</td>
<td>Your Store is...</td>
<td>YouStore.gif</td>
</tr>
</tbody>
</table>

Table Name: Owner
ID: Store ID (int)
Email: Email Address of Storeowner (text)
Fname: First Name of Storeowner (text)
Lname: Last Name of Storeowner (text)
Addr1: Street name of Mailing Address
Addr2: Extra Street name of Mailing Address
City: City Name of Mailing Address
State: State Abbr. of Mailing Address
Zip: Zipcode of Mailing Address
Table 5. Database Table: Owner

<table>
<thead>
<tr>
<th>ID</th>
<th>Email</th>
<th>Fname</th>
<th>Lname</th>
<th>Addr1</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>ylee@csci...</td>
<td>James</td>
<td>Lee</td>
<td>12332 River Dr.</td>
</tr>
<tr>
<td>101</td>
<td>jane@csci...</td>
<td>Howard</td>
<td>Kim</td>
<td>1010 San Mario ST.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Addr2</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apt #120</td>
<td>Foothill Ranch</td>
<td>CA</td>
<td>10020-0000</td>
</tr>
<tr>
<td>P.O. Box 10</td>
<td>Garden Grove</td>
<td>CA</td>
<td>11238-1111</td>
</tr>
</tbody>
</table>

Table Name: Product

ID: Product ID generated by DB (int)
Name: Name of Product (text)
(should be unique)
Category: Product Category (text)
Price: Price of Product (float)
Image: File Name of Product Image (text)
Description: Detailed Product Description (text)
SID: Store Id, which owns this Product (int)

Table 6. Database Table: Product

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Category</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>Dell 1220</td>
<td>comp</td>
<td>1000.89</td>
</tr>
</tbody>
</table>
1001 | Ping Golf Club | spor | 534.67

<table>
<thead>
<tr>
<th>Image</th>
<th>Description</th>
<th>SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dell.gif</td>
<td>Dell 1220 is the best PC in ...</td>
<td>101</td>
</tr>
<tr>
<td>Ping.gif</td>
<td>This Club is graphite...</td>
<td>100</td>
</tr>
</tbody>
</table>

**Table Name: Transaction**

- **Tid:** Transaction generated by DB (text)
- **Tdate:** Transaction Date (date)
- **Email:** Buyer’s Email Address (text)
- **ID:** Product ID (int)
- **Qt:** Quantity of Item (int)
- **Status:** Order Status (text)
- **Shipping:** Shipping option (text) (0: Standard, 1: Next Day)
- **Taxrate:** Applicable Tax Rate (float)
- **Sid:** Store ID (int) (Seller ID)

**Table 7. Database Table: Transaction**

<table>
<thead>
<tr>
<th>Tid</th>
<th>Tdate</th>
<th>Email</th>
<th>ID</th>
<th>Qt</th>
</tr>
</thead>
<tbody>
<tr>
<td>20020</td>
<td>09-20-2000</td>
<td><a href="mailto:ylee@csci.csusb">ylee@csci.csusb</a>...</td>
<td>260</td>
<td>1</td>
</tr>
<tr>
<td>20020</td>
<td>09-21-2000</td>
<td><a href="mailto:kang@csci.csusb">kang@csci.csusb</a>...</td>
<td>300</td>
<td>5</td>
</tr>
<tr>
<td>Status</td>
<td>Shipping</td>
<td>Taxrate</td>
<td>SID</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>---------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>COMPLETE</td>
<td>0</td>
<td>7.75</td>
<td>260</td>
<td></td>
</tr>
<tr>
<td>PENDING</td>
<td>1</td>
<td>10.15</td>
<td>300</td>
<td></td>
</tr>
</tbody>
</table>

Table Name: Tax
State: State Abbreviation (text)
Rate: Applicable Tax Rate (float)

Table 8. Database Table: Tax

<table>
<thead>
<tr>
<th>State</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ</td>
<td>6.25</td>
</tr>
<tr>
<td>CA</td>
<td>7.75</td>
</tr>
</tbody>
</table>

Table Name: Review
SID: Store ID (int)
Writer: The email of review writer (text)
Rate: Score Rating (int)
(only accept values -5 to 5)
Content: content of customer review (text)

Table 9. Database Table: Review

<table>
<thead>
<tr>
<th>SID</th>
<th>Writer</th>
<th>Rate</th>
<th>Content</th>
</tr>
</thead>
</table>
1.6.4.3 SQL Commands

The following is a list of the SQL commands used in CARTSHARE implementation [13].

- CREATE DATABASE db
- DROP DATABASE db
- CREATE TABLE table (column datatype [,column datatype] ...)  
- DROP TABLE table
- SELECT select_list FROM table [WHERE conditions] [ORDER BY column] [ASC|DESC]
- INSERT [INTO] table VALUES (values_list)
- DELETE FROM table [WHERE conditions]
- UPDATE table SET column = expr [,column = expr, ...]

1.7 Performance Requirements

1.7.1 Reliability

The reliability of SHARE was verified via extensive testing of all features.

1.7.2 Efficiency
Since this is an interactive program, the response time for the next display is very important. In order to make this product more efficient, Database Connection pool was used to reduce connection time from Java Servlet to the database.

1.7.3 Testability

Each requirement was identified and tested in the final product.

1.8 Exception Handling

Error messages are displayed on the browser and recorded on Error log file, which is specified on property file.

1.9 Conclusion

CARTSHARE offers various functions that make it easy to create and maintain online stores.

By sharing resources required to build an online store, the system provides an affordable E-commerce solution. It uses various mechanisms such as the HTML form based file upload, Database connection pooling, and user friendly GUI. It includes functionalities, which other existing store builders have.
The flexible design of CARTSHARE allows easy modification for transforming it to specific applications such as Online Auction.

CARTSHARE system is validated for its functionality and is ready for commercial use with Secure Socket Layer (SSL) [4].

1.10 Proposed Future Development

1. CARTSHARE system uses one online store interface for all users. With several store interface template, users can customize and differentiate own store interfaces according to their tastes.

2. To guarantee a secure transaction, CARSHARE system should be installed on Secure Sock Layer (SSL) enabled web server.

1.11 Acceptance Criteria

1.11.1 Test Acceptance Criteria

The final product has met all requirements stated in this document (Section 1.6, Section 1.7). It was tested in conformance with the following test criteria.

1.11.2 Unit Testing

The Unit testing focuses on the verification effort of the smallest unit of CARTSHARE system. The operation of
each function and GUI was tested and result of the tests are summarized in Table 1.1.1

Table 10. Unit Testing

<table>
<thead>
<tr>
<th>Unit Tested</th>
<th>Test Performed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Store Creation</td>
<td>Tested for inserted values in DB</td>
</tr>
<tr>
<td></td>
<td>Store directory created in public_html directory</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Store Logo Add</td>
<td>Tested for logo image file exist in store directory</td>
</tr>
<tr>
<td></td>
<td>Correct file name in DB</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Product Addition</td>
<td>Tested for correct values in DB</td>
</tr>
<tr>
<td></td>
<td>Product image file exist in designated directory</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Product Modification</td>
<td>Product information is updated in DB</td>
</tr>
<tr>
<td></td>
<td>Replaced image file exist</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Product Deletion</td>
<td>Tested for removing product data from DB</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Transaction report</td>
<td>Tested for valid transaction history</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Transaction search</td>
<td>Tested for valid transaction data within range of date specified</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Store Info</td>
<td>Tested for displaying correct store info to the browser</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Store Update</td>
<td>Tested for updated values in DB</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Member Signup</td>
<td>Tested for inserted values in DB</td>
</tr>
<tr>
<td>Encryption for Password and Payment info</td>
<td>X</td>
</tr>
<tr>
<td>Order Status</td>
<td>Tested for displaying order history to the browser</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Account Update</td>
<td>Tested for updated values in DB</td>
</tr>
<tr>
<td>Encryption for Password and Payment info</td>
<td>X</td>
</tr>
<tr>
<td>Customer Service</td>
<td>Tested for sending description to store owner and customer</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Show shopping cart</td>
<td>Tested for display valid product name and quantity</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Update Quantity</td>
<td>Tested for updated value from quantity of specified item</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
1.11.3 Integration Testing

The Integration testing focuses on the verification effort of the links between each functional page of CARTSHARE system. The Result of the tests is summarized in Table 10.

<table>
<thead>
<tr>
<th>Role</th>
<th>User Input</th>
<th>Desired Output</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Store owner</td>
<td>Click the Add a Product link</td>
<td>Display Product Addition page</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Click the Modify a Product link</td>
<td>Display the Modify a Product link</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Click the Delete a Product link</td>
<td>Display the Delete a Product link</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Click the All Transaction link</td>
<td>Display All Transaction report page</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Click the New Transaction link</td>
<td>Display New Transaction report page</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Click the Search Transaction link</td>
<td>Display Transaction search page</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Click the Store Update link</td>
<td>Display store information page to be modified</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Click the Add Store Logo link</td>
<td>Display Add Store Logo page</td>
<td>X</td>
</tr>
</tbody>
</table>
1.11.4 System Testing

In this phase of testing, CARTSHARE is tested as a site operator.

Table 12. System Testing

<table>
<thead>
<tr>
<th>Unit Tested</th>
<th>Test Performed</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Database system</td>
<td>Tested for changing DB user and password from system property file.</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Tested for changing DB name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tested for changing DB system name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tested for changing JDBC driver info</td>
<td></td>
</tr>
<tr>
<td>Log file for DB connection pool</td>
<td>Tested for changing log file location</td>
<td>X</td>
</tr>
<tr>
<td>Log file for system error</td>
<td>Tested for changing name and location of error log file</td>
<td>X</td>
</tr>
<tr>
<td>Payment Processing mode</td>
<td>Tested for changing payment processing mode</td>
<td>X</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------</td>
<td>---</td>
</tr>
</tbody>
</table>

59
CHAPTER 2 : DETAILED DESIGN

This chapter presents the refinements of the architecture of CARTSHARE system. What follows is the description of each main function of CARTSHARE system. For each of the main function algorithm is given in pseudo language.

<table>
<thead>
<tr>
<th>1. Function Name</th>
<th>Buy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where Used</td>
<td>Shopping Cart</td>
</tr>
<tr>
<td>Purpose</td>
<td>Add an item to shopping cart</td>
</tr>
<tr>
<td>Subitems Note</td>
<td></td>
</tr>
</tbody>
</table>

Procedure Buy
Begin
  If Method is GET
    Call DB Connection Broker
    Retrieve product information From DB
    Display product information
  Else Method is Post
    Call DB Connection Broker
    Retrieve Shopping Cart From Session
    If Same Product exist
      Update Quantity of the item
    Else
      Add Product to Shopping Cart
    End If
  Else
    Return Error Message
End If
End

<table>
<thead>
<tr>
<th>2. Function Name</th>
<th>ShowCart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where Used</td>
<td>Shopping Cart</td>
</tr>
<tr>
<td>Purpose</td>
<td>Display contents of shopping cart</td>
</tr>
<tr>
<td>Subitems Note</td>
<td></td>
</tr>
</tbody>
</table>

Procedure ShowCart
Begin
If Method is GET
Call DB Connection Broker
Retrieve Product ID and Quantity
Retrieve Product Info From DB
Display list of products
Else Method is Post
Return Error Message
End If
End

3. Function Name
Where Used Shopping Cart
Purpose Update Quantity of Item
Subitems
Note

Procedure UpdateCart
Begin
If Method is POST
Get Quantity
If Quantity is 0
Delete Item from cart
Else
Update Quantity of the item
Else
Return Error Message
End If
End

4. Function Name
Where Used Shopping Cart
Purpose Proceed to place an purchase order
Subitems
Note

Procedure CheckOut
Begin
Establish DB Connection
Verify Email and Password From DB
If Email and Password are valid
Retrieve User Info
Show Content of Shopping Cart
Show the Button to Confirm Order

61
Else
   Return Error Message
End If

5. Function Name | ConfirmOrder
   Where Used | Shopping Cart
   Purpose | Commit an purchase order
   Subitems | Send order receipt by email
   Note

Procedure ConfirmOrder
Begin
   Establish DB Connection
   Retrieve User Info
   Show Content of Shopping Cart
   Insert Transaction Info to DB
   Send Order Receipt by Email
End

6. Function Name | CreateStore
   Where Used | Store Builder
   Purpose | Create new store
   Subitems | Upload Store Logo Image
   Note | Store Name should be unique

Procedure CreateStore
Begin
   Get Store Information from the browser
   Establish DB Connection
   If Store name exist
      Return Error Message
   Else If Store name is new
      Encrypt Password
      Generate Store ID
      Insert Store Info to DB
      Create Store Directory
   Else
      Display Error Message
   End If
End

7. Function Name | LoginToStoreManager
   Where Used | Store Manager
<table>
<thead>
<tr>
<th>Purpose</th>
<th>Load store manager page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subitems</td>
<td>Insert Store ID TO Current Session</td>
</tr>
<tr>
<td>Note</td>
<td></td>
</tr>
</tbody>
</table>

Procedure LoginToStoreManager
Begin
Get Store Name and Password
Establish DB Connection
Decrypt Password
If Store Name and Password are valid
Load Store Manager Page
Else
Return Error Message
End

8. Function Name Add
Where Used Store Manager
Purpose Add a new Product to Store DB
Subitems Check session validation
Note |

Procedure Add
Begin
Get Product Info
If Session has Store ID
Establish DB Connection
Insert product Info into DB
Show success Message
Else
Return Error Message
Go To Login Page
End if
End

9. Function Name Retrieve
Where Used Store Manager
Purpose Display Product Information to be modified
Subitems Check session validation
Note |

Procedure Retrieve
Begin
Get Product ID
If Session has Store ID
   Establish DB Connection
   retrieve product Info From DB
   Display current product Info
Else
   Return Error Message
   Go To Login Page
End If
End

10. Function Name    | Modify
  Where Used         | Store Manager
  Purpose            | Modify Product Information
  Subitems           | Check session validation
  
Procedure Modify
Begin
   Get Product Info
   If Session has Store ID
      Establish DB Connection
      Update product Info From DB
      Show success Message
   Else
      Return Error Message
      Go To Login Page
   End If
End

11. Function Name    | Delete
  Where Used         | Store Manager
  Purpose            | Delete product From store
  Subitems           | Check session validation
  Note

Procedure Delete
Begin
   Get Product Id
   If Session has Store ID
      Establish DB Connection
      Delete product From DB
      Show success Message
   Else
      Display Error Message
   End If

64
End IF

End

<table>
<thead>
<tr>
<th>12. Function Name</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where Used</td>
<td>Store Manager</td>
</tr>
<tr>
<td>Purpose</td>
<td>Display list of product</td>
</tr>
<tr>
<td>Subitems</td>
<td>Check session validation</td>
</tr>
</tbody>
</table>

Procedure List
Begin
Begin
If Method is GET
   If Session has Store ID
      Establish DB Connection
      Retrieve product list within Store ID
      Display product list
   Else
      Display Error Message
   Else
      Display Error Message
Else
   Display Error Message
End IF
End

<table>
<thead>
<tr>
<th>13. Function Name</th>
<th>Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where Used</td>
<td>Store Manager</td>
</tr>
<tr>
<td>Purpose</td>
<td>Display list of purchase history</td>
</tr>
<tr>
<td>Subitems</td>
<td>Check session validation</td>
</tr>
</tbody>
</table>

Procedure Transaction
Begin
   Establish DB Connection
   IF Request Method is POST
      Get Order Date and Status
      Retrieve Data from DB
      Display list of Order and Status
   Else
      Request Method is GET
      Get Order Status
      Retrieve Data from DB
      Display list of Order and Status
   Else
      Display Error Message
### 14. Function Name Change

**Where Used:** Store Manager  
**Purpose:** Update Order Status  
**Subitems:** Check session validation  

**Procedure Change**  
```plaintext
Begin  
  Establish DB Connection  
  IF Request Method is POST  
    Get Order Number  
    Update Order Status From DB  
    Display list of Order History  
  Else  
    Display Error Message  
  End IF  
End
```

### 15. Function Name StoreInfo

**Where Used:** Store Manager  
**Purpose:** Retrieve Store Info from Store DB  
**Subitems:** Check session validation  

**Procedure StoreInfo**  
**Procedure Delete**  
```plaintext
Begin  
  If Session has Store ID  
    Establish DB Connection  
    Retrieve Store Info From DB  
    Display Current Store Info  
  Else  
    Display Error Message  
  End IF  
End
```

### 16. Function Name StoreUpdate

**Where Used:** Store Manager  
**Purpose:** Update Store Information From DB  
**Subitems:** Check session validation  

---

66
Procedure StoreUpdate
Begin
    Begin
        If Method is POST
            If Session has Store ID
                Get New Store Info
                Establish DB Connection
                Update Store Info From DB
            Else
                Display Error Message
            End If
        Else
            Display Error Message
        End If
    End
End

17. Function Name | AddStoreLogo
Where Used       | Store Manager
Purpose          | Add or Change Store Logo Image
Subitems         | Check session validation
Note

Procedure AddStoreLogo
Begin
    Begin
        If Method is POST
            If Session has Store ID
                Get Image File
                Establish DB Connection
                Add or Update File name From DB
                Add or Replace Image file
            Else
                Display Error Message
            End If
        Else
            Display Error Message
        End If
    End
End

18. Function Name | CategoryList
Where Used       | Main Page
Purpose          | Display available product Categories
Subitems         
Note

Procedure CategoryList
Begin
Extract Request Method
If Method is GET
   Get Store Name
   Establish DB Connection
   If Store Name is null
      Retrieve all product categories
   Else
      Retrieve categories within Store Name
   End If
Else
Display category list
End IF
End

19. Function Name | Store
Where Used        | Main Page
Purpose           | Display product list
Subitems          |
Note

Procedure Store
Begin
If Method is GET
   Get Category and Store Name
   Establish DB Connection
   If Store Name is null
      Retrieve products from all Store
   Else
      Retrieve products within Store Name
   End If
Else
Display product list
End IF
End

20. Function Name | Detail
Where Used        | Main Page
Purpose           | Display product information in detail

68
Subitems | Note
---|---

Procedure Detail
Begin
If Method is GET
Get Product ID
Establish DB Connection
Retrieve product Info From DB
Display product Information
Else
Display Error Message
End IF
End

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where Used</td>
<td>Main Page</td>
</tr>
<tr>
<td>Purpose</td>
<td>Find a Product with provided keyword</td>
</tr>
</tbody>
</table>

Procedure Search
Begin
If Method is POST
Get Keyword
Establish DB Connection
Retrieve related products From DB
Display product list
Else
Display Error Message
End IF
End

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Signup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where Used</td>
<td>Main Page</td>
</tr>
<tr>
<td>Purpose</td>
<td>Create new user account</td>
</tr>
<tr>
<td>Subitems</td>
<td>Note</td>
</tr>
</tbody>
</table>

Procedure Signup
Begin
Establish DB Connection
If Method is POST
Extract User Info
If User ID is Exist
  Display Error Message
Else
  Encrypt Password
  Encrypt Credit Card Number
  Insert User Info into DB
End If
Else
  Display Error Message
End If
End

<table>
<thead>
<tr>
<th>Function Name</th>
<th>UserInfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where Used</td>
<td>Main Page</td>
</tr>
<tr>
<td>Purpose</td>
<td>Display User Account Info</td>
</tr>
</tbody>
</table>

Procedure UserInfo
Begin
  If Method is POST
    Establish DB Connection
    Extract User Info
    If Valid User ID and Password
      Retrieve User Info From DB
      Insert User Id into Session
    Else
      Display Error Message
    End If
  Else If Method is GET
    If Valid Session
      Get User ID
      Retrieve User Info From DB
    Else
      Display Error Message
      Go to Login Page
    End If
  Else
    Display Error Message
End If
End

<table>
<thead>
<tr>
<th>Function Name</th>
<th>UserUpdate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where Used</td>
<td>Main Page</td>
</tr>
<tr>
<td>Purpose</td>
<td>Update User Information</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Subitems</td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td></td>
</tr>
</tbody>
</table>

Procedure UserUpdate
Begin
If Method is POST
   Establish DB Connection
   Extract changed User Info
   If Session has User ID
      Insert changed User Info into DB
   Else
      Display Error Message
      Go to Login Page
   End If
Else
   Display Error Message
End If
End

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where Used</td>
<td>Main Page</td>
</tr>
<tr>
<td>Purpose</td>
<td>Display Order Status to Customer</td>
</tr>
<tr>
<td>Subitems</td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td></td>
</tr>
</tbody>
</table>

Procedure Order
Begin
If Method is POST
   Establish DB Connection
   Extract User Info
   If Valid User ID and Password
      Retrieve Order History From DB
      Display Order History List
   Else
      Display Error Message
   End If
Else
   Display Error Message
End If
End

| Function Name | ForgotPassword |
### Where Used Main Page

**Purpose**
Retrieve and send password by User ID

**Subitems**

**Note**

---

**Procedure** ForgotPassword

**Begin**

If Method is POST

- Establish DB Connection
- Get User ID
- If User ID found in DB
  - Retrieve Password From DB
  - Decrypt Password
  - Send Password to User Email
- Else
  - Display Error Message
End If

Else

- Display Error Message
End If

End

---

27. **Function Name**

<table>
<thead>
<tr>
<th>Where Used</th>
<th>Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility</td>
<td>Record any run-time Error to file</td>
</tr>
</tbody>
</table>

**Procedure** Record

**Begin**

- Get Log file Location
- Open File Stream
- Record Date and Error Message
- Close File Stream

End
REFERENCE CITED


Http://java.sun.com/docs/books/tutorial/servlets/index.htm


[12] Oracle8i (http://www.oracle.com)

