

Automated Supply Database System

By

Jonathan Armstead

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Engineering Technology

University of Cincinnati
College of Applied Science

May 2005

Automated Supply Database System

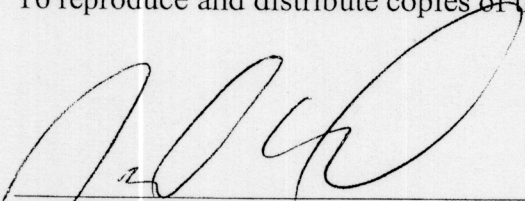
By

Jonathan Armstead


Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Engineering Technology

© Copyright 2005 Jonathan Armstead

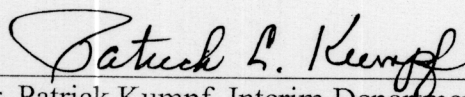
The author grants to the Information Engineering Technology Program permission
To reproduce and distribute copies of this document in whole or in part.


Jonathan Armstead

6-2-05
Date


Robert Schlemmer, Faculty Advisor

6-2-05
Date


Dr. Patrick Kumpf, Interim Department Head

6-2-05
Date

Acknowledgements

I would like to give a special thanks to Brandon Hampton, store manager of Finish Line, for really mapping out this system for me. I wouldn't have been able to even come up with the things needed in putting together an inventory system if it wasn't for him giving me all the information needed. Also, I would like to thank my co-workers at CIMx for testing my project everyday, even though I know they didn't want too. I would also like to give a special thanks to Professor Schlemmer for keeping me on track with my project.

Table of Contents

Section	Page
Title Page	i
Signature Page	ii
Acknowledgements	iii
Table of Contents	iv
List of Figures	vi
Abstract	vii
1. Statement of the Problem	1
2. Description of the Solution	1
2.1. User Profile	3
2.2. Design Protocols	3
3. Deliverables	3
4. Design and Development	4
4.1. Budget	4
4.2. Timeline	4
4.2.1. Senior Design I Accomplishments	4
4.2.2. Senior Design II Accomplishments	4
4.2.3. Senior Design III Accomplishments	5
4.3. Software	5
4.4. Hardware	5
5. Proof of Design	5
5.1. Login Screen	6
5.2. Workers Main Page	7
5.3. Manager Screen	7
5.4. Add Item	8
5.5. Edit Items	9
5.6. Remove Items	10
5.7. Modify Employee Personnel Data	11
5.8. Add New Employee Screen	12
5.9. Sales Amount by Employee Screen	13
5.10. Items Sales by Date	14
5.11. Items Low/Out of Stock	15
5.12. Sales Screen	16
5.13. Database Design	18
6. Testing Procedures	18
7. Conclusions and Recommendations	19

7.1. Conclusions	19
7.2. Recommendations	19-20
Appendix A.	
A 1. Login Page Code Snippet	21
A 2. View Sales by Date Code Snippet	21-24 25-28
References	29

List of Figures

Figure Number	Page
Figure 1. Budget	4
Figure 2. Login Screen	6
Figure 3. Manager Menu Screen	7
Figure 4. Manager Screen	8
Figure 5. Add Item Screen	9
Figure 6. Edit Item Screen	10
Figure 7. Remove Item Screen	11
Figure 8. Modify Employee Personnel Data screen	12
Figure 9. Add Employee Screen	13
Figure 10. Sales Amount by Employee Screen	14
Figure 11. View Items Sales	15
Figure 12. Items Low/Out of Stock Screen	16
Figure 13. Sales Screen	17
Figure 14. Database Design	18

Abstract

Automated Supply Database System is a system design to control the entire inventory for the Finish Line shoe store located in the Tri-County Mall. Finish Line is a billion dollar a year company, and the demand of shoes and accessories are growing at an enormous rate. Currently they're counting and checking in stock by hand, making the keeping of inventory inaccurate and very time consuming. The problem is that there is no way to really keep an accurate count on stock, and many times the warehouse sends an overload of shoes based on an inaccurate count. Finish Line has commissioned this system to help monitor and control future inventory, which will increase the efficiency of the store and eliminate fewer products that give the least amount of profit. This system was designed using the latest tools: VB.NET and Access 2003. The system is easy to use giving Management and Sales Associates the ability to complete any task needed.

Automated Supply Database System for Finish line

1. Statement of the Problem

Many shoe stores such as Finish Line are making great strides to become more efficient with their resources. Database support for management is something that has a serious need for an Automated Supply Database system (4). The database tracks thousands of shoes and accessories. Finish Line is a billion dollar a year company, and the demand of shoes and accessories are growing rapidly (3). Currently they are counting and checking stock by hand, making the stock inaccurate and very time consuming (4). The problem is that there is no way to really keep an accurate count on stock, and many times the warehouse sends an overload of shoes and accessories based on an inaccurate count (4).

2. Description of the Solution

One of the most important objectives with an Automated Supply Database system is keeping track of inventory. Controlling a companies inventory through an Automated Supply Database system can reduce the time to create inventory reports and make sure that management orders wisely. The Automated Supply Database system will also help monitor and control future inventory, which will increase the efficiency of the store. Inventory reports will also help the store stock more products that give greater profits and fewer products that give the least amount of profit.

The functions needed in an Automated Supply Database system include:

- A User-friendly interface, because management doesn't have the time to learn a complicated system.
- Ability to print a variety of reports.
- The storing of inventory information.

- Inventory information is accessible from multiple locations.
- Capability of completing sales transactions

The program is flexible in terms of its installation capabilities. It needs to be scalable to fit within the capabilities of different organization's IT structures. For instance, it must be capable of operating on a single desktop PC as well as on a multi-user client/server network.

2.1 User Profile

There are two categories of application users. The two levels include Management and Sales Associates.

Management (salaried employees)

Management will be the daily users of the application. They will be experienced computer users as well as having experience with e-mail clients and web browsers. They are also capable of editing Microsoft Access databases. This user will vary in regards to technical literacy. The portion of the application that will be viewed by the client will require very little technical literacy.

Management is also responsible for maintaining the basic information in the database. These items include employee names and identification numbers. Other items include sales reports and other items dealing with inventory of the store.

Sales Associates

Sales Associates will also be daily users of the application. They will vary in technical literacy. The portion of the application that will be viewed by the client will require very little technical literacy. Once logged in they are only able to make sales transactions and search for inventory items.

2.2 Design Protocols

The core of this application is a relational database. The database can retain all of the inventory data, as well as sales data. The archived data can be recalled for viewing and printing various reports for the management team. The application can operate in a single-user environment as well as in a multi-user network environment.

There are two main elements to this application, the user interface and the database. The user interface is linked to the database. All sales transactions, inserting, deleting, and searching items are entered into the database through the user interface. The database then calculates the totals from the sales information and stores a permanent record of all the transactions that were made. With the calculations the management will be able to determine what they have in stock and any sales information they need to retrieve.

3. Deliverables

In order to provide a well-organized project certain items were considered necessary for an automated supply database system. During the design phase of this project the following deliverables were defined:

From a client PC, Management will be able to:

- Logon to any client PC with their single username and password
- Print a variety of reports
- Be able to add, delete, or remove items
- Check e-mail and surf the internet
- View company contacts from email address book
- Print to a networked printer
- Search items within the inventory

From a client PC, Sales Associates will be able to:

- Logon to any server with their single username and password
- Commit sales transactions
- Search product items within the inventory

4. Design and Development

The next section describes the project's timeline, overall budget with hardware, and software costs.

4.1 Budget

Item	Cost
Development Budget	
Compte Computer System	\$1,500
Microsoft Windows XP Professional	\$200
Microsoft Visual Studio.net 2003	\$900
MS Access 2003	\$230
Project Total	\$2,830

Figure 1. Budget

4.2 Timeline

The project involved several challenges, learning mistakes/opportunities, and accomplishments. Below are the achievements of the Senior Design sequence.

4.2.1 Senior Design I Accomplishments - Fall 2004

During Senior Design I, I accomplished the following:

Weeks 1-5

- Research and Analysis
- Documentation
- Write Draft Proposal

Week 6-10

- Write Senior Design Proposal
- Present Proposal to IET Faculty
- Continue to design the user interface
- Preliminary design of database tables

4.2.2 Senior Design II Accomplishments – Winter 2005

During Senior Design II, I accomplished the following:

Week 1-5

- Finalized MS Access database design
- Finalized user interface design
- Documentation
- Testing
- Rough Draft of Design Freeze

Week 6-10

- Finalize user interface design
- Testing
- Documentation
- Complete Final Draft of Design Freeze
- Meeting with Advisor: Professor Schlemmer
- Senior Design Presentation and Project Demo for IET Faculty

4.2.3 Senior Design III Accomplishments - Spring 2005

During Senior Design III, I accomplished the following:

Week 1-5

- Complete project functionality
- Complete Testing
- Complete interface color and design

Week 6-10

- Submit final documentation and demonstrate final project.

4.3 Software

The software used in this automated system includes:

- Access 2003
- VB.net 2003

4.4 Hardware

The hardware used in this automated system includes:

- Web Server
- Database Server
- Client computer with Internet access

5. Proof of Design

The next section shows in detail how deliverables of the project were fulfilled and what challenges I encountered.

5.1 Login Screen

This is the login screen that contains a username and password. If the correct user information is supplied the user will be authenticated by the database and allowed to proceed to the workers area. If the user incorrectly enters their username and password, and error message will pop-up stating that it's an incorrect username/password. (See Figure 2.) ASP.NET was employed as the backend tool to connect to the database. See example of ASP.NET code in Appendix A 1.

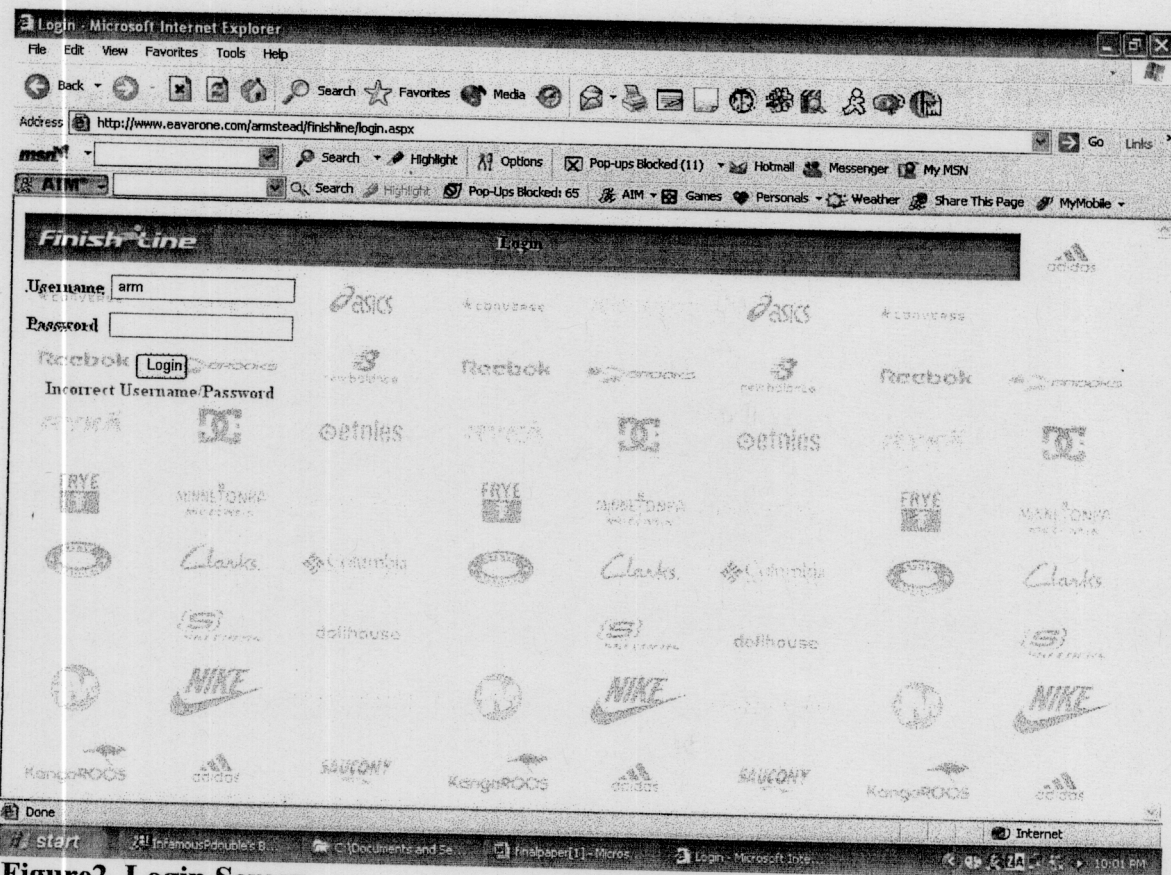


Figure 2. Login Screen

5.2 Workers Main Page

This is the workers main page. But it varies to whom has logged-in. For example, if a manger logs into the application, his welcome screen gives the option of sales or management. The regular sales associate only have the option to complete a sale.

Figure 3. Shows a screen shot of the Manager Menu screen.

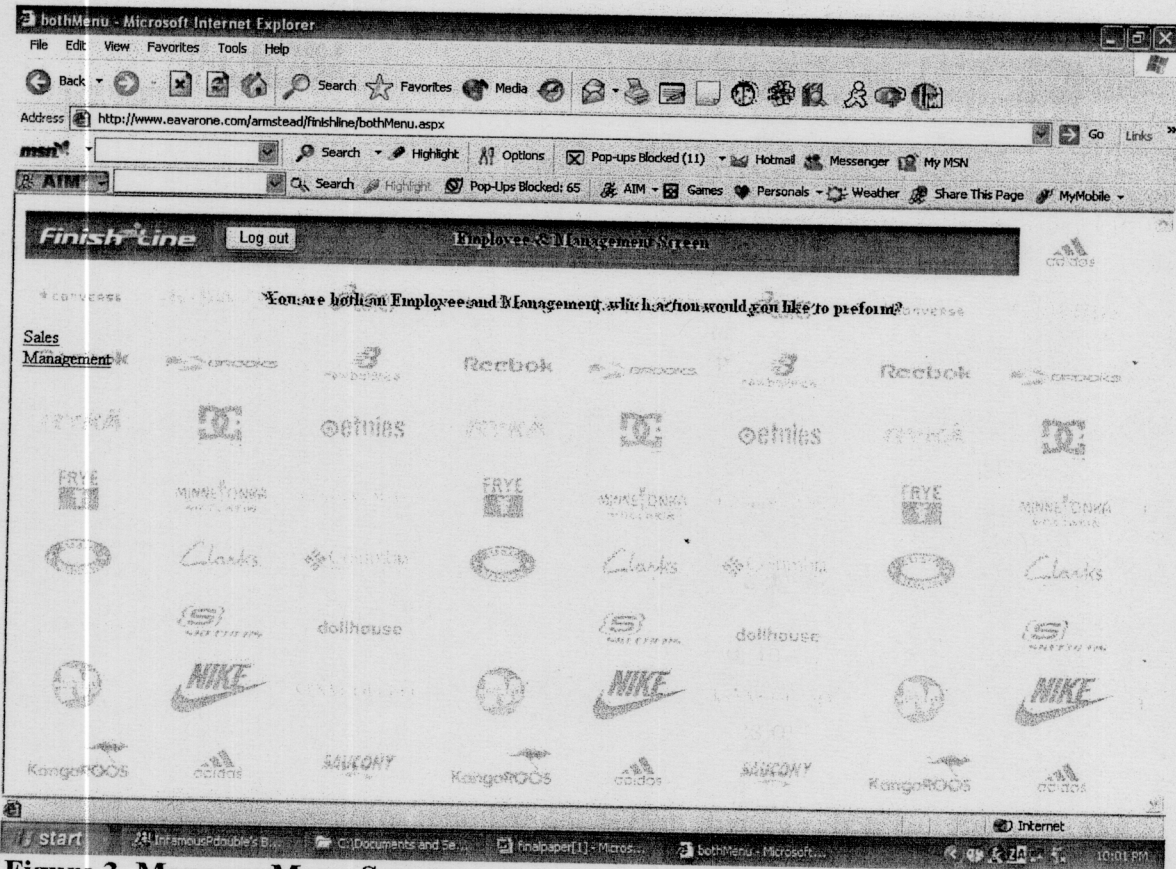


Figure 3. Manager Menu Screen

5.3 Manager Screen

When the user clicks on the “Manager” link they will be able to do a variety of things including (See Figure 4.):

- Remove Items
- Modify Employee Personnel Data
- Add New Employee
- Sales Amount by Employee
- Item Sales by Date

- Items Low/Out of Stock
- Send/Receive Emails
- Add Items
- Edit Items

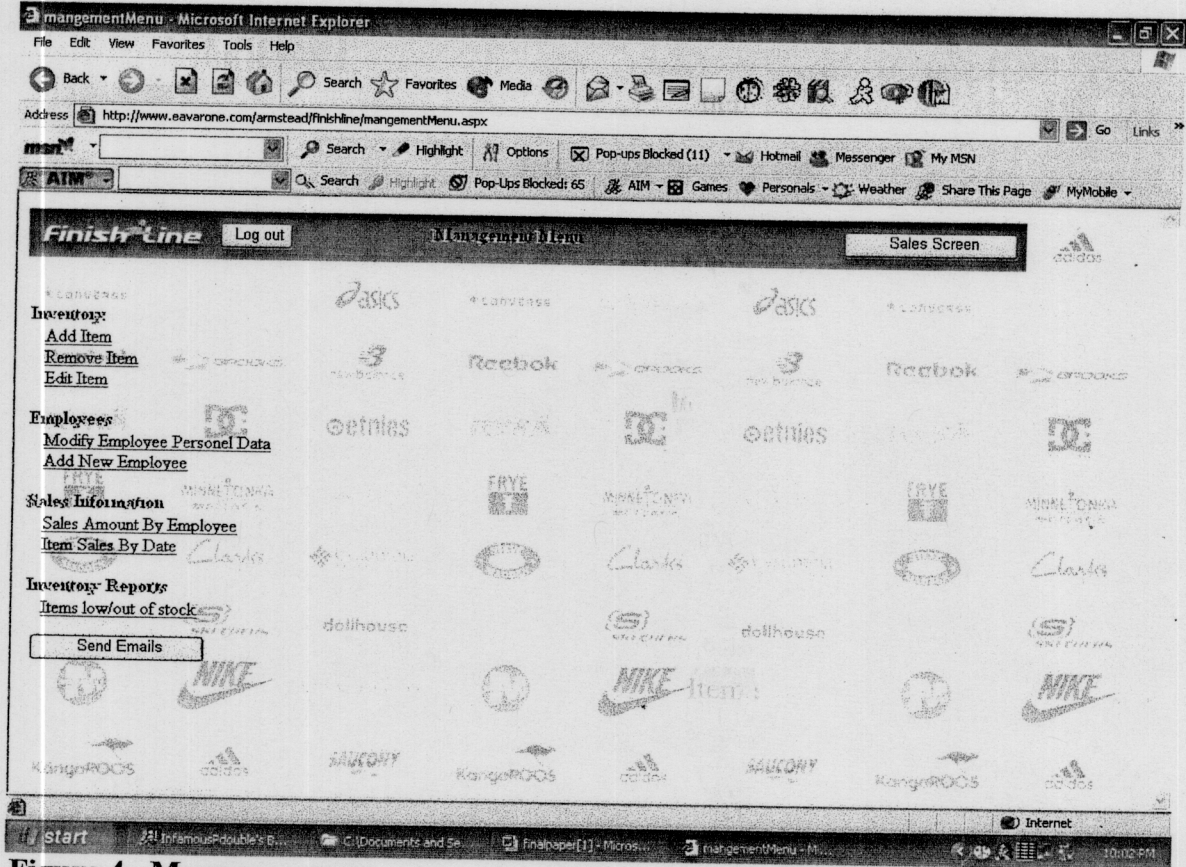


Figure 4. Manager screen

***If the Manager wants to complete a sale then they can click on “Sales” in the main menu.**

5.4 Add Item

The Add Item screen gives management the opportunity to add new items to the inventory. Figure 5 shows a screen shot of the Add Item screen.

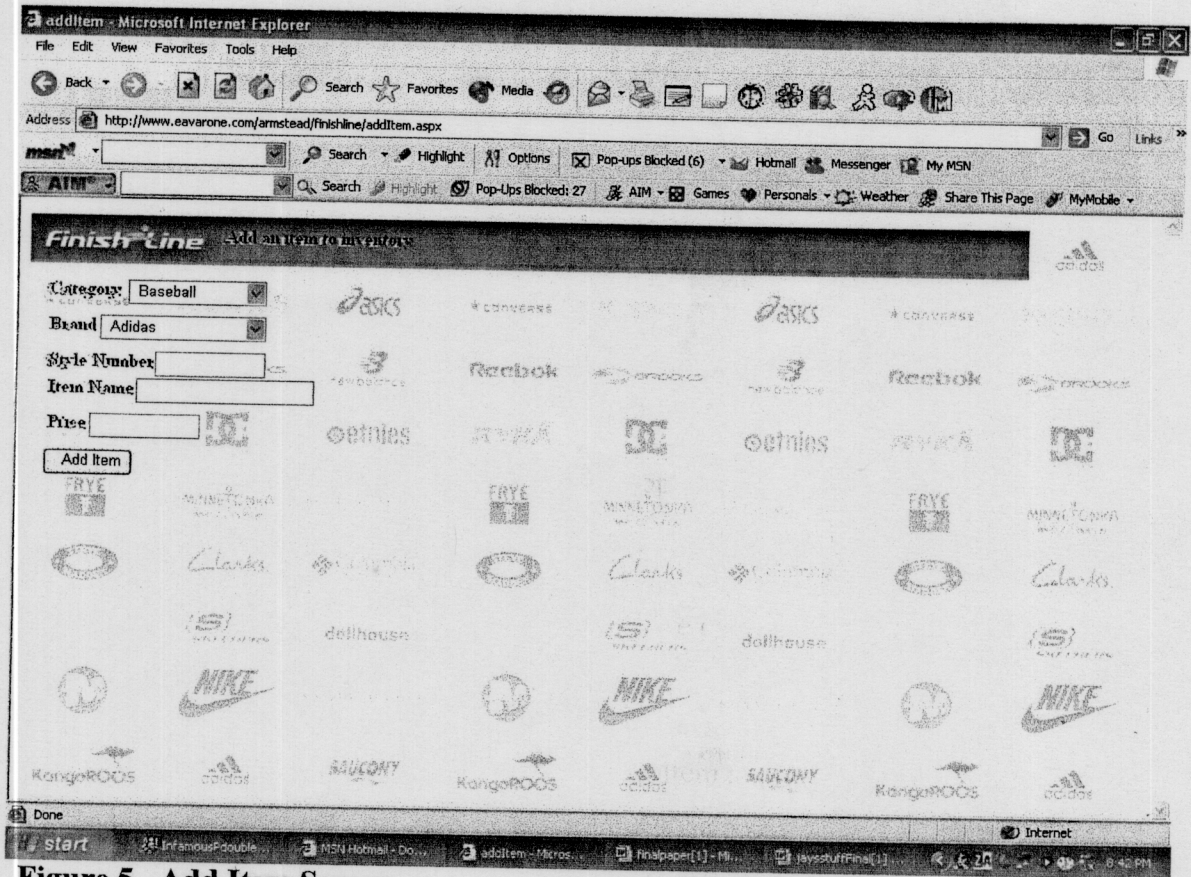


Figure 5. Add Item Screen

5.5 Edit Items

The Edit Item screen gives management the opportunity to edit items in the inventory. This is a great tool for management, because sometimes mistakes can happen and with this tool you can go back and make corrections were needed. Figure 6 shows a screen shot of the Edit Item screen.

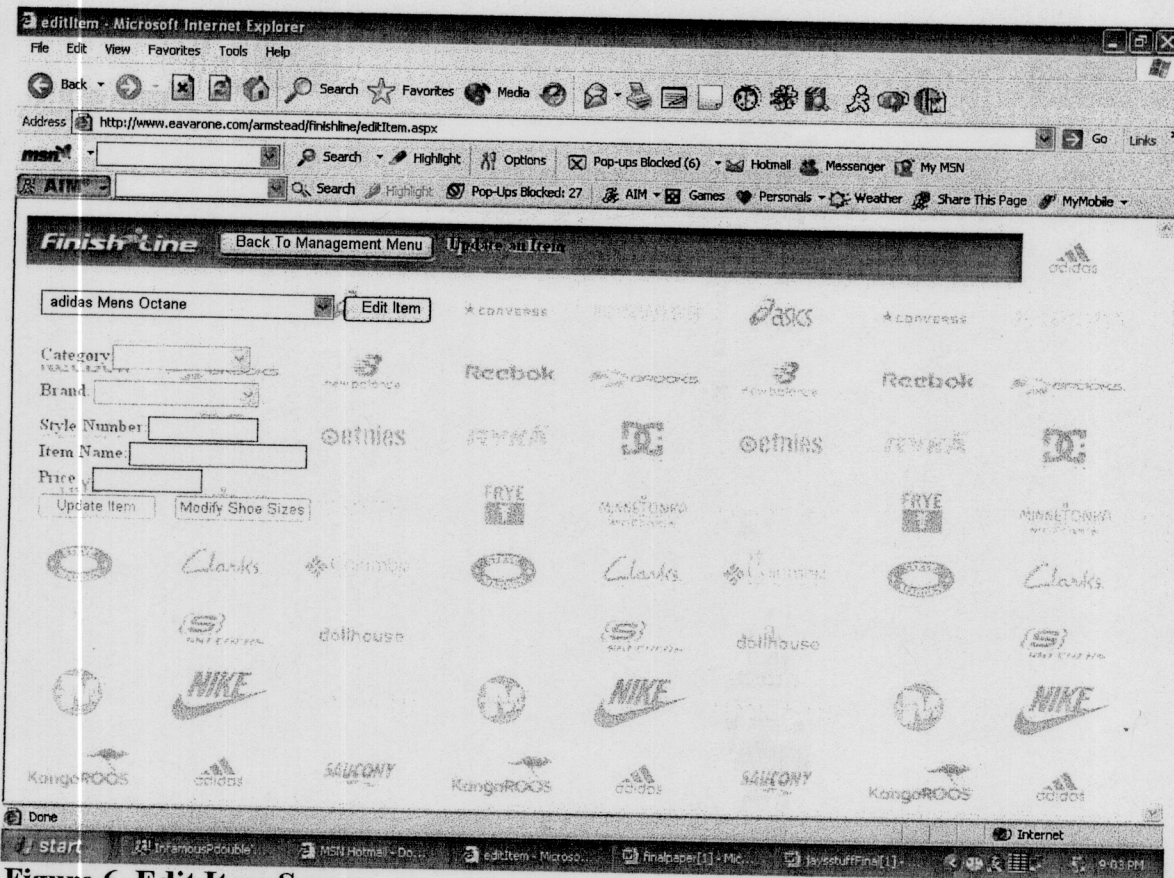


Figure 6. Edit Item Screen

5.6 Remove Items

The Remove Item screen gives management the opportunity to remove items within the inventory. This option is good for management, because it gives them the option to remove old inventory that may not be in the store anymore. Figure 7 shows a screen shot of the Remove Item screen.

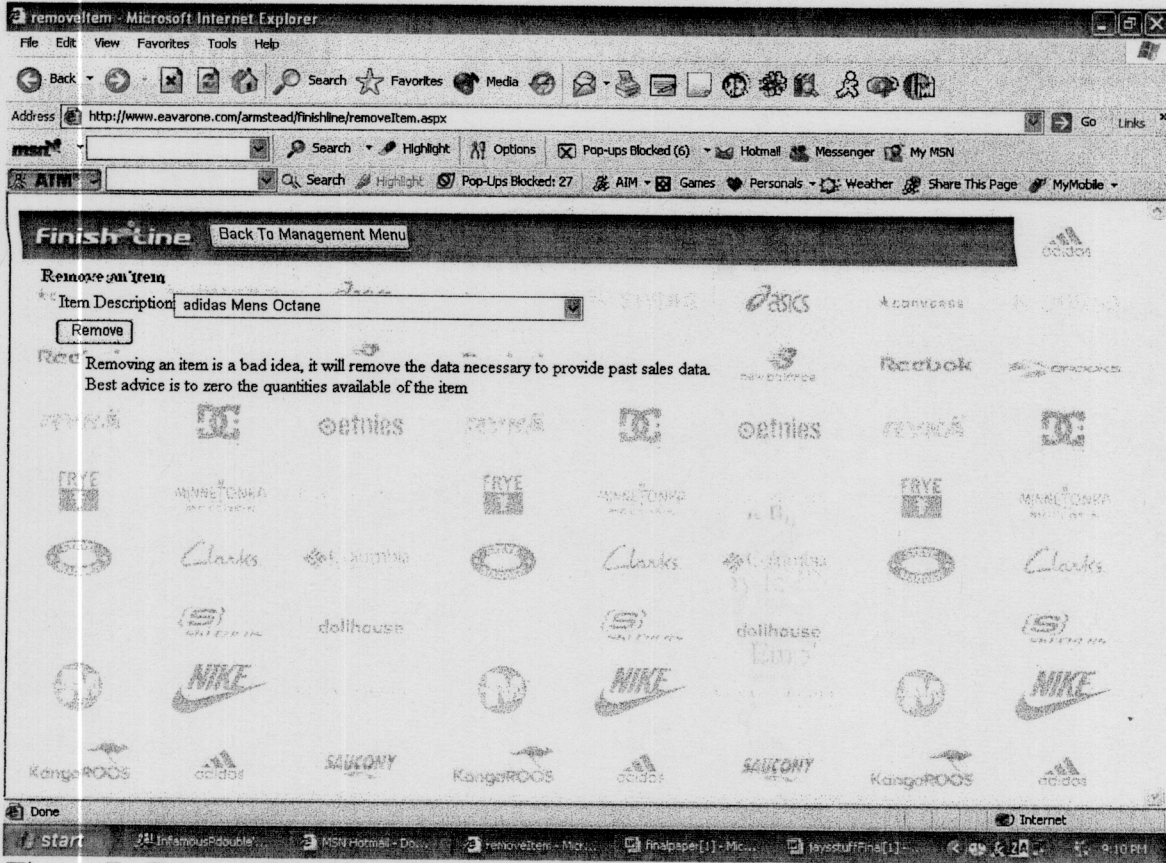


Figure 7. Remove Item Screen

5.7 Modify Employee Personnel Data

The Modify Employee Personnel Data screen gives management the opportunity to modify employee data. There are always personnel changes that go on within a business and this option gives management an easy way to change employee information. Figure 8 shows a screen shot of the Modify Employee Personnel Data screen.

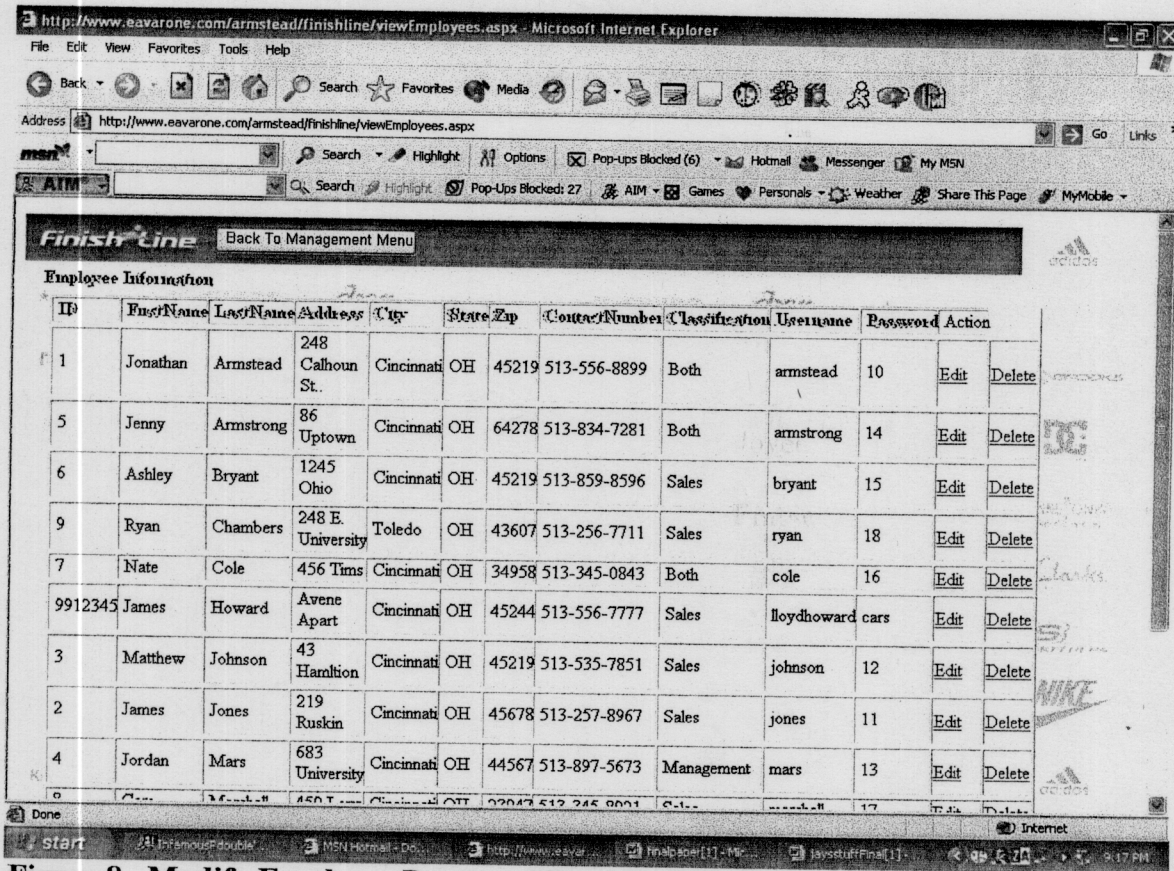


Figure 8. Modify Employee Personnel Data screen

5.8 Add New Employee Screen

This screen allows management to add new employees to the company's database. Figure 8 gives a screen shot of the Add New Employee Screen.

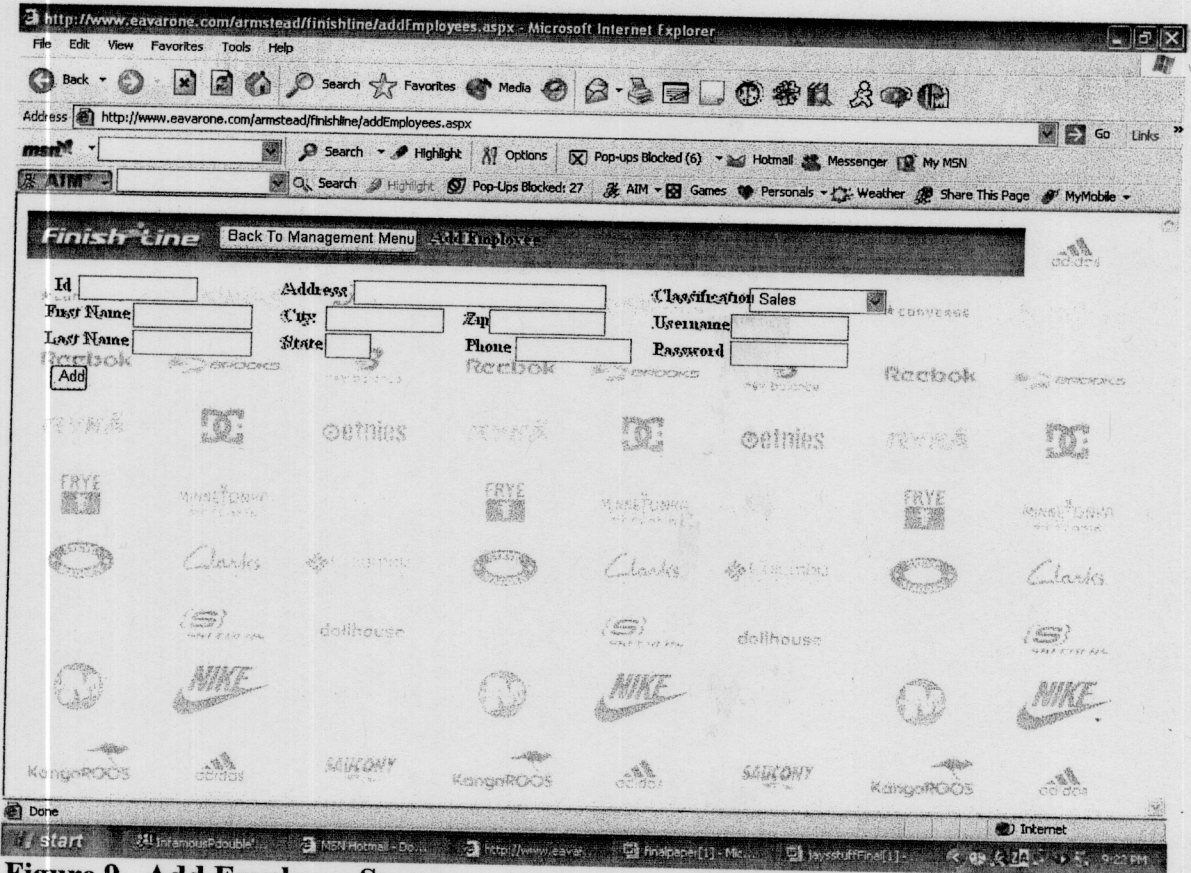


Figure 9. Add Employee Screen

5.9 Sales Amount by Employee Screen

The Sales Amount by Employee Screen shows all the information for each employee that has completed a sale within the system. It allows management to see the total amount sold for each employee. Figure 10 gives a screen shot of the Sales Amount by Employee Screen.

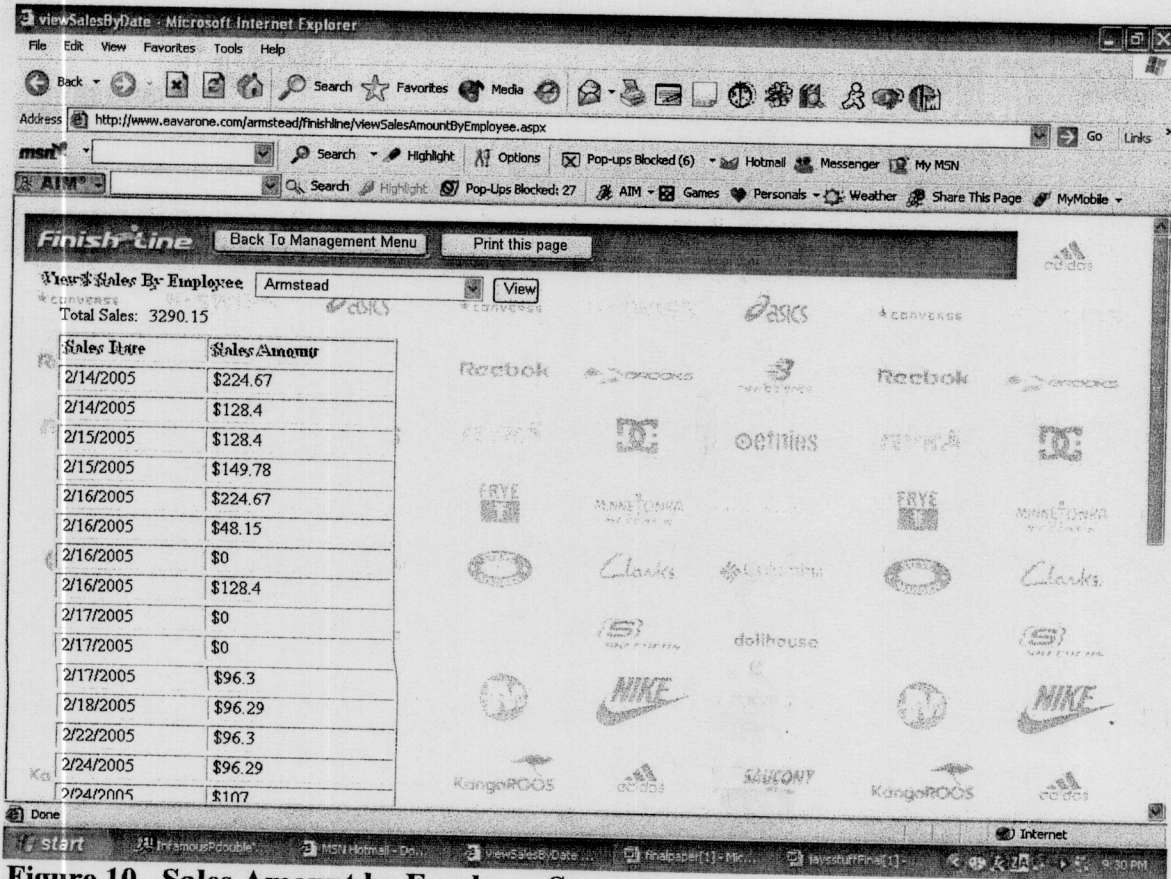


Figure 10. Sales Amount by Employee Screen

5.10 Items Sales by Date

The Items Sales by Date screen shows all the items sold for each date. This is a very important option, because it gives management the opportunity to go back to any date and see what items were sold for a particular date. See the code snippet in Appendix A 2 to see how the sales information is generated from the database. Figure 11 shows the Items Sales by Date screen.

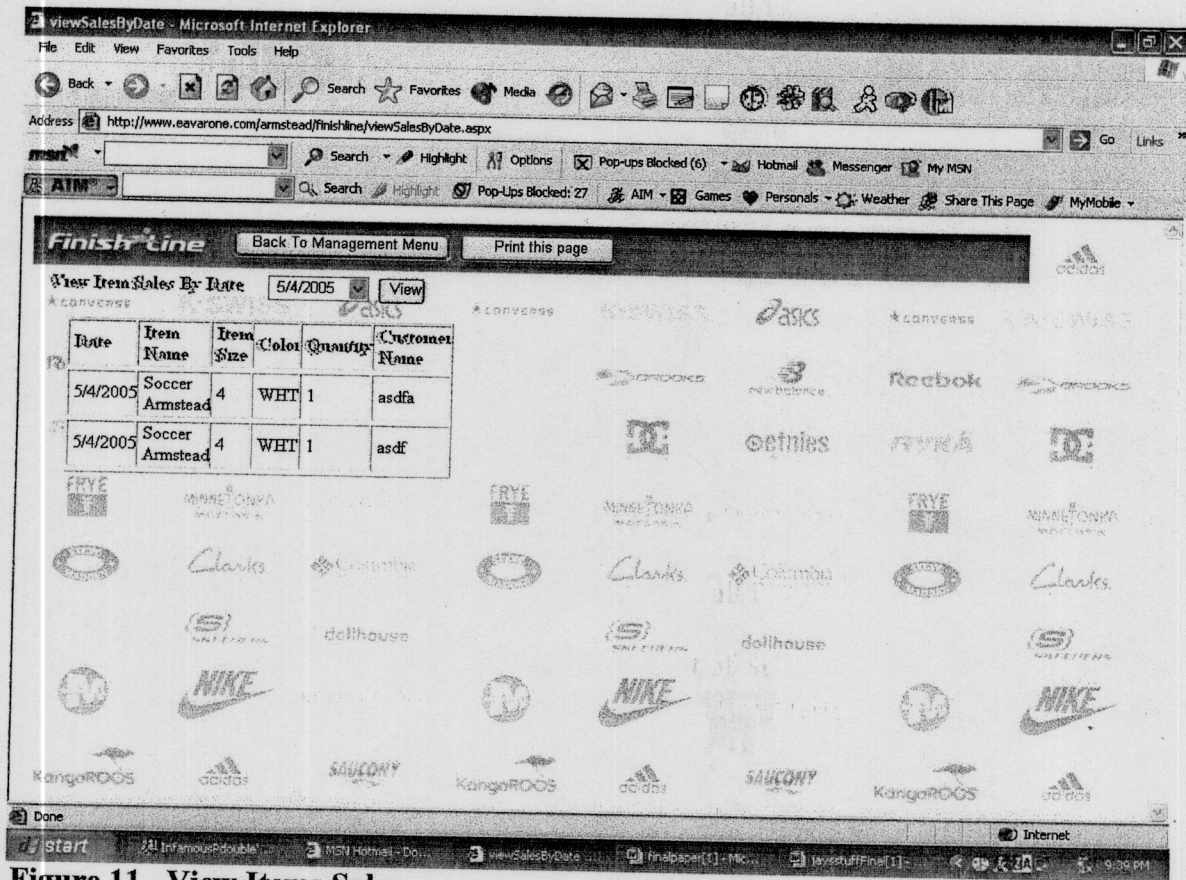


Figure 11. View Items Sales

5.11 Items Low/Out of Stock

The Items Low/Out of Stock screen is most important screen in the system. This is the screen that determines what needs to be ordered and what stock items are running low. With this screen, management is able to control all inventory that needs to be order. See Figure 12 for a screen shot of the Items Low/Out of Stock screen.

inventoryReport - Microsoft Internet Explorer

Address: http://www.eavarone.com/armstead/finishline/inventoryReport.aspx

Finish Line Back To Management Menu Print this page

Show items with 2 or less remaining in stock View

Style Number	Item Name	Item Size	Color	Quantity
6675	Cam Air Ultra Extreme	3	Blue	2
7507	Soccer Armstead	4	WHT	1
061281	Air Nutso IV	1	Red	1
664818	adidas Mens Superstar	10	Red	2
401	Asics Mens Split Second V	1	Black	2
66755	Bob Extreme	7	Black	2
41123	Brooks Mens Epiphany	9	White	2
29011235	Cam Air 2	5	Red	2

Done

start InfomousPdo TOONWEAD MSM HTML inventory RE finalpaper[1] - Microsoft Word Internet

Figure 12. Items Low/Out of Stock Screen

5.12 Sales Screen

The Sales Screen allows management and sales associates to commit sales transactions as well as the ability to search for items within the inventory. Figure 13 shows the Sales Screen.

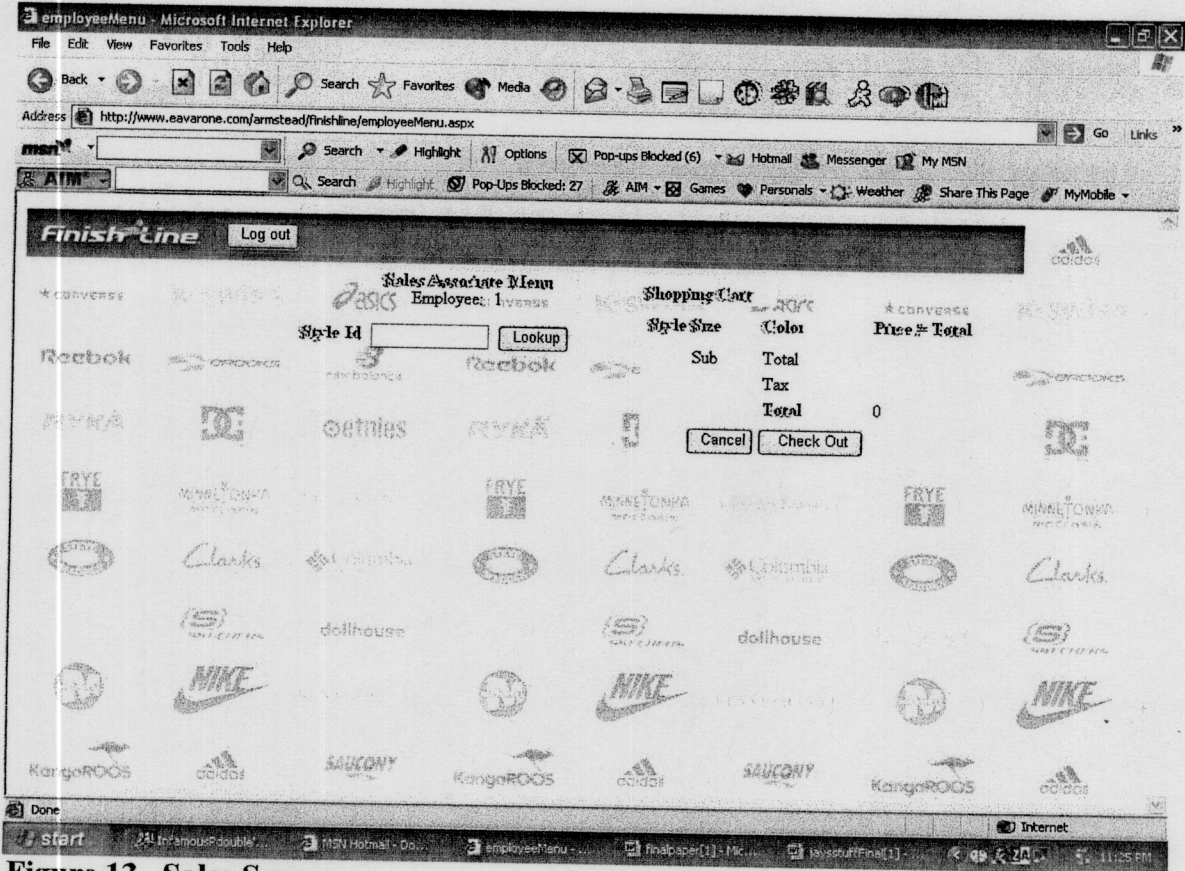


Figure 13. Sales Screen

5.13 Database Design

Figure 14 shows the database tables and relationships:

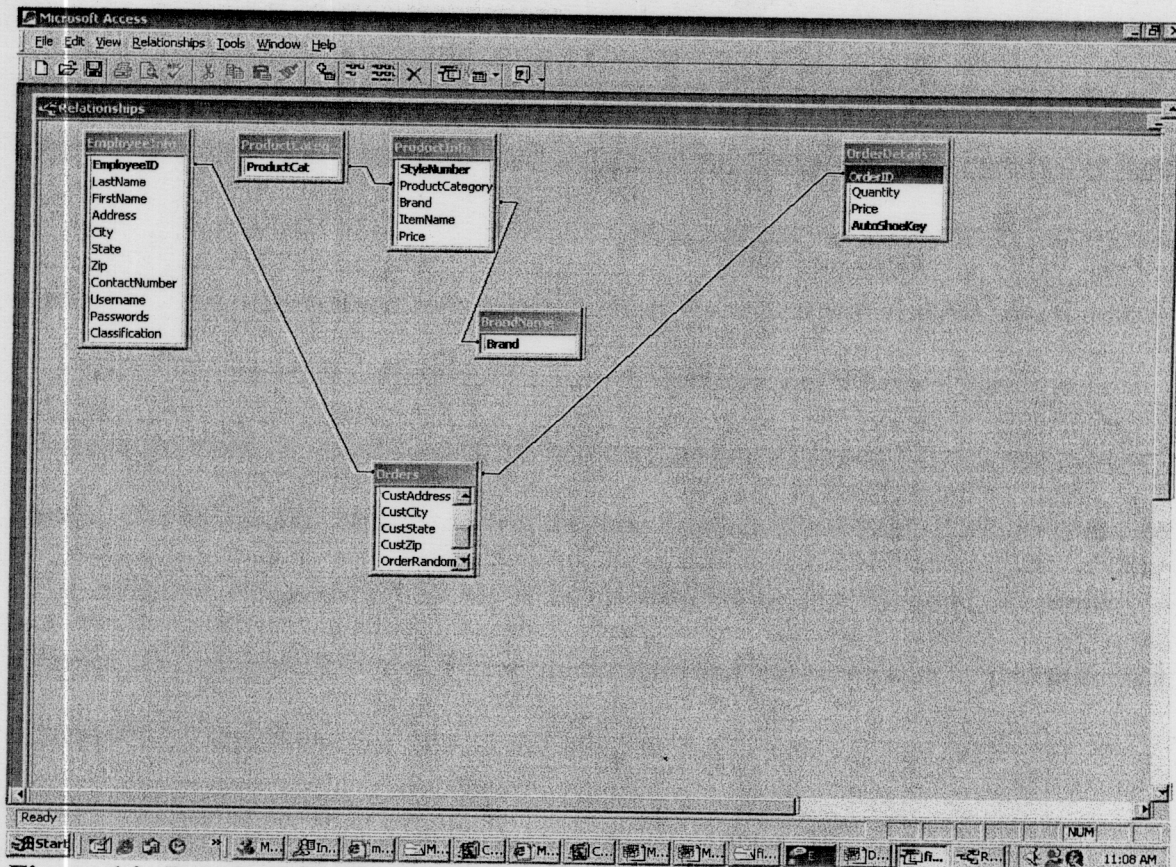


Figure 14. Database Design

6. Testing Procedures

Unit testing was an integral part of the development process. Each piece of code was tested after every update or modification that was made. When a screen was determined to be complete it was given to co-workers at my job for further unit testing. Unit testing really helped to eliminate syntax errors, as well as getting feedback on things I should take out or add to the system. The group of testers really helped to acquire information about the overall effectiveness, and visual appeal of the system. Further beta

and acceptance testing was performed by peers to test for broken links and bugs within the system. All findings were noted and changed based on priorities.

7. Conclusions and Recommendations

7.1 Conclusion

This project was created in response to Finish Line's need for a system that would eliminate the overload of stock. I created a system that can reduce inventory reports and make sure that management orders wisely. Also, the system will also help monitor and control future inventory, which will increase the efficiency of the store. Each screen is designed to work for a specific task. To prepare the project I used VB.net, and Access 2003. The project was completed over the three quarters of the Senior Design sequence. The budget of approximately \$2,830 is the estimate for the completion of this project. The project fulfilled all Design Freeze deliverables and testing was performed to ensure the product's usability.

7.2 Recommendations

While working on this project, I encountered several challenges. With me doing an inventory system for a shoe store. I had to research and review other inventory systems so I'll know what was needed in making an inventory system work. That was very challenging, because there so much information and so many things you can do to separate you inventory system from others that are on the market. So, I advise anybody trying to design an inventory system to gather as much information as you can, because it can be helpful in separating your system for others.

A solid understanding of VB.net and database designs is recommended before beginning an automated supply database system like the one that I've created. Beginning

users of VB.net and Access 2003 need to understand that there might be a steep learning curve in order to use the software proficiently.

Appendix A.

Code Snippets

A 1. Login Page Code Snippet

This code snippet is the ASP.NET code for the Login Page. It shows the connection string to the Access 2003 database, checks for successful authentication, and in case of successful authentication redirects the user to the next page.

```
<%@ import Namespace="System.Data.OleDb" %>
<%@ import Namespace="System.Data" %>
<%@ Page Language="vb" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <title>Login</title>
    <meta name="GENERATOR" content="Microsoft Visual Studio .NET
7.1">
    <meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema"
content="http://schemas.microsoft.com/intellisense/ie5">
    <script language="vb" runat="server">
      Sub ButtonClick(sender as object, e as EventArgs)
```

```

'Validates if the login info entered is correct
Dim objConnection as OleDbConnection
Dim cmdSelect as OleDbCommand
Dim dtrdValidateLogin as OleDbDataReader
Dim strUserName as string
Dim strPassword as string
dim test as integer
dim employeeID as string
dim empStatus as string
dim empPassword as string

'Setting variables to the text box values on the login form
strUserName = txtUsername.text
strPassword = txtPassword.text

objConnection = new
OleDbConnection(ConfigurationSettings.AppSettings("connString"))

cmdSelect = new OleDbCommand("Select * from
EmployeeInfo where Username=@Username and Passwords=@Passwords",
objConnection)

cmdSelect.Parameters.Add("@Username", strUserName)
cmdSelect.Parameters.Add("@Passwords", strPassword)

```

```
objConnection.Open()
```

```
dtrdValidateLogin = cmdSelect.ExecuteReader()
```

'Testing if the authentication information exists in the
database -- Notice that it is case sensitive

```
While dtrdValidateLogin.read()
```

```
    employeeID = dtrdValidateLogin("EmployeeID")
```

```
    empStatus = dtrdValidateLogin("Classification")
```

```
    'empPassword = dtrdValidateLogin("Password")
```

```
    test = 1
```

```
end While
```

```
dtrdValidateLogin.Close()
```

```
objConnection.Close()
```

'If they did not enter valid user/password combo, display
error, otherwise procede to next screen

```
if test = 0 then
```

```
    lblLoginResults.text = "Incorrect
```

```
Username/Password"
```

```
else
```

```
    'Response.Cookies("firstname")="Alex"
```

```

Dim cookie As HttpCookie = New
HttpCookie("uid")

cookie.Value = employeeID
Response.Cookies.Add(cookie)

'Dim newCookie As HttpCookie = New
HttpCookie("EmpID")

'newCookie.Values.Add("IDnumber", employeeID)

dim brian as string
brian = Request.Cookies("uid").Value
'brian = Request.Cookies("EmpID")("IDnumber")

session("employeeID") = employeeID 'saving the
employee id to a session variable so it can later be accessed in the order screens

'depending on their status (management, sales, or
both) they will get their appropriate screen

if empStatus = "Management" then
    response.redirect("mangementMenu.aspx")
end if

if empStatus = "Sales" then
    response.redirect("employeeMenu.aspx")
end if

if empStatus = "Both" then
    response.redirect("bothMenu.aspx")

```

```
end if
```

```
end if
```

```
End Sub
```

```
</script>
```

A 2. View Sales by Date Code Snippet

The code below is used to show all of the sales transactions for each date. When a user clicks on a certain date in the drop down menu, the code checks the database and displays the total sales for the selected date.

```
<title>viewSalesByDate</title>
```

```
<meta content="Microsoft Visual Studio .NET 7.1"
```

```
name="GENERATOR">
```

```
<meta content="Visual Basic .NET 7.1" name="CODE_LANGUAGE">
```

```
<meta content="JavaScript" name="vs_defaultClientScript">
```

```
<meta content="http://schemas.microsoft.com/intellisense/ie5"
```

```
name="vs_targetSchema">
```

```
<script language="vb" runat="server">
```

```
Sub ViewSales(sender as object, e as EventArgs)
```

```
    'Displays all items sold for a particular date or all dates
```

```
    Dim objConnection As OleDbConnection
```

```
    Dim objCmd As OleDbCommand
```

```
    Dim strConnection As String
```

```

        strConnection =
ConfigurationSettings.AppSettings("connString")

        objConnection = New OleDbConnection(strConnection)

        objConnection.Open()

        objCmd = New OleDbCommand()

        objCmd.Connection = objConnection

        if dropSales.SelectedItem.Value = "All" then

            'using a predefined query to display this information
            objCmd.CommandText = "[orderByAnyDate]"

            objCmd.CommandType =
CommandType.StoredProcedure

            listSales.DataSource =

objCmd.ExecuteReader(CommandBehavior.CloseConnection)

            listSales.DataBind()

            objConnection.close()

        else

            'using a predefined query to display this information
            objCmd.CommandText = "[orderByDate]"

            objCmd.CommandType =
CommandType.StoredProcedure

```

```

        objCmd.Parameters.Add("@odate",
dropSales.SelectedItem.Value)

        listSales.DataSource =
objCmd.ExecuteReader(CommandBehavior.CloseConnection)

        listSales.DataBind()
        objConnection.close()

    end if

    listSales.visible = true

end sub

Sub ButtonClick3(sender as object, e as EventArgs)
    'sends user back to the management Menu
    response.redirect("mangementMenu.aspx")
end sub

Sub Page_Load()
    if Not IsPostBack then
        Dim strConnection as string =
ConfigurationSettings.AppSettings("connString")

        Dim objConnection as new
OleDbConnection(strConnection)

        Dim cmdSelect as OleDbCommand
        Dim dtrItemSales as OleDbDataReader
        Dim strSelectStatement as string

```

```
objConnection.Open()

'Populating the list box with the dates of all the sales
strSelectStatement = "SELECT DISTINCT
Orders.OrderDate FROM Orders"

cmdSelect = new
OleDbCommand(strSelectStatement, objConnection)

dtrItemSales = cmdSelect.ExecuteReader()
dropSales.Datasource = dtrItemSales
dropSales.DataTextField = "OrderDate"
dropSales.DataValueField = "OrderDate"
dropSales.Databind()
dtrItemSales.Close()
dropSales.items.insert(0,"All") 'add the option of
seeing all dates for items sold

objConnection.Close()

end if

end sub

</script>
```

References

1. Chase Information Systems. "Pre-Packaged Information Systems (Database) pricing" <http://www.chaseinfosystems.com/doc4.htm> . September 2001.
2. Computer Works. "Database Systems - Product Pricing" http://www.computerworks.bc.ca/database_pricing.htm . 2003.
3. Kimmel, Paul. "Visual Basic® .NET Unleashed". January 17, 2002
4. Stephens, Rod. "Visual Basic® .NET Database Programming". April 11, 2002.
5. White, Andrew. "Designing Web Interfaces". April 24, 2002.