VISULIZATION OF A RESTAURANT MANAGEMENT

DATA ANALYSIS

USING JAVA TECHNOLOGIES

By

XIAN S. TAN

Bachelor of Science

Nanjing University

Jiangsu Province, People’s Republic of China

1985

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2008

VISULIZATION OF A RESTAURANT MANAGEMENT

DATA ANALYSIS

USING JAVA TECHNOLOGIES

Thesis Approved:

George E. Hedrick

Thesis Adviser

John P. Chandler

Johnson Thomas

A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

I wish to express my deepest appreciations to my advisor Dr. George Hedrick for being my major advisor. His enthusiastic support, guidance, encouragement, and friendship throughout the completion of the thesis are memorable in my life. Without his support, guidance, encouragement and help, I could not imagine that I would finish the thesis. I also want to extend my appreciations to Dr. John Chandler and Dr. Johnson Thomas for their teaching and their idea that led to the selection of the thesis topic, their advisement, suggestions, and their valuable help toward the completion of this thesis.

I also want to express my appreciations to all the teachers in the computer science department. Without their teaching, this thesis would have been without a solid foundation of the theory and practice of the science of computing.

I would like to thank my dear husband for his unconditional love, understanding, support and encouragement.

I would like to thank my mother for her caring for my small son, and her unconditional support not only during my education, but in my entire life. I would say without her help, the thesis would be impossible. I also would like to thank my daughter’s understanding of my devotion of time toward the thesis.

Finally, I would like to praise God for providing me with confidence, wisdom, persistence and strength that guided me through the entire process.

Table of Contents

List of Figures

Chapter 1 Introduction

Data mining is the process of finding important hidden information or patterns in large databases to help companies for making decisions. It predicts future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions [1, 6, 7, 9, 10, 11]. Visualization is an important technology for data mining. Data mining and visualization also can answer business questions that traditionally were too time-consuming to resolve [9, 11]. They search databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations [9]. We restrict our study of data mining in relational databases to a small restaurant management system.

With the progress of computer hardware and software in the past decades, much data has been collected by businesses and stored in computers. Industry leaders use data mining to analyze data from different perspectives, then summarize it into useful information that can be used to increase revenue, cut costs, or both. The business owner uses the data to calculate the revenue for the last several years and to answer questions; such as, “What are the total sales in the organization for the last several years and/or the last several months” [9].

Data mining has been used extensively for the last decade in industries, such as merchant transactions, the heath care industry, the banking system, telecommunications, human resource information management, and many other areas [8, 9]. However, there are few publications about the use of data mining for small businesses mainly for reasons such as the expense of data collection, maintaining and analyzing the data. Very few small businesses collected and stored business transaction data. Now more and more small businesses have computers to manage their business transactions. If they make the effort, they can use data mining to lower their business costs, make better customer relationships, and increase their profit.

Thearling [9] gave the definition of data mining as, “Data mining is, in some ways, an extension of statistics, with a few artificial intelligence and machine learning twists added.” As with statistics, data mining is not a business solution, it is just a technology. For example, a restaurant owner needs to decide who should receive information about a new product and menu. If he/she uses an advertising company to do advertising in a certain zip code, the cost is much higher than mailing directly to the customers. The information mined is contained in a historical database of previous interactions with customers and the features associated with the customers, such as their zip codes and their responses. Data mining software uses this historical information to build a model of customer behavior that could be used to predict which customers would be likely to respond to the new product and menu. By using this information, a marketing manager can select the customers who are most likely to respond to save advertising costs.

There has been much progress in data mining technology in the past decade due to the advancement of the internet so that information can be distributed, viewed, and collected through the internet. The greater amount of data available and the increase in computing power on the desktop make it possible to do more data mining research and development.

The most commonly used techniques in data mining are summarized briefly as follows [1, 3, 6, 7, 9, 11].

Artificial Neural Networks

Artificial Neural Networks are non-linear predictive models that learn through training and resemble biological neural networks in structure.

Decision Trees

Decision Trees are tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification And Regression Trees (CART) and Chi Squared Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques that are used for the classification of a dataset. They provide a set of rules that one can apply to a new (unclassified) dataset to predict which records will have a given outcome. CART segments a dataset by creating 2-way splits while CHAID segments using Chi Squared tests to create multi-way splits. CART typically requires less data preparation than CHAID.

Genetic Algorithms

Genetic algorithms are optimization techniques that use process such as genetic combinations, mutations, and natural selection in a design that are based on the concepts of evolution.

Nearest Neighbor Method

The Nearest Neighbor Method is a technique that classifies each record in a dataset based on a combination of the classes of the k record(s) most similar to it in a historical dataset. Sometimes it is called the k-nearest neighbor technique.

Rule Induction

The extraction of useful “if-then” rules based on statistical significance is called rule induction.

Data Visualization

The graphic presentation of complex relationships is called data visualization. Graphical tools are used to illustrate data relationships [2, 3]. Pictures can convey information far more succinctly than textual descriptions. For example, a bar chart graph gives comparison of the numbers, and this is more straight-forward than describing each number. A pie chart graph gives a comparison of the percentage distribution, and this is more straight-forward than describing each percentage. There are many other graphs that can be used in data mining. In a broad sense, a graphical presentation is more than only descriptions in sentences. A tabular listing can be considered part of a graphical presentation. A tabular listing can be a concise list of columns of data and their relationships. A graphical presentation can also include captions and legends. Thus a graphical presentation may include graphs for data, diagrams for relationships, and listings of text information in tabular forms, or mixture of all three.

Web-based Reporting and Data Mining

A very popular architecture for today's businesses is to use the internet (HTML) as the front-end because of the ease of deploying software – this requires very little effort because of having a thin-client, as opposed to the client-server architecture which may require deploying software to every client computer [8]. The latter examples include Java Applications and VB Applications. There are numerous examples of successful business application that use the internet-based architecture, because it is flexible and easy to use.

A web-based Data Mining Application can use HTML as the basic presentation tool. It can have drill-down web pages. In a drill-down web page, a summary report is displayed at the beginning. The links inside the summary report lead to more detailed reports. Each detailed report can have more links to further more detailed reports. A detailed report can also have multiple links to less detailed and/or summary level reports. Such web pages are also called "drillable" or "drill-down" web pages. The user can modify the reporting criteria on-the-fly to obtain different reports. Common examples of such a reporting system is called “Ad-Hoc” reporting and/or an “Executive Dashboard” reporting. In an “Ad-Hoc” reporting, additional reports are created to investigate certain data report behaviors. In “Executive Dashboard” reporting, high level summary tables and charts are presented at the beginning and detailed reports are accessed by clicking on the links in the dashboard. While “Ad-Hoc” is more focused on the detail and/or particular set of filtering and reporting options, the “Executive Dashboard” is more focused on the higher level statistics which are suitable of executives and/or managers [14]. As far as a small business is concerned, a data mining application might be a mixture of both “Ad-hoc Reporting” and “Dashboard Reporting”, because of limited programming resources and narrower research extent. In addition to the typical statistics, data mining should be able to answer significant numbers of business questions such as: what do we do with the statistics? And where is the information in the data? How do we use data mining to improve our advertising, products, and services? How do we have better business intelligence than our competitors? With the help of business data modeling, it becomes easier to apply the knowledge we obtained from the old data to new business situations.

Java is an object-oriented programming language that is particularly suitable for the internet because the Java code is versatile across many operating systems. Complex objects, relationships, and algorithms can be implemented with Java Technology, and the Java Technology can be written once and then be used in many operating systems of computers.

There are publications about the use of Java in data mining [11, 12]. The Weka system provides a rich set of powerful Machine Learning algorithms for Data Mining tasks [12]. Witten and Zdravko's [11, 12] publications primarily discussed the Machine-Learning algorithms, the Java implementation of the algorithms, and the data visualization using Java. They are good tools for learning data mining concepts and algorithms.

A business data mining system is more than a set of algorithms. A data mining system with a fixed set of algorithms will not be enough to provide all the necessary information for a business. The end user should not have to be an expert in using the particular algorithms. Basic and complex reports with tabular listings and graphical presentations must be developed, in addition to algorithms. New algorithms can be implemented if necessary. Sometimes, information presentation, visualization, navigation and distribution are more essential than algorithms. Therefore, a business data mining and visualization system should be able to provide the necessary information and the intelligence for the execution and the improvement of businesses.

Based on the above considerations, we decided to develop a data mining and visualization application using the Java programming language to demonstrate the basic Java Programming techniques applicable to real business situations.

Focus of Thesis: Data Mining for A Restaurant Management

We owned a small Chinese restaurant, and were involved with all aspects of restaurant management. The savings of advertising dollars and maximizing profit both are very important. Advertising cost is a big expense for a small restaurant, but if we want to keep good business, customers must have the restaurant's menu available whenever they want to order Chinese food. We want to keep track of the customer's ordering history and to know how well we served the customers in order to improve our products and services. We want to make and keep good relationships with the customers so that they keep coming back to us. We need daily, weekly, monthly, and quarterly statistics and annual sales performance and cost analysis so that we know how well we are doing in our business and we should be able to adjust our efforts accordingly. We must monitor the raw material usage on a sale's basis to keep enough food supply, yet not to over-stock so we keep the food fresh. Therefore, having an automated management and data mining system is very important for a small business such as a small restaurant.

This thesis describes the building of a web based Data Mining tool. This tool is easily accessible by authorized users without having to distribute a significant amount of software. This web based application is easy to use by just pointing and clicking.

HTML is used to present the data so that the authorized user who has a web browser can view the data reports through the web. Therefore, the author of this paper thinks it is much better than the client-server architecture. It is more flexible that it does not require client software installation and deployment. We choose Oracle as the database because it is a very popular database for data access performance and ease of use through programming with JDBC and ODBC [15, 16]. In addition to HTML, we use Java Sever Page to access the database and to present the statistical results on the web.

Compared with the techniques mentioned earlier, this thesis is relevant to the graphing and presentation of information. It shows examples of using the Java Programming language to build a highly interactive and drillable Web application to mine and explore a relational database system for a restaurant management system. Similar programming techniques can be applied to other small businesses. The thesis demonstrates how to use Java programming to build a data mining system that can be used in real life to save small businesses advertising money and to improve customer relationships. It demonstrates Java programming using Graphical User Interface (Netbeans IDE) tool to build Java GUI Applications and to do Web Programming with Apache Tomcat JSP and Java Servlets. The data presentation and visualization demonstrate that the Java Programming Language and Server technology are effective for visualization and data mining. The example programming code can be used to teach others, to program functional Java Applications for a real life business.

Chapter 2 Building Database System For Restaurant Management

2.1 Use Oracle as Database to Store Restaurant Management Data

In order to do data mining, we must have sufficient data, and for our purposes the data is best stored in a relational database system (RDBMS) for easy querying and programming. Other data formats such as text files, binary files, or other non-RDBMS formats cannot be accessed using Structured Query Language (SQL).

The Windows Version of the Oracle 9I Database can be downloaded free of charge, and the Oracle databases are well established for communicating with Java programs. The Oracle 9I database can be downloaded from the Oracle Technology Website [17]:

http://www.oracle.com/technology/software/index.html

A download requires signing-in with a user profile in order to down-load the software. There are three files to download. The installation files are extracted to a directory, and then the setup starts.

The Oracle 9I database was installed onto a personal computer in the D:\program files\Oracle\ora92 directory, where D: is the system drive. The instance of the database was named “DATAEXP”, meaning “Data Explorer”.

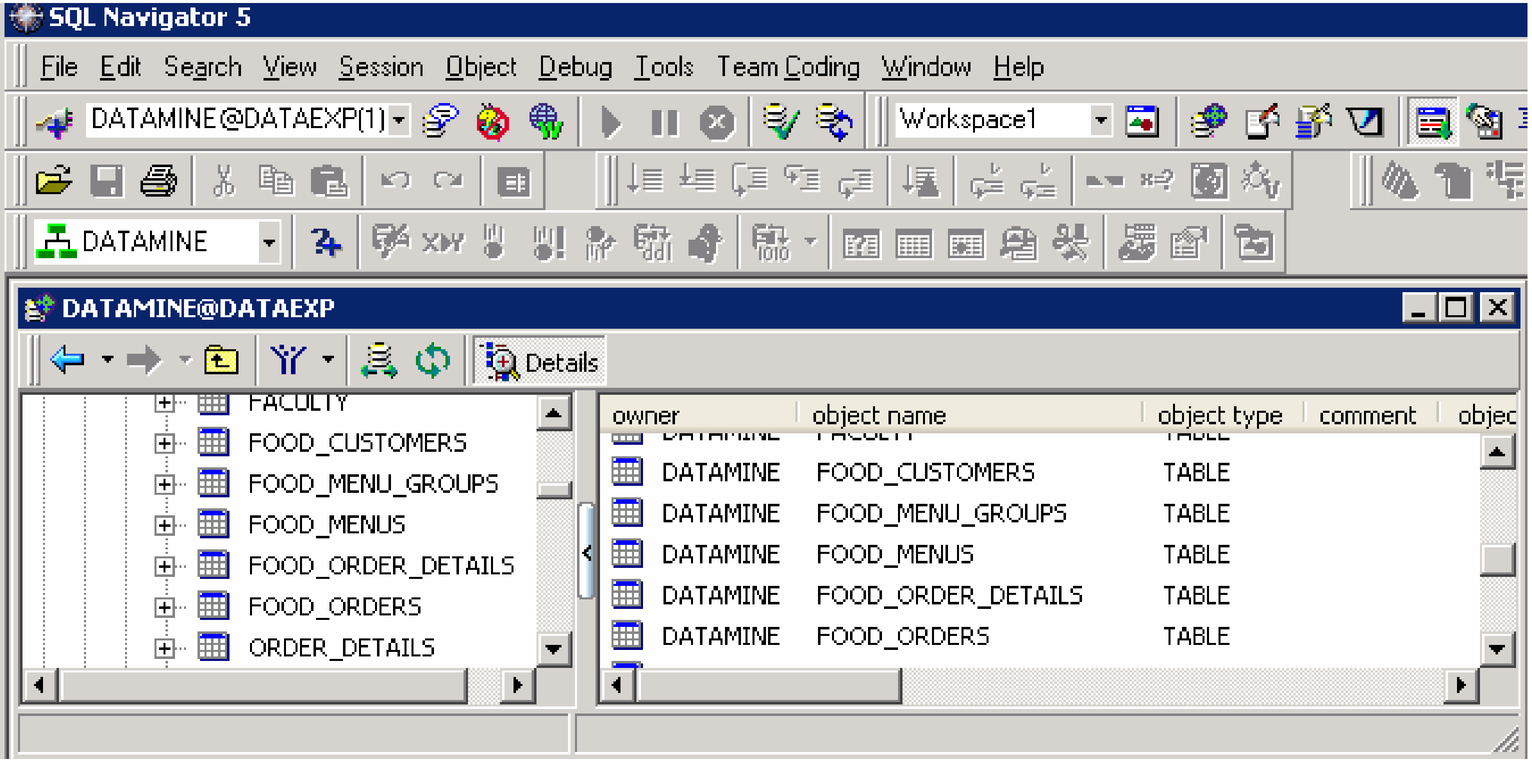
We use SQL navigator [18] as the developer’s tool for working with Oracle databases. This is a User-Interface tool which allows easy navigation, editing and issuing SQL commands, and analyzing results from the SQL queries. The SQL navigator is a better tool to work with Oracle databases than the Oracle's Command-line interface, called SQL-Plus. The SQL navigator provides a user interface that allows navigation and the use of browsing database objects such as Windows Explorer, and allows creating SQL statements using a user-friendly editor, much like notepad for cutting, copying and pasting. The SQL-plus command line interface is much like a DOS command window, requiring much command typing and it is not as convenient as SQL navigator.

2.2 Creating Database Tables for Data-Mining Study

A database schema is a logical grouping of database objects with the same prefix. A database schema is usually owned by a particular user account, yet other user accounts might have the ability to create, modify and/or access the objects within the schema.

The user-account "DATAMINE" stores objects relevant to the data mining study. All the objects created are within the "DATAMINE" schema. Figure 2.1 shows the database tables as seen from SQL navigator.

The primary tables in the data mining study follow, considering the focus of the study is food ordering patterns.



FACULTY
FOOD_CUSTOMERS
FOOD_MENU_GROUPS
FOOD_MENUS
FOOD_ORDER_DETAILS
FOOD_ORDERS
ORDER_DETAILS

owner	object name	object type	comment	objec
DATAMINE	FOOD_CUSTOMERS	TABLE		
DATAMINE	FOOD_MENU_GROUPS	TABLE		
DATAMINE	FOOD_MENUS	TABLE		
DATAMINE	FOOD_ORDER_DETAILS	TABLE		
DATAMINE	FOOD_ORDERS	TABLE		

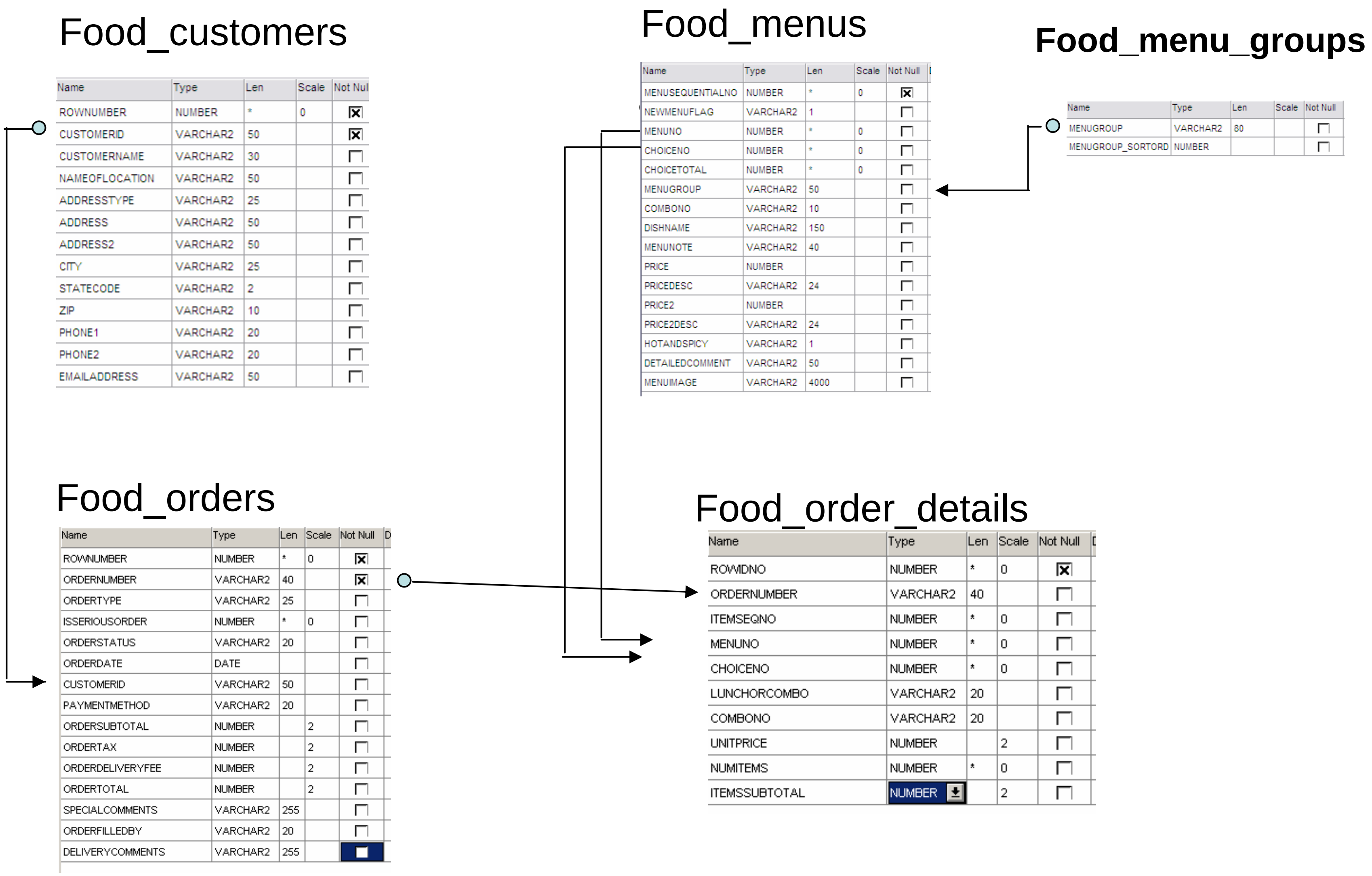
Figure 2.1 Database Tables for the Schema "DATAMINE",
used for the purpose of Data Mining study. Use “SQL Navigator” as the tool
To browse database objects and to design SQL Queries.

Food_menus	- Menu Items
Food_menu_groups	- Menu Item Groups
Food_customers	- Customer Names and Contact Info
Food_orders	- Customer Food Orders
Food_order_details	- Customer Food Order Detail (Menu Item Ordered)

Figure 2.2 is an ERD (Entity Relationship Diagram), showing the tables and their relationships in the Restaurant Management System. A primary key designates a unique value for each row in a table for a given data field. A foreign key designates a child-relationship to the parent table which has the corresponding field.

The Food_customers table stores the customer contact information. The CustomerId primary key in the Food_customers table is a foreign key for the Food_orders table, which stores food orders. Each customer may make many orders. The OrderId primary key in the Food_orders table is a foreign key for the Food_orders_detail tables, meaning that one order may have many menu items ordered. The primary key “MenuId” in the Food_menus tables is also a foreign key in the “Food_order_details” table. The Food_menu_groups table store the food menu group information. The field “menuGroup” is a foreign key for the “Food_menus” table to classify food menu into groups such as “Appetizers”, “Sea Food”, “Lunch Special”, “Combo”, etc.

The above 5 tables are sufficient to describe the basic food ordering processes and thus will be used for the data mining study for the Restaurant Management System.



Food_customers

Name	Type	Len	Scale	Not Nul
ROWNUMBER	NUMBER	*	0	☒
CUSTOMERID	VARCHAR2	50		☒
CUSTOMERNAME	VARCHAR2	30		□
NAMEOFLOCATION	VARCHAR2	50		□
ADDRESSTYPE	VARCHAR2	25		□
ADDRESS	VARCHAR2	50		□
ADDRESS2	VARCHAR2	50		□
CITY	VARCHAR2	25		□
STATECODE	VARCHAR2	2		□
ZIP	VARCHAR2	10		□
PHONE1	VARCHAR2	20		□
PHONE2	VARCHAR2	20		□
EMAILADDRESS	VARCHAR2	50		□

Food_menus

Name	Type	Len	Scale	Not Null
MENUSEQUENTIALNO	NUMBER	*	0	☒
NEWMENUFLAG	VARCHAR2	1		□
MENUNO	NUMBER	*	0	□
CHOICENO	NUMBER	*	0	□
CHOICETOTAL	NUMBER	*	0	□
MENUGROUP	VARCHAR2	50		□
COMBONO	VARCHAR2	10		□
DISHNAME	VARCHAR2	150		□
MENUNOTE	VARCHAR2	40		□
PRICE	NUMBER			□
PRICEDESC	VARCHAR2	24		□
PRICE2	NUMBER			□
PRICE2DESC	VARCHAR2	24		□
HOTANDSPICY	VARCHAR2	1		□
DETAILEDCOMMENT	VARCHAR2	50		□
MENUIMAGE	VARCHAR2	4000		□

Food_menu_groups

Name	Type	Len	Scale	Not Null
MENUGROUP	VARCHAR2	80		□
MENUGROUP_SORTORD	NUMBER			□

Food_orders

Name	Type	Len	Scale	Not Null
ROWNUMBER	NUMBER	*	0	☒
ORDERNUMBER	VARCHAR2	40		☒
ORDERTYPE	VARCHAR2	25		□
ISSERIOUSORDER	NUMBER	*	0	□
ORDERSTATUS	VARCHAR2	20		□
ORDERDATE	DATE			□
CUSTOMERID	VARCHAR2	50		□
PAYMENTMETHOD	VARCHAR2	20		□
ORDERSUBTOTAL	NUMBER		2	□
ORDERTAX	NUMBER		2	□
ORDERDELIVERYFEE	NUMBER		2	□
ORDERTOTAL	NUMBER		2	□
SPECIALCOMMENTS	VARCHAR2	255		□
ORDERFILLEDBY	VARCHAR2	20		□
DELIVERYCOMMENTS	VARCHAR2	255		□

Food_order_details

Name	Type	Len	Scale	Not Null
ROWIDNO	NUMBER	*	0	☒
ORDERNUMBER	VARCHAR2	40		□
ITEMSEQNO	NUMBER	*	0	□
MENUNO	NUMBER	*	0	□
CHOICENO	NUMBER	*	0	□
LUNCHORCOMBO	VARCHAR2	20		□
COMBONO	VARCHAR2	20		□
UNITPRICE	NUMBER		2	□
NUMITEMS	NUMBER	*	0	□
ITEMSSUBTOTAL	NUMBER		2	□

Figure 2.2 Tables and Entity Relationships (ERD) For the Restaurant Management Database System. The small dots denote primary keys. The small arrows denote the corresponding foreign keys.

The tables are created with the SQL navigator, either by using the graphical table creation editor, or by issuing the SQL command (DDL) such as below:

```
CREATE TABLE food_menus
          (
           menusequentialno           NUMBER(*,0) NOT NULL,
           newmenuflag                VARCHAR2(1),
           menuno                     NUMBER(*,0),
           choiceno                   NUMBER(*,0),
           choicetotal                NUMBER(*,0),
           menugroup                  VARCHAR2(50),
           combono                    VARCHAR2(10),
           dishname                   VARCHAR2(150),
           menunote                   VARCHAR2(40),
           price                      NUMBER,
           pricedesc                  VARCHAR2(24),
           price2                     NUMBER,
           price2desc                 VARCHAR2(24),
           hotandspicy                VARCHAR2(1),
           detailedcomment            VARCHAR2(50),
           menuimage                  VARCHAR2(4000)
          )
```

2.3 Java Application for Data Entry

The data entry can be programmed as either a Java Application (executable GUI program) or as Web-Application (Web Pages). We programmed a Java application for data entry. The programming tasks for developing a Java Application (GUI) can be significantly different from developing a Java Web-based Application (Web Pages). The Java application code (executables and supporting class files) must be deployed onto each individual computer in order to run the Java program on that particular computer. One advantage of using a Java application is that all the codes remain on the local computer. On the other hand, for a Web-based application, no code needs to be distributed to the end users's computer. The end users enter the web address in a browser in order to use the web application. End users may be able to see some HTML and JavaScript code from the browser.

2.4 NetBeans IDE For Developing Java Applications

We use NetBeans IDE to develop the Java Application for data entry. This is an Integrated Development Environment for developing Java applications and Java Web pages. It provides the basic programming framework and allows the drawing of graphical interfaces using the IDE, thus simplifying the programming tasks.

The NetBeans IDE can be downloaded from Sun MicroSystems's WebSite [19]:

http://java.sun.com/j2se/1.4.2/download-netbeans.html

There are other versions of the software available from the same site.

After downloading, the software Java2 v.1.4.2_16 SDK and the NetBeans IDE are installed by executing the setup executable for windows environment. The following environment will also be setup through Windows “Control Panel” – “System” – “Advanced” – “Environment Variables”:

JAVA_HOME: c:\ j2sdk1.4.2_16

PATH: c:\ j2sdk1.4.2_16\bin

A Java Program communicates with a database through JDBC (Java Database Connectivity).

In order for the NetBeans IDE to communicate with the Oracle 9I database, through JDBC, make sure the following Oracle Jar Files have been added into NetBeans 5.0 Classpath – In NetBeans 5.0, click on tools, Library Manager, Classpath Tab, Add Jar/Folder:

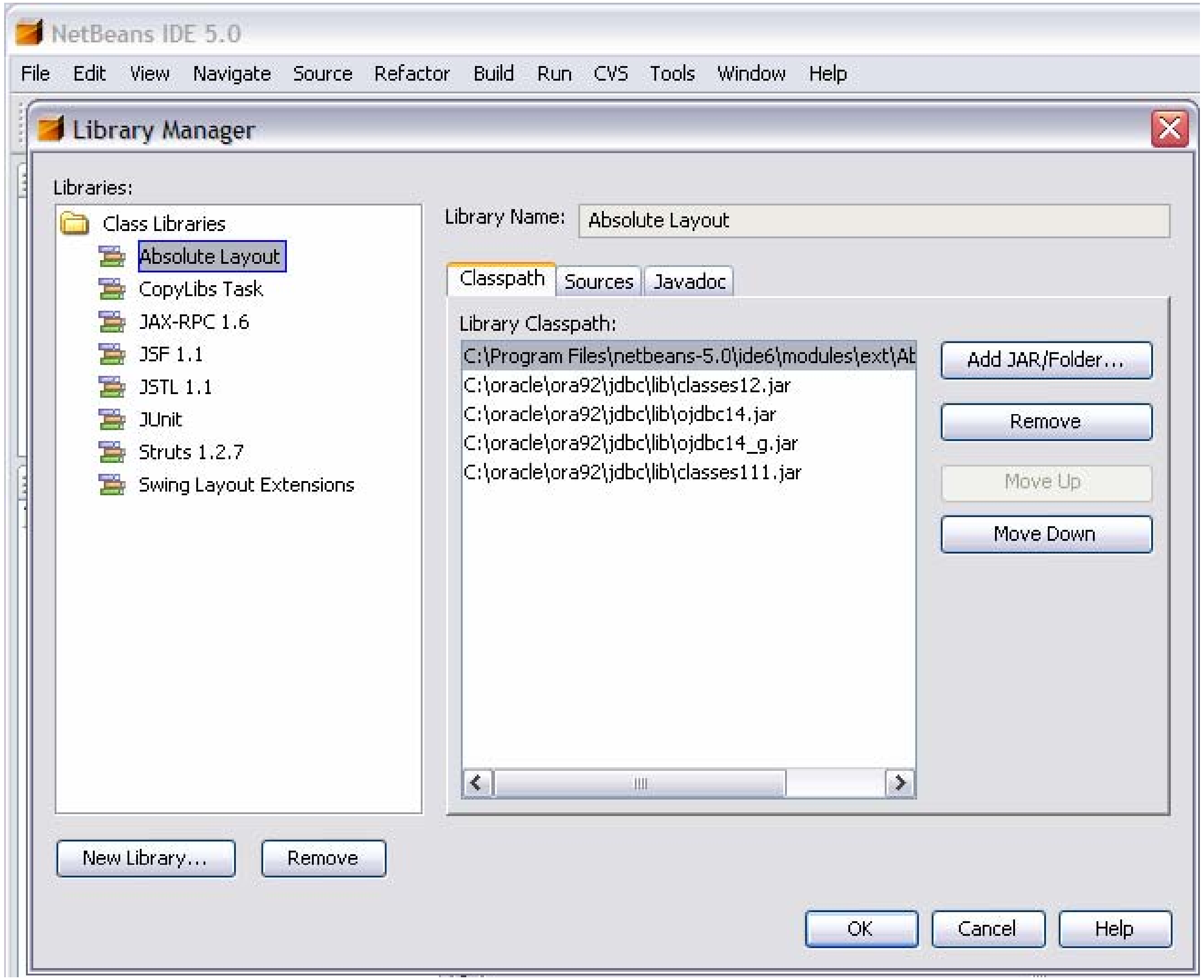
C:\oracle\Ora92\jdbc\lib\classes12.jar

C:\oracle\Ora92\jdbc\lib\classes14.jar

C:\oracle\Ora92\jdbc\lib\classes14_g.jar

C:\oracle\Ora92\jdbc\lib\classes111.jar

The NetBeans “CLASSPATH” for the JDBC Connection to Oracle 9I Database is as shown in Figure 2.3.



Libraries:

- Class Libraries
  - Absolute Layout
  - CopyLibs Task
  - JAX-RPC 1.6
  - JSF 1.1
  - JSTL 1.1
  - JUnit
  - Struts 1.2.7
  - Swing Layout Extensions

Library Name: Absolute Layout

Library Classpath:

```
C:\Program Files\netbeans-5.0\ide6\modules\ext\Ab
C:\oracle\ora92\jdbc\lib\classes12.jar
C:\oracle\ora92\jdbc\lib\ojdbc14.jar
C:\oracle\ora92\jdbc\lib\ojdbc14_g.jar
C:\oracle\ora92\jdbc\lib\classes111.jar
```

Figure 2.3 NetBeans IDE 5.0 "CLASSPATH" for communicating with Oracle Database through JDBC (Java Database Connectivity).

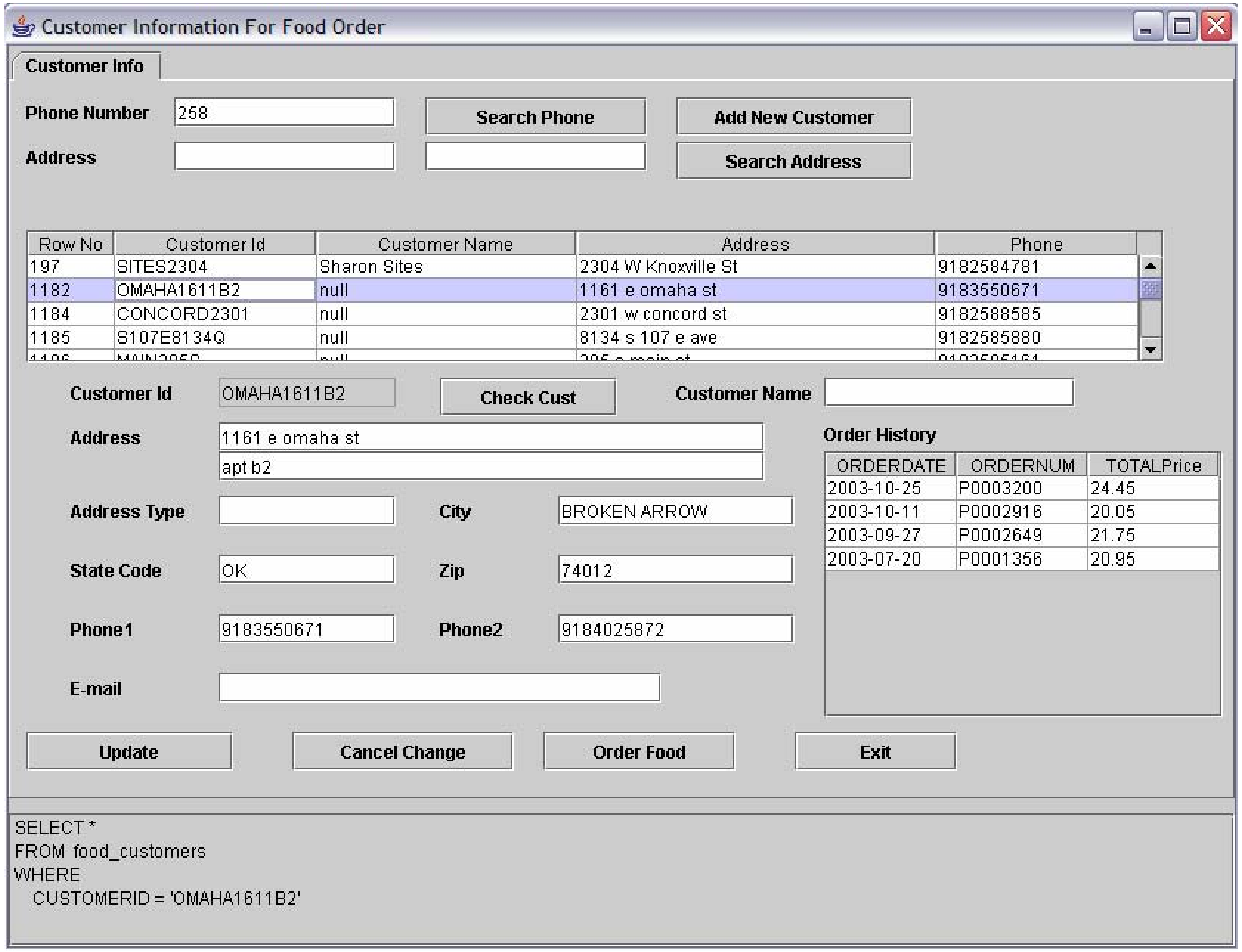
2.5 Creating Java Application Program (GUI) for Data Entry

The Java application consists of the following 3 forms for data entry: the “Customer Information for Food Order”; the “Food Menu and Order Items” form for selecting menu items and quantities; the “Check Out” form for Order Totals and payments.

Customer Information Form

The first form is the “Customer Information for Food Order” form (Figure 2.4). In this form, partial phone number and/or partial address can be searched. If a matching address and/or phone exist, then the address is listed in the JTable below the search boxes.
The reason for using partial phone search and partial address search is that most customers are repeating customers. Searching for customer contact information speeds the process for entering customer information and preventing errors in data entry. The phone search also can be connected to the caller-id from the phone lines to reduce the entry of customer information to the minimum. The phone order-taker only needs to verify the caller’s phone number, and then enter full or partial phone number to initiate the search.

Any number of digits can be entered in the “Phone Number” textbox. For the “Search Phone” button, add an event procedure “jbtnSearchCustomerActionPerformed” to the “Action Performed” event to respond to the clicking of the button. The event function



Phone Number: 258

Address:

Row No	Customer Id	Customer Name	Address	Phone
197	SITES2304	Sharon Sites	2304 W Knoxville St	9182584781
1182	OMAHA1611B2	null	1161 e omaha st	9183550671
1184	CONCORD2301	null	2301 w concord st	9182588585
1185	S107E8134Q	null	8134 s 107 e ave	9182585880

Customer Id: OMAHA1611B2

Customer Name:

Address: 1161 e omaha st
apt b2

Address Type:

City: BROKEN ARROW

State Code: OK

Zip: 74012

Phone1: 9183550671

Phone2: 9184025872

E-mail:

Order History

ORDERDATE	ORDERNUM	TOTALPrice
2003-10-25	P0003200	24.45
2003-10-11	P0002916	20.05
2003-09-27	P0002649	21.75
2003-07-20	P0001356	20.95

```
SELECT *
FROM  food_customers
WHERE
   CUSTOMERID = 'OMAHA1611B2'
```

Figure 2.4 Customer Information for Food Order form: Searching customer by partial phone and partial address. If matching customers are found, they will be listed in the JTable below the search boxes. Clicking on a customer will list the customer contact information and customer order history. The "Add New Customer" button allows adding a new customer. If customer information is modified, then click "Update" to update customer information. Click "Order Food" to proceed to entering food order.

uses the partial phone number to construct the SQL to get the data from the database to populate the “JTable” for customer list.

Similar to the phone search, any partial address (street number, street name, etc.) can be entered in the two address boxes. An event function “jbtnAddressSearchActionPerformed” is added to respond to the clicking of the “Search Address” button. The function uses the partial addresses to construct the SQL to get the data from the database to populate the “JTable” for customer list. Upon clicking on an item in the “JTable” customer list, the detailed customer information is displayed in the lower part as individual textboxes. To update customer information, we type in the updated customer address and phone number in the textboxes, then click on the “Update” button. An “ActionPerformed” event function is used to actually update the customer information in the database.

Data Mining Feature

Anyone who has ever used amazon.com [13] to buy or search for a book has had the experience of seeing additional information beyond the title of the book that satisfies the search. Other relevant books and statistics about similar books are displayed below the book that is the object of the search. The past search and order history are also displayed along side this information. By supplying such additional information, customers usually are more satisfied and end up buying more books and/or keep coming back. This is a very good example of using data mining in e-commerce web sites. Many web sites, such as bestbuy.com and officedepot.com, also offer customer feed back about the products purchased and the delivery experience.

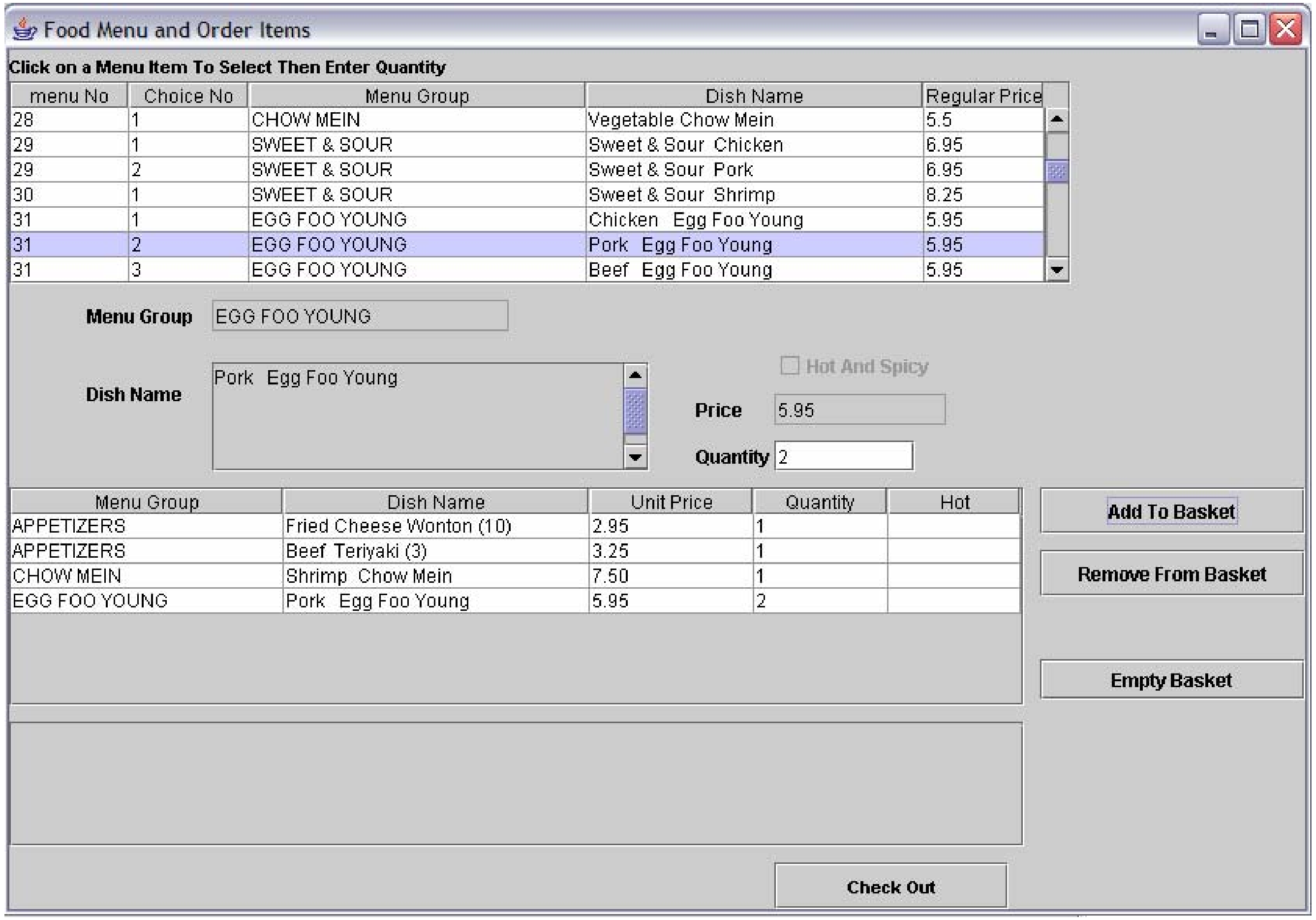
For a restaurant ordering and management system, we can use similar data mining techniques. In the food orders form, the customer’s order history is displayed in the lower right corner, listing the order dates and amounts. Delivery notes and customer service notes are automatically displayed in the “Customer Service Notes” tab upon clicking a customer in the list. This allows the phone person to serve the customer better, to make better and faster recommendation for customers, and to prepare food faster and more accurately to meet the customer’s requirements, and to improve delivery timeliness.

If a customer’s information cannot be found in the system, then we use “Add New Customer” button to enter new customer information. The “Save” button in stead of the “Update” button saves customer information.

We click on the “Order Food” button from the “Customer Information for Food Order” to enter the “Food Menu and Order Items” form.

Food Menu and Order Items Form

The second form is the “Food Menu and Order Items” form (Figure 2.5). The top part is a JTable to list the available menu items. The order taker clicks on a menu item in the list to select a menu item. The order taker enters the quantity for the item selected, then the



Click on a Menu Item To Select Then Enter Quantity

menu No	Choice No	Menu Group	Dish Name	Regular Price
28	1	CHOW MEIN	Vegetable Chow Mein	5.5
29	1	SWEET & SOUR	Sweet & Sour Chicken	6.95
29	2	SWEET & SOUR	Sweet & Sour Pork	6.95
30	1	SWEET & SOUR	Sweet & Sour Shrimp	8.25
31	1	EGG FOO YOUNG	Chicken Egg Foo Young	5.95
31	2	EGG FOO YOUNG	Pork Egg Foo Young	5.95
31	3	EGG FOO YOUNG	Beef Egg Foo Young	5.95

Menu Group: EGG FOO YOUNG

Dish Name: Pork Egg Foo Young

□ Hot And Spicy

Price: 5.95

Quantity: 2

Menu Group	Dish Name	Unit Price	Quantity	Hot
APPETIZERS	Fried Cheese Wonton (10)	2.95	1	
APPETIZERS	Beef Teriyaki (3)	3.25	1	
CHOW MEIN	Shrimp Chow Mein	7.50	1	
EGG FOO YOUNG	Pork Egg Foo Young	5.95	2	

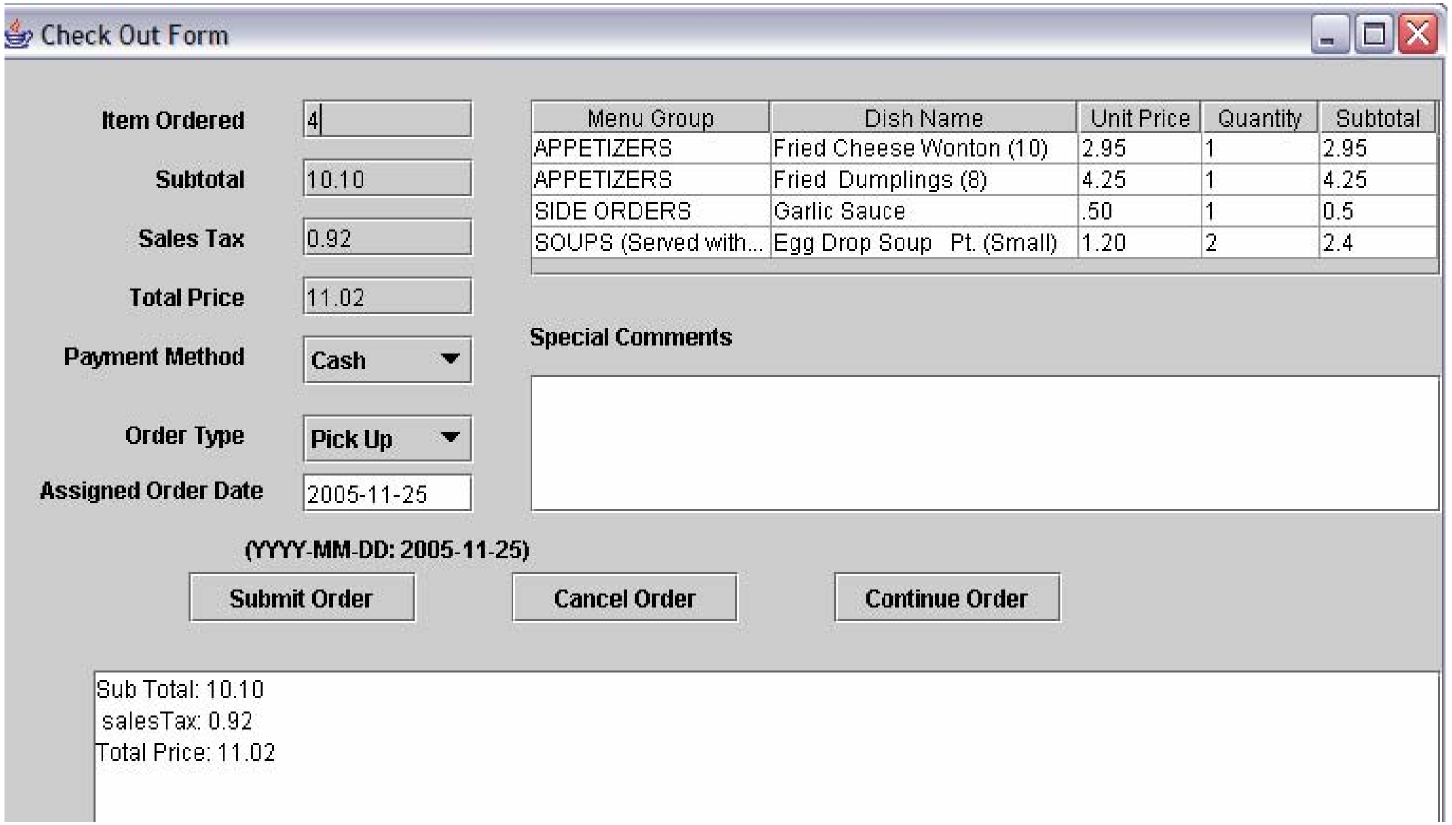
Figure 2.5 Food Menu and Order Items. The top part is a JTable to list the available menu items. Click on a menu item in the list to select a menu item. Change the quantity for the item selected. Click on the "Add To Basket" button to add the menu item to the entered order items list. If an order item needs to be edited, then click in the order items list, and either update quantity or click on the "Remove From Basket" to remove the item. To remove all items, click on the "Empty Basket" button. Click on the "Check Out" button to proceed to check out.

order taker clicks on the "Add To Basket" button to add the menu item to the entered order items list. If an order item needs to be edited, then the order taker clicks in the order items list, and either updates quantity or clicks on the "Remove From Basket" to remove the item. When it is necessary to remove all items, he/she clicks on the "Empty Basket" button. The order taker clicks on the "Check Out" button to proceed to check out.

Check-out Form

The check-out form (Figure 2.6) includes the ordered items (jTable list), total count of item, subtotal, tax amount, total, payment method, and order type (pick up or delivery), and order date.

A "Special Comments" field allows entering special comments such as customer requirements for specific food items, delivery instructions and comments about how to improve customer service. When a customer decides to continue to order more items, then the order taker presses the "Continue Order" button. When customer decides to complete the order, then the order taker presses the "Submit Order" button.



Check Out Form

Item Ordered	4
Subtotal	10.10
Sales Tax	0.92
Total Price	11.02
Payment Method	Cash
Order Type	Pick Up
Assigned Order Date	2005-11-25

(YYYY-MM-DD: 2005-11-25)

Menu Group	Dish Name	Unit Price	Quantity	Subtotal
APPETIZERS	Fried Cheese Wonton (10)	2.95	1	2.95
APPETIZERS	Fried Dumplings (8)	4.25	1	4.25
SIDE ORDERS	Garlic Sauce	.50	1	0.5
SOUPS (Served with...	Egg Drop Soup Pt. (Small)	1.20	2	2.4

Special Comments

```
Sub Total: 10.10
 salesTax: 0.92
Total Price: 11.02
```

Figure 2.6 Check-out form. This form includes items ordered, subtotal, tax, total price payment method, order type, and order date. In the right side, a JTable lists the items ordered in brief detail to allowed final review of the food items ordered. A "Special Comments" field allows to enter special comments such as customer requirements for specific food items, delivery instructions and comments about how to improve customer service. When customer decides to continue to order more items, then press the "Continue" button. When customer decides to complete the order, then press the "Submit Order" button.

Summary of Data Entry

With the above data entry forms, we have captured most of the necessary information for our data mining purposes (customer information, food menus and order items, payments, deliveries, and comments). By displaying data mining features such as search function, customer order history, relevant food, and customer service notes, customers can be served better and greater customer satisfaction can be achieved.

Chapter 3 Data Mining Statistics and Web-based Reporting For A Restaurant Management System Using Java

Upon the capture of business transactional data, the next task is to discover the hidden patterns and information inside the data. With the advent of the internet and web technology, information distribution has become more convenient and more easily accessible without having to install a lot of software on an end user’s computer. Java is a programming language that is particularly suitable for developing web applications such that the applications will run in many different operating systems without having to modify and/or to recompile the programs.

This chapter focuses on the fundamentals of programming a data mining application using the Java programming language. It includes software architecture for a web-based reporting system, using Java to write web-based reports, using Java to program interactive and dynamic contents, and using Java programming to do statistical computations. Real examples for enabling the data mining for a restaurant management system result from the Java programs.

Software Architecture for Web-based Reporting System Using Java

The typical web-based reporting system using Java technology is a web server (HTTP Server) with Java Server Pages (JSP) and Java Servlet.

A Java server page (JSP Page) is an HTML page with additional Java statements embedded between the “<%” and the “%>” tags, much like ASP (ASP.NET) and/or other server-side pages. A servlet is a Java class located in the servlet directory that has enabled “http request” and “http response” methods. The servlet generates HTML code to return to the web browser. A JSP Page is translated into a servlet before being executed by the Servlet Engine.

Appendix A documents the detail of Apache Tomcat 5.05 Configuration. Tomcat serves [20] both JSP and Servlet Webpages. In this thesis project, Tomcat is configured to work through Port 8082 on the Web-Server PC, which can be referenced as “localhost” or a specific computer name, such as “roger”. The data mine application is configured as a “Context” named “datamine”. Therefore, any JSP page for the data mining application has a web-address such as

http://localhost:8082/datamine/any_jsp_web_page_name.jsp

A simple JSP page example is located at

c:\datamine\simpleExample.jsp

and the corresponding URL is

http://localhost:8082/datamine/simpleExample.jsp

The source code of “simpleExample.jsp” is included in Appendix B.

A JSP page does not need to be compiled by the developer. A JSP page is automatically compiled before being executed by Internet Explorer.

A simple Servlet example is located at

```
  C:\datamine\WEB-INF\classes\SimpleServletExample.java
```

With the corresponding URL

http://localhost:8082/datamine/SimpleServletExample.java

The source code of "SimpleServletExample.java" is included in Appendix D.

The Java Servlets are compiled at command prompt [21], such as:

```
C:\
  cd  C:\datamine\WEB-INF\classes\

Javac SimpleServletExample.java
```

After a servlet is compiled successfully, then it can be executed from internet explorer.

Popular JSP/Servlet Engines include BEA Web-Logic, IBM WebSphere, and Apache Tomcat, etc. Tomcat is open-source software and is free for use in any operating system, whereas, other JSP/Servlet Engines might be proprietary and require licenses and payments. Therefore, Apache Tomcat is the JSP/Servlet Engine for this study.

JSP Example with Database Access

An example JSP page with database access to the Oracle 9I database DATAEXP is located at

C:\datamine\simpleJSPExampleWithDatabase.jsp

With the corresponding URL

http://localhost:8082/datamine/simpleJSPExampleWithDatabase.jsp

The source code of the JSP page "simpleJSPExampleWithDatabase.jsp" is listed in Appendix C.

Servlet Example with Database Access

An example Servlet page with database access to the Oracle 9I database DATAEXP is located at

C:\datamine\WEB-INF\classes\SimpleServletExampleWithDatabase.java

With the corresponding URL

http://localhost:8082/datamine/servlet/SimpleServletExampleWithDatabase

The source code of the Servlet page "SimpleServletExampleWithDatabase.java" is listed in Appendix E.

Reporting in Tabular Form

One way to present data mining results (data reports) is to report information in tabular form, where data is presented in rows and columns along with additional statistical values and labels. A simple database example is presented earlier in the section titled "**JSP**

Example With Database Access”. To summarize, the general structure for presenting the results from a SQL query is as the following:

(a) Connect to database through JDBC connection;

(b) Create dynamic SQL statements to obtain data from the database;

(c) Create a “ResultSet” from the SQL;

(d) Loop through the “ResultSet” and present data in HTML Table, using “<TABLE>”, “<TR>”, and “<TD>” tags.

We can include multiple SQL query results in a JSP page using HTML tables. In addition, multiple JSP Pages can work together to accomplish the tasks of a data mining application.

In this study, the HTML tabular reports are drillable. That is, a summary report is displayed at the beginning. When a link inside the report is clicked, a more detailed report is displayed. When a link in the more detailed report is clicked, a further detailed report is displayed. There are also links in the reports that allow getting back to less detailed and yet summary level reports. It is easy to allow navigation from summary to detail and/or from detail to summary. Drillable HTML reports are especially suitable for data mining applications. Multiple levels of drillable reports are implemented in this study.

Graphs (Visualization) of Data

Java2D API is a graphics library provided by Sun Microsystems to do Java graphing. The Java2D API is available in J2SE 1.4.2 and above to draw 2-dimensional graphs using the Java programming language. An introduction to the Java2D API [22] is located at

http://java.sun.com/docs/books/tutorial/2d/TOC.html

A sample JSP page for a 2D graph is developed using Java2D API, and the source code is located in

C:\datamine\simpleJava2DGraph.jsp

The corresponding URL is

http://localhost:8082/datamine/simpleJava2DGraph.jsp

The source code for the 2D graph JSP page is listed in Appendix F.

The following lines declare a graphics context with an Image of (350, 250) pixels.

```
BufferedImage bi = new BufferedImage(350, 250, BufferedImage.TYPE_INT_RGB);
Graphics2D biContext = bi.createGraphics();
```

The following lines draw a color filled rectangle

```
biContext.setColor(Color.cyan);
Rectangle rect0 = new Rectangle(0, 0, 350, 250);
biContext.fill(rect0);
```

The following lines draw a straight line

```
GeneralPath p = new GeneralPath();
p.moveTo(40, 40);
p.lineTo(150, 180);
```

The following lines draw a text string

```
java.awt.Font font;
font = new java.awt.Font("times", 0, 20);
biContext.setFont(font);
biContext.drawString("Sample Java2D Text", 30, 100);
```

The following lines write the Image into a Graphics file

```
String ifile = "samplejava2d.jpg";
File f = new File("c:\\datamine\\graphs\\" + ifile);
ImageIO.write(bi, "png", f);
```

The following HTML code displays the image file in the WebBrowser

```
<IMG SRC="/datamine/graphs/samplejava2d.jpg" ALT="Sample Java2D Graph">
```

More complicated and complete Java2D graphs can be created using the above basic rectangle fills, line drawings, text drawings and other Java2D API calls. The data for data mining can be read from the database with dynamically constructed SQL statements based on the particular summary reporting requirements, such as Group-By conditions and totals. Then a graph which represents the statistical results is created to show the data in graphical form, that is, images.

Restaurant Data Mining Web Application

The Restaurant Data Mining Web Application is a JSP application. The layout of the main application is shown in Figure 3.1. On the left side are the main menus. On the center and to the right are the main display areas that are implemented as “Imbedded Frame” (<IFrame> tags in HTML).

When a user clicks on one of the main menu items on the left, a separate JSP Page is loaded into the IFrame, through the JavaScript Function “navigateToPage (‘nameofthejsppage.jsp’)” to set the SRC property of the <IFrame> to the JSP Page to be displayed.

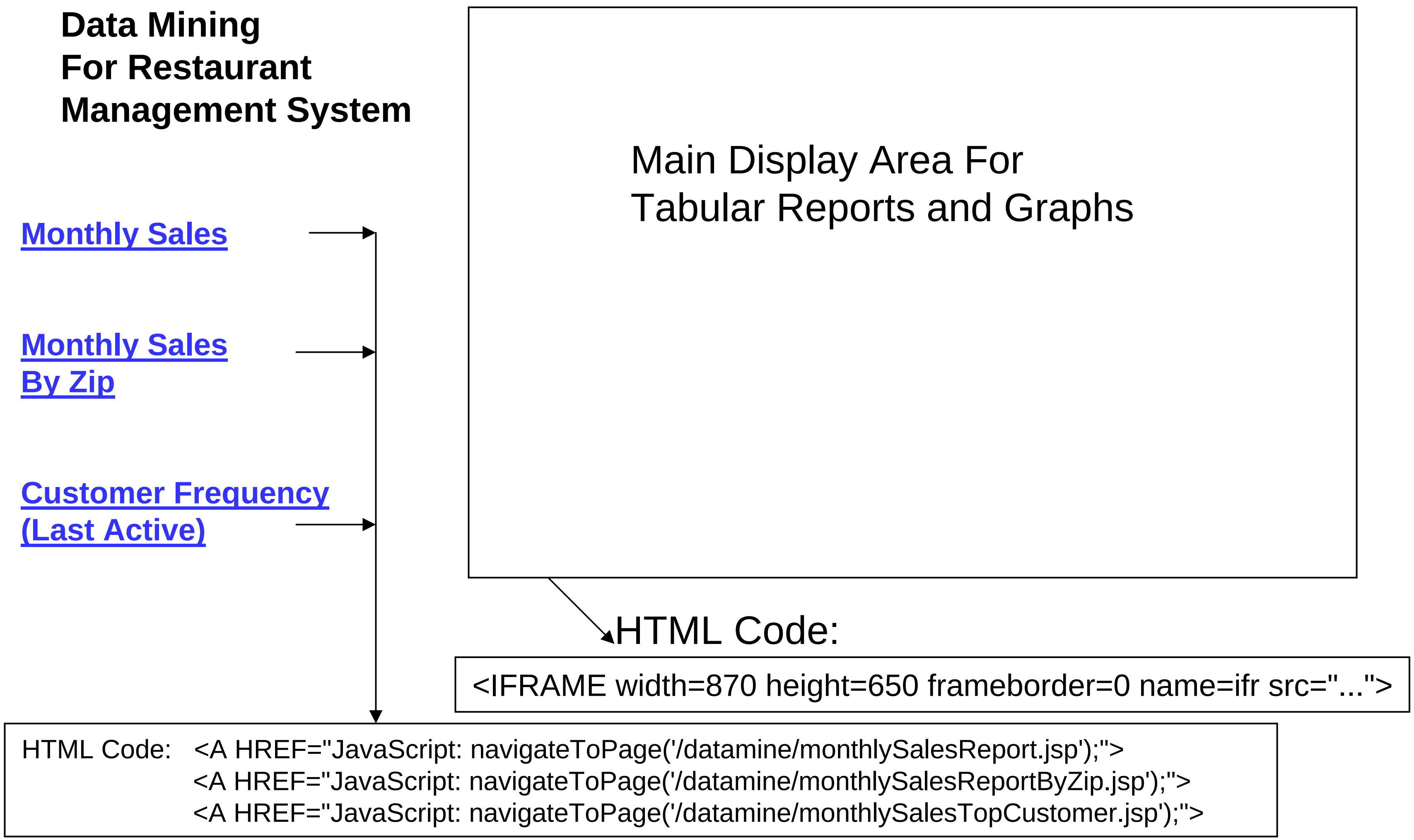
The main JSP page for the datamine application has the following URL:

http://localhost:8082/datamine/restaurant_datamine.jsp

The JSP Program resides in c:\datamine\ directory.

By default, the main page also loads the monthly sales report (“monthlySalesReport.jsp”) into the <IFrame>, through the <Body onLoad=”…”> tag.

In the main JSP page, the right side is initially defined by the <SPAN> tag:



Data Mining
For Restaurant
Management System

Monthly Sales

Monthly Sales
By Zip

Customer Frequency
(Last Active)

Main Display Area For
Tabular Reports and Graphs

HTML Code:

```
<IFRAME width=870 height=650 frameborder=0 name=ifr src="...">
```

```
HTML Code:   <A HREF="JavaScript: navigateToPage('/datamine/monthlySalesReport.jsp');">
             <A HREF="JavaScript: navigateToPage('/datamine/monthlySalesReportByZip.jsp');">
             <A HREF="JavaScript: navigateToPage('/datamine/monthlySalesTopCustomer.jsp');">
```

Figure 3.1 Layout of the main JSP Page for the Restaurant Data Mining application.
http://localhost:8082/datamine/restaurnt_datamine.jsp
Left side are the main menu items. Right side is an <Iframe> which displays subpages for the data mining application in an Imbedded Frame.

```
<SPAN id=spanPageDetail>  </span>
```

In the JavaScript function “navigateToPageNow(pageURL)”, the “<IFrame>” is dynamically generated through the JavaScript statement:

```
        document.all.spanPageDetail.innerHTML = "<IFRAME width=870 height=650
frameborder=0 name=ifr src=\"" + pageURL + "\"></iframe>";
```

The advantage of dynamically generating <IFrame> is that the parameters of the <IFrame> such as width, height, border, and URL can be changed flexibly each time when a new Page URL is to be loaded.

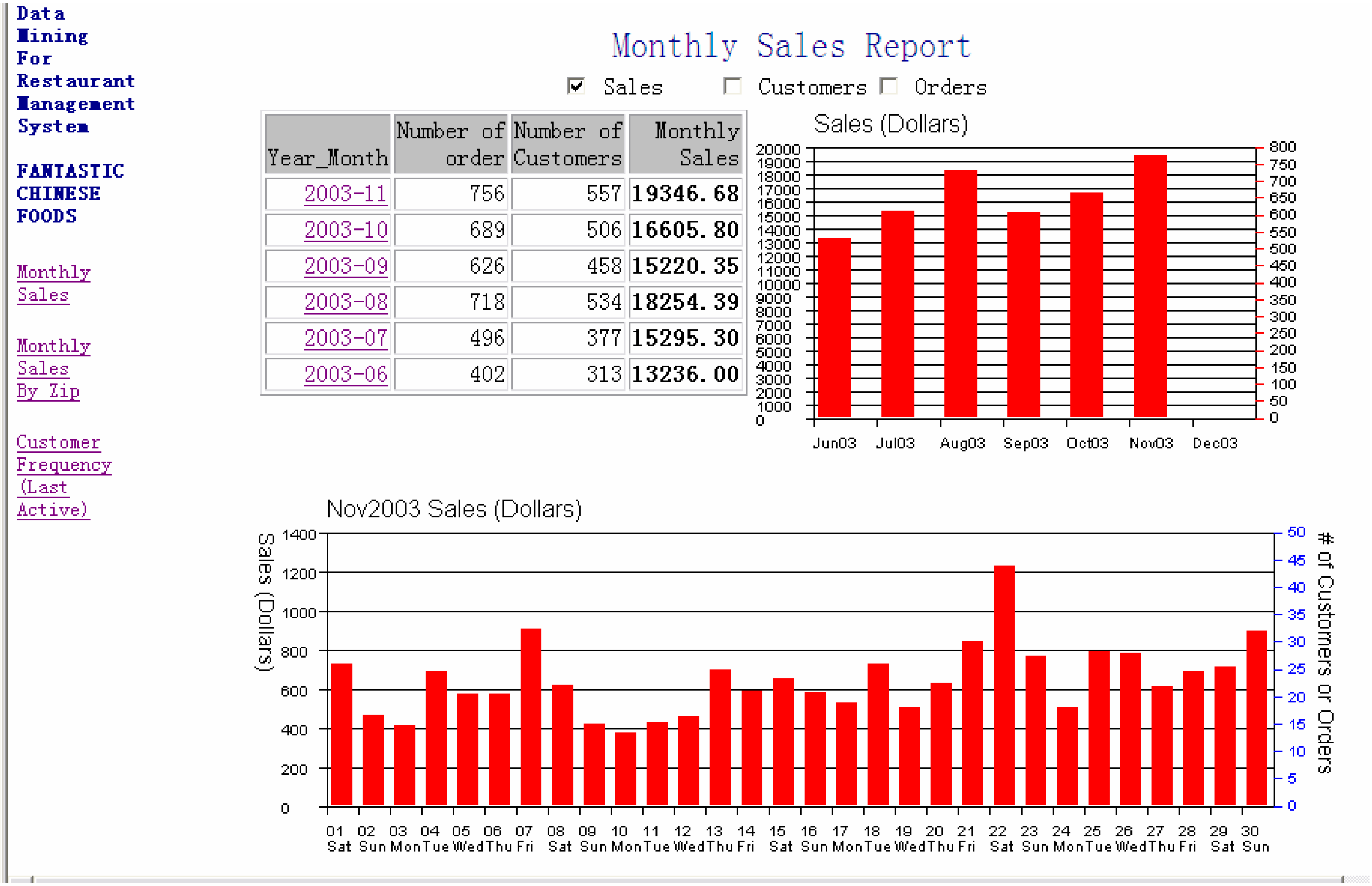
Monthly Sales Report

The “Monthly Sales Report” (Figure 3.2) has the URL:

http://localhost:8082/datamine/monthlySalesReport.jsp

```
     page URL =   “http://localhost:8082/datamine/monthlySalesReport.jsp”;
     document.all.spanPageDetail.idataminennerHTML = "<IFRAME width=870 height=650
frameborder=0 name=ifr src=\"" + pageurl + "\"></iframe>";
```

On the left of the report is an HTML Table with the monthly sales totals for each month. On the right is the corresponding bar-chart graph for each month’s sales. The link for each month is clickable, and when it is clicked, a detailed graph for the daily sales for the month is displayed at the bottom.



Data Mining For Restaurant Management System

FANTASTIC CHINESE FOODS

Monthly Sales

Monthly Sales By Zip

Customer Frequency (Last Active)

Monthly Sales Report

☑ Sales ☐ Customers ☐ Orders

Year_Month	Number of order	Number of Customers	Monthly Sales
2003-11	756	557	**19346.68**
2003-10	689	506	**16605.80**
2003-09	626	458	**15220.35**
2003-08	718	534	**18254.39**
2003-07	496	377	**15295.30**
2003-06	402	313	**13236.00**

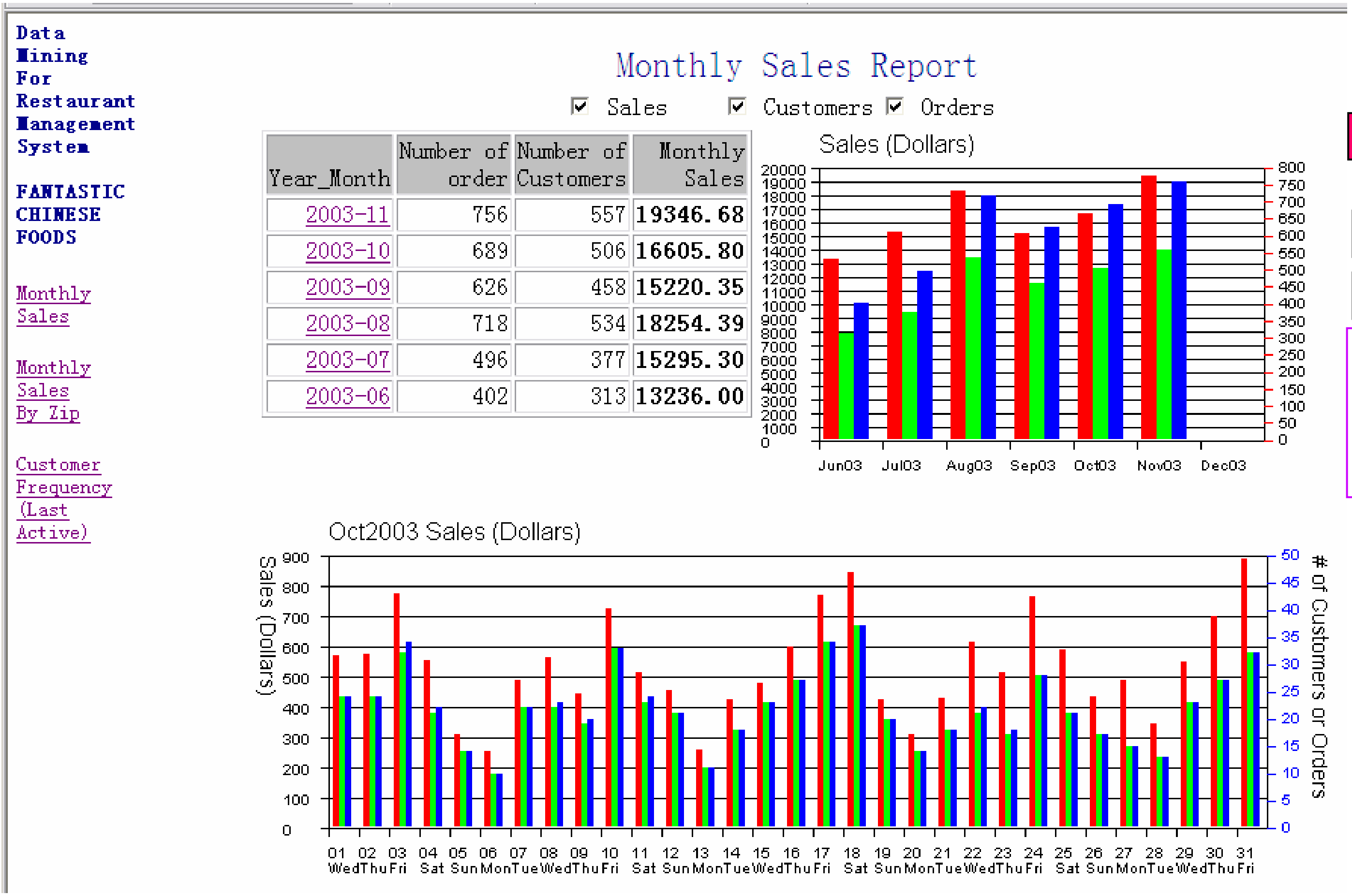
Figure 3.2 Monthly sales report. On the left of the report is an HTML Table with the monthly sales totals for each month. On the right is the corresponding. The link for each month is clickable, and when it is clicked, a detailed graph for the daily sales for the month is displayed at the bottom. In addition, when the "Customers" and/or "Orders" checkboxes are checked, the number of customers and orders are displayed side-by-side in the graph along with the sales total.

From the monthly sales report, we can see the comparison of the sales for each month, the daily total sales.

Figure 3.3 shows additional features for the “Monthly Sales Report”. The number of customers and the number of orders are displayed side-by-side in the graph along with the monthly sales dollar amounts. The sales dollars uses the vertical axis on the left, whereas the number of customers and the number of orders use the vertical axis on the right. Any combinations of "Sales", "Customers" and "Orders" can be displayed, by selecting the checkboxes and then clicking on a month in the left.

The side-by-side graph in Figure 3.3 shows the correlation between number of orders and customers, versus the total sales dollars. Days near the weekend (Thursday, Friday, Saturday) have higher sales dollar amounts, and numbers of customers and orders. Therefore, more food and more service personnel are required for those days in order to have better customer service.

Because it has the flexibility to click on any month to display the monthly sales report, it is easy to review any month’s sales volumes (sales dollar, customers, and orders) from the report. Therefore, the restaurant management knows the progress for each month’s sales and service volumes.



Data Mining For Restaurant Management System

FANTASTIC CHINESE FOODS

Monthly Sales

Monthly Sales By Zip

Customer Frequency (Last Active)

Monthly Sales Report

☑ Sales ☑ Customers ☑ Orders

Year_Month	Number of order	Number of Customers	Monthly Sales
2003-11	756	557	**19346.68**
2003-10	689	506	**16605.80**
2003-09	626	458	**15220.35**
2003-08	718	534	**18254.39**
2003-07	496	377	**15295.30**
2003-06	402	313	**13236.00**

🟥 **Sales**

🟩 **Customers**

🟦 **Orders**

The greater the difference, the more of repeat customers.

Figure 3.3 Monthly sales report with additional features. Number of customers and number of orders are displayed side-by-side in the graph along with the monthly sales dollar amounts. Any combinations of "Sales", "Customers" and "Orders" can be displayed, by selecting the checkboxes and then click on a month in the listing on the left.

Monthly Sales By Zip Code

The “Monthly Sales by Zip Code” report (Figure 3.4) has the URL:

http://localhost:8082/datamine/monthlySalesReportByZip.jsp

The link from the main JSP page (restaurant_datamine.jsp) is:

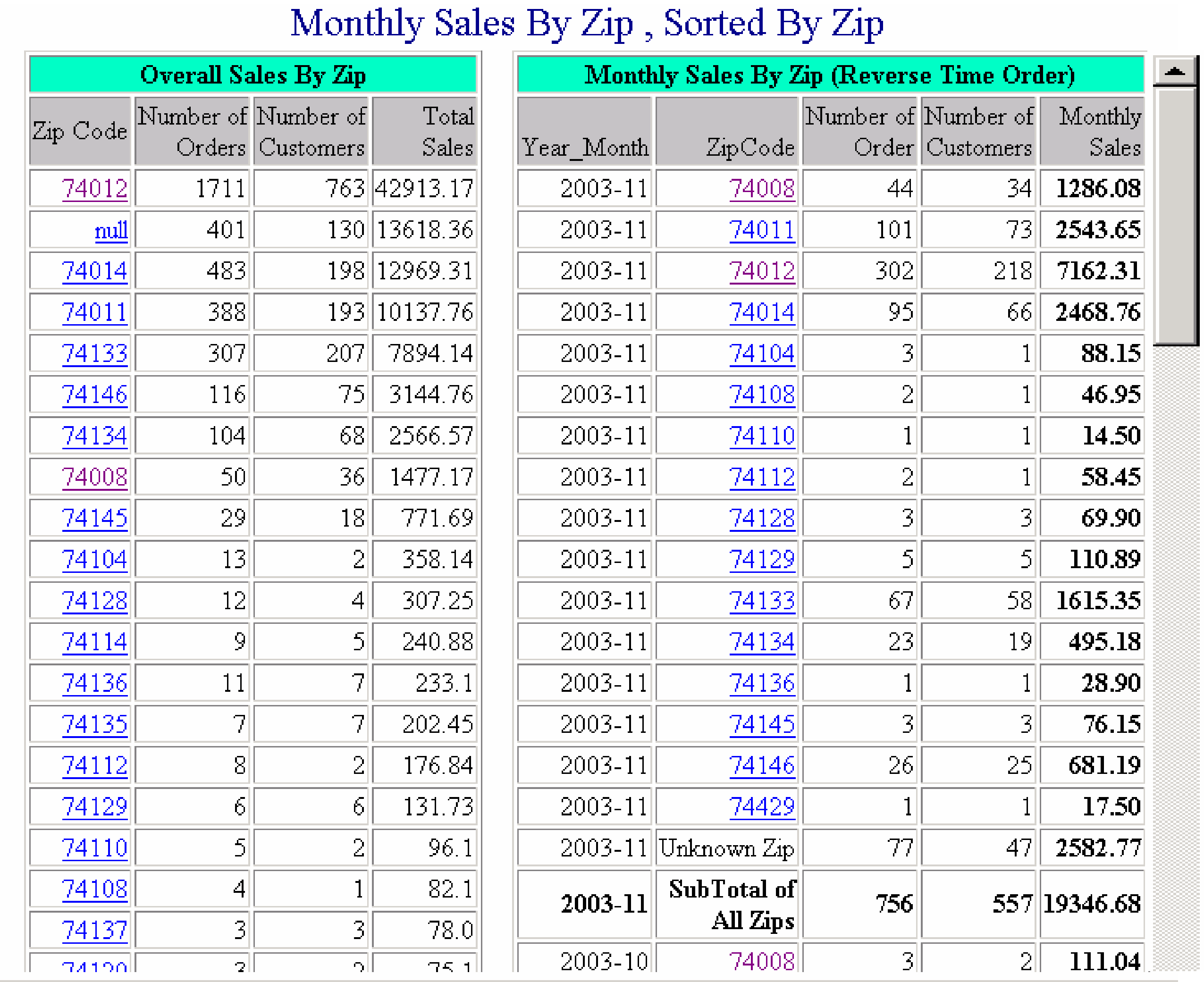
```
<A HREF=”JavaScript: navigateToPage(‘/datamine/monthlySalesReportByZip.jsp’);”>Monthly Sales
B y Zip</A>
```

The report is loaded from the main page into the <IFrame> through JavaScript:

```
PageURL =   “http://localhost:8082/datamine/ monthlySalesReportByZip.jsp ”;
document.all.spanPageDetail.innerHTML = "<IFRAME width=870 height=650 frameborder=0
name=ifr src=\"" + pageurl + "\"></iframe>";
```

The “Monthly Sales By Zip” report is displayed as in Figure 3.4. On the left side, the overall sales for each zip code is listed. On the right, the total sales for each zip code for each month is listed. Scroll down to see other months and other zips. Each zip code is clickable to display details (graphs) for the sales for the selected zip code. From the listing, the user can decide which zip code to navigate to look for details, based on the overall sales dollars and number of orders on the left, or based on the sales dollars and number of orders in any specific month on the right.

When the user clicks on a given “Zip Code”, a detailed report by zip code is displayed (Figure 3.5). This detailed report is again drillable. When the user clicks on a month, the

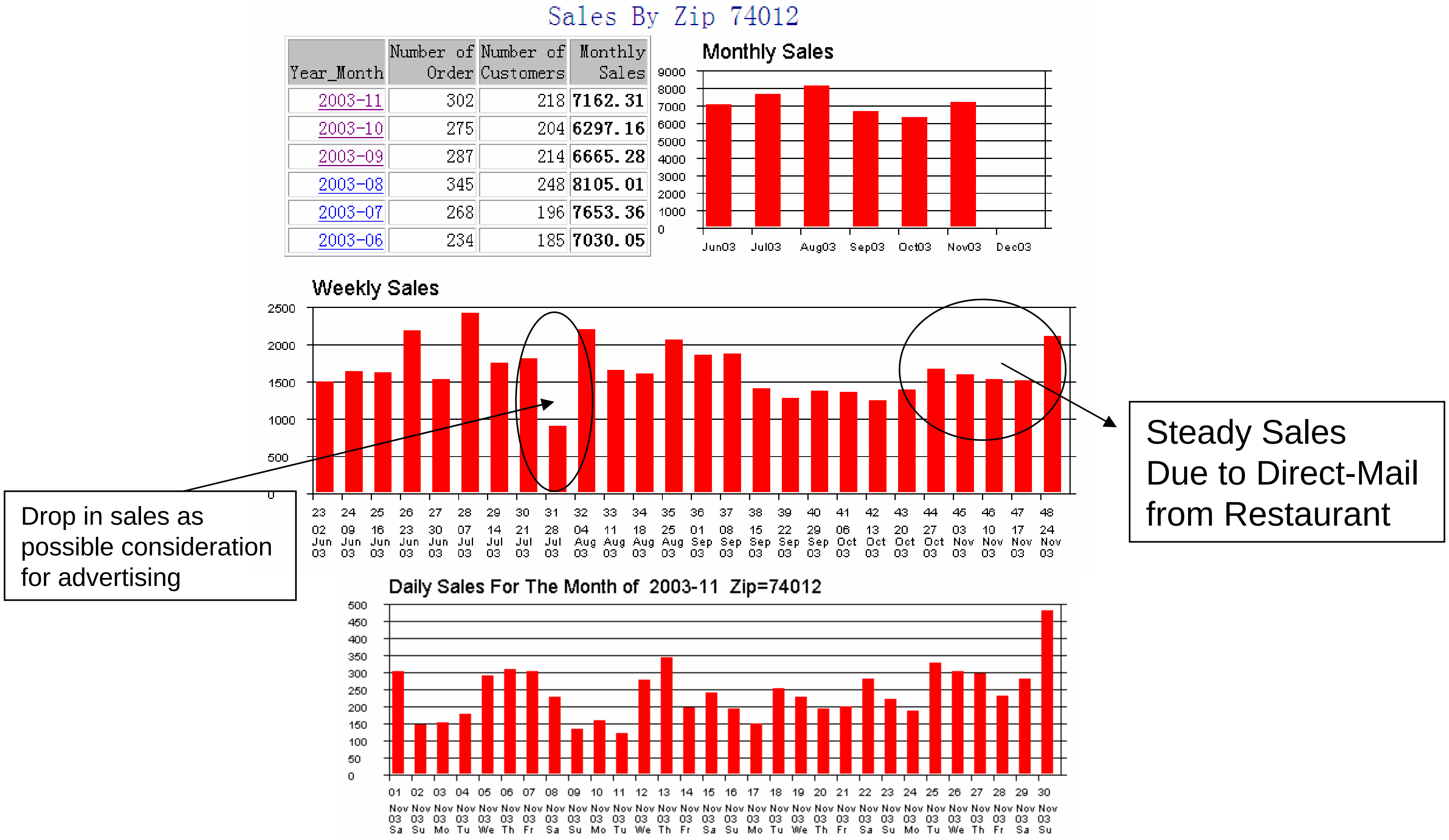


Monthly Sales By Zip , Sorted By Zip

Overall Sales By Zip			
Zip Code	Number of Orders	Number of Customers	Total Sales
74012	1711	763	42913.17
null	401	130	13618.36
74014	483	198	12969.31
74011	388	193	10137.76
74133	307	207	7894.14
74146	116	75	3144.76
74134	104	68	2566.57
74008	50	36	1477.17
74145	29	18	771.69
74104	13	2	358.14
74128	12	4	307.25
74114	9	5	240.88
74136	11	7	233.1
74135	7	7	202.45
74112	8	2	176.84
74129	6	6	131.73
74110	5	2	96.1
74108	4	1	82.1
74137	3	3	78.0
74120	3	2	75.1

Monthly Sales By Zip (Reverse Time Order)				
Year_Month	ZipCode	Number of Order	Number of Customers	Monthly Sales
2003-11	74008	44	34	**1286.08**
2003-11	74011	101	73	**2543.65**
2003-11	74012	302	218	**7162.31**
2003-11	74014	95	66	**2468.76**
2003-11	74104	3	1	**88.15**
2003-11	74108	2	1	**46.95**
2003-11	74110	1	1	**14.50**
2003-11	74112	2	1	**58.45**
2003-11	74128	3	3	**69.90**
2003-11	74129	5	5	**110.89**
2003-11	74133	67	58	**1615.35**
2003-11	74134	23	19	**495.18**
2003-11	74136	1	1	**28.90**
2003-11	74145	3	3	**76.15**
2003-11	74146	26	25	**681.19**
2003-11	74429	1	1	**17.50**
2003-11	Unknown Zip	77	47	**2582.77**
2003-11	**SubTotal of All Zips**	**756**	**557**	**19346.68**
2003-10	74008	3	2	**111.04**

Figure 3.4 Monthly Sales By Zip report. On the left side, the overall sales for each Zip Code is listed. On the right, the total sales for each zip code for each month is listed. Scroll down to see other months and other zips. Each Zip Code is clickable to display details (graphs) for the sales for the selected Zip Code. From the listing, the user can decide which Zip Code to navigate to look for details.



Sales By Zip 74012

Year_Month	Number of Order	Number of Customers	Monthly Sales
2003-11	302	218	**7162.31**
2003-10	275	204	**6297.16**
2003-09	287	214	**6665.28**
2003-08	345	248	**8105.01**
2003-07	268	196	**7653.36**
2003-06	234	185	**7030.05**

Drop in sales as possible consideration for advertising

Steady Sales Due to Direct-Mail from Restaurant

Figure 3.5 Monthly Sales By Zip report - Detailed Report for selected Zip Code. This detailed report is again drillable. When the users clicks on a month, the detailed daily sales number for the Zip Code are graphed at the bottom part of the report. The user can also turn on the graph for the weekly sales for the zip code. Low and high sales for each month, each week, and each day in a month can be reviewed to determine progress and improvements.

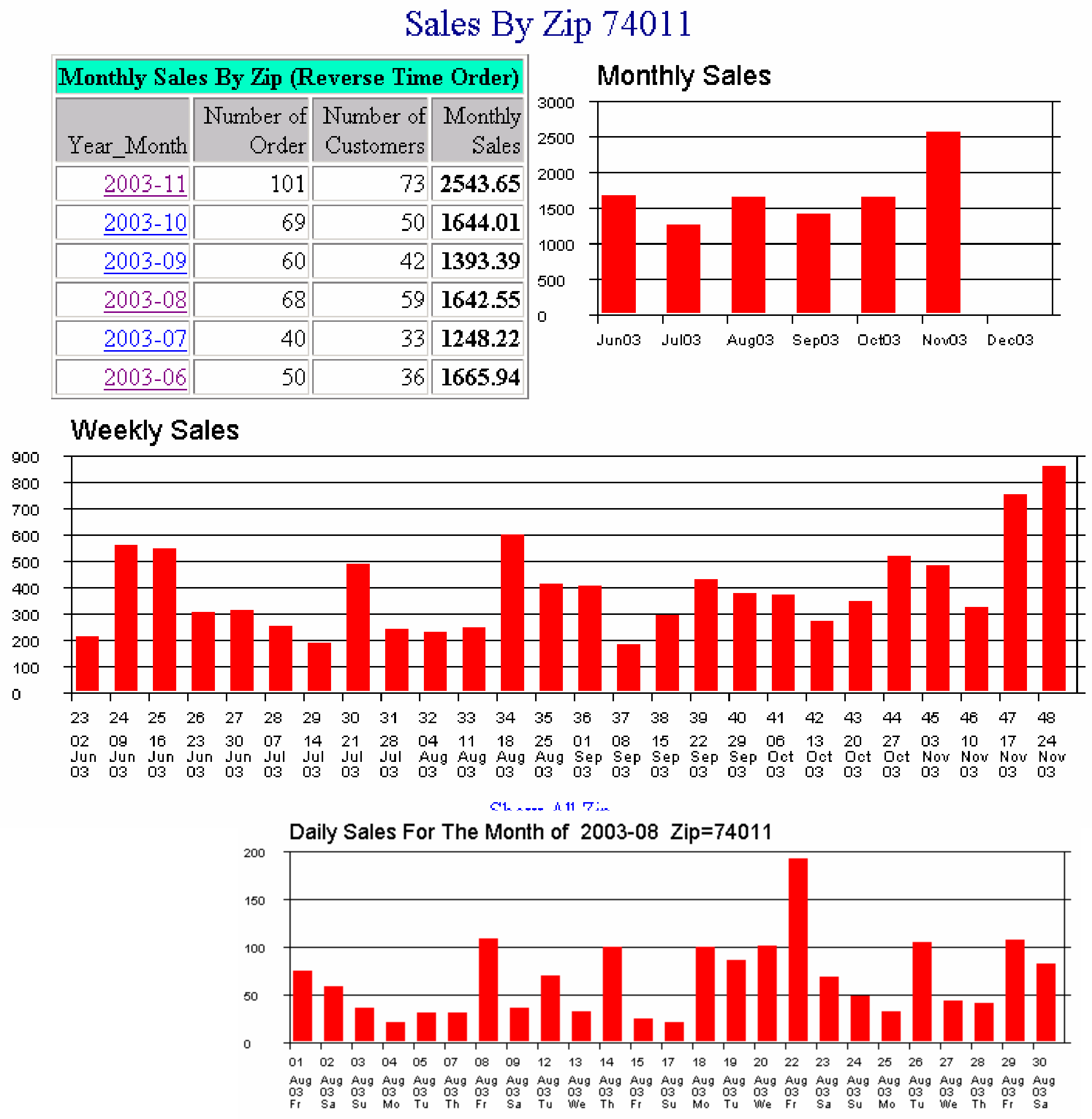
detailed daily sale for this zip code is graphed at the bottom part of the report. The user can also turn on the graph for the weekly sales for the zip code in the same page. Thus the user can find low sales points and the user can investigate the causes and take certain actions such as advertising and improving food and customer service.

From Figure 3.5, we can see the monthly sales for zip code 74012 is fairly stable. Advertising in this zip code is useful, but can be at greater interval.

Figure 3.6 is the monthly sales for zip code 74011. There is significant fluctuations of sales in the weeks shown. The overall sales for this zip code are good. The effects of advertising can be seen as the peak sales do occur after the advertising. After some time, the sales drop. Thus advertising in zip code 74011 is important and the “Return on Investment” (ROI) is good.

In contrast, the sales for zip code 74008 (Figure 3.7) is only showing up in Nov. 2003, significantly higher than any other months. It happened because no advertising was done before November. If more advertising / marketing had been done earlier on zip code 74008, there would have been more sales opportunities in the earlier months.

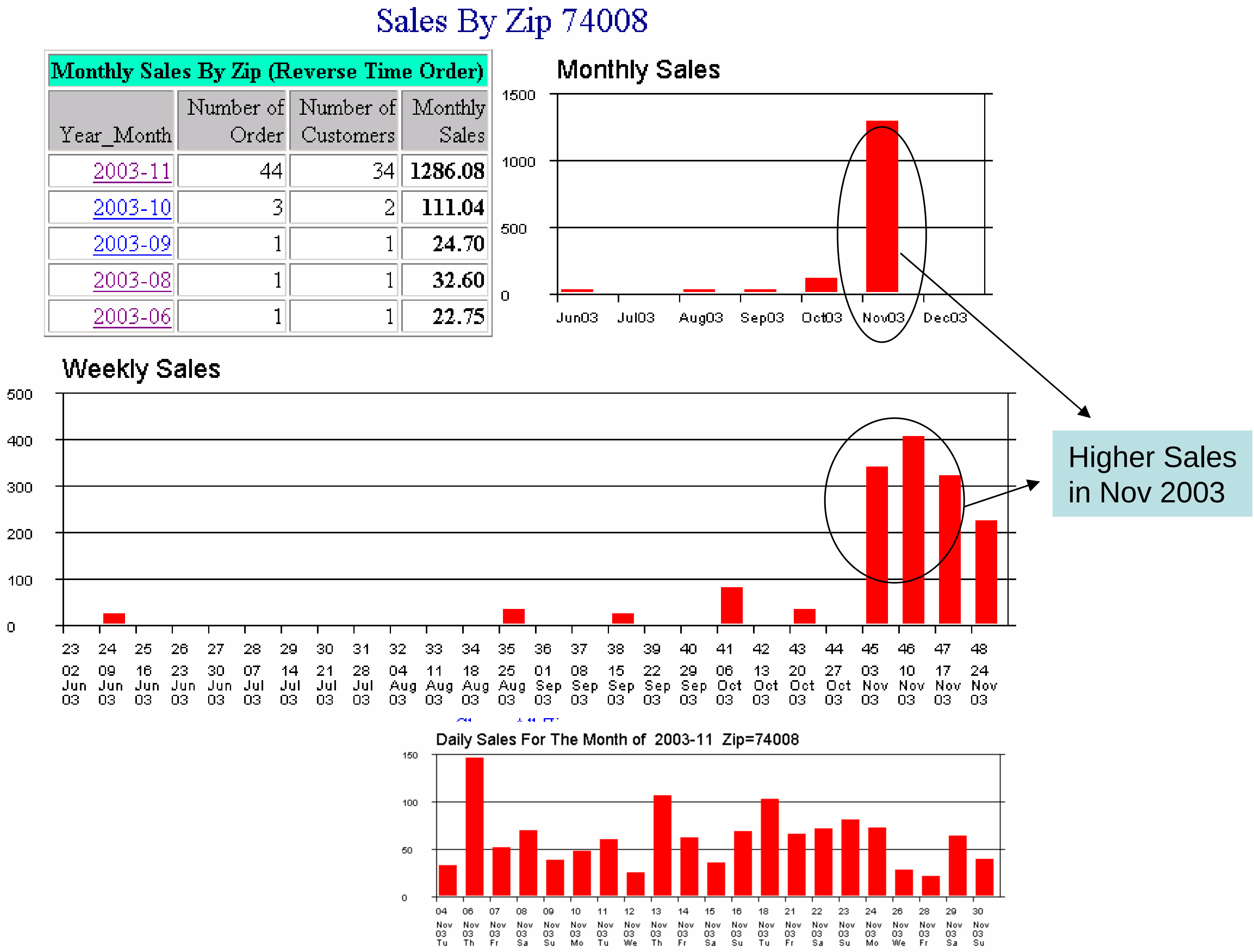
Figure 3.8 Monthly Sales By Zip report - zip code 74145. This figure shows the advertising is not worthwhile. There were two advertising time points, each time costing approximately $400 for advertising. But the return in sales is less than the advertising dollars spent. Therefore, such zip codes should be dropped from future advertising.



Sales By Zip 74011

Monthly Sales By Zip (Reverse Time Order)			
Year_Month	Number of Order	Number of Customers	Monthly Sales
2003-11	101	73	**2543.65**
2003-10	69	50	**1644.01**
2003-09	60	42	**1393.39**
2003-08	68	59	**1642.55**
2003-07	40	33	**1248.22**
2003-06	50	36	**1665.94**

Figure 3.6 Monthly Sales By Zip report - Detailed Report for selected Zip Code. This detailed report is again drillable. When the users clicks on a month, the detailed daily sales number for the Zip Code are graphed at the bottom part of the report. The user can also turn on the graph for the weekly sales for the zip code. Low and high sales for each month, each week, and each day in a month can be reviewed to determine progress and improvements.

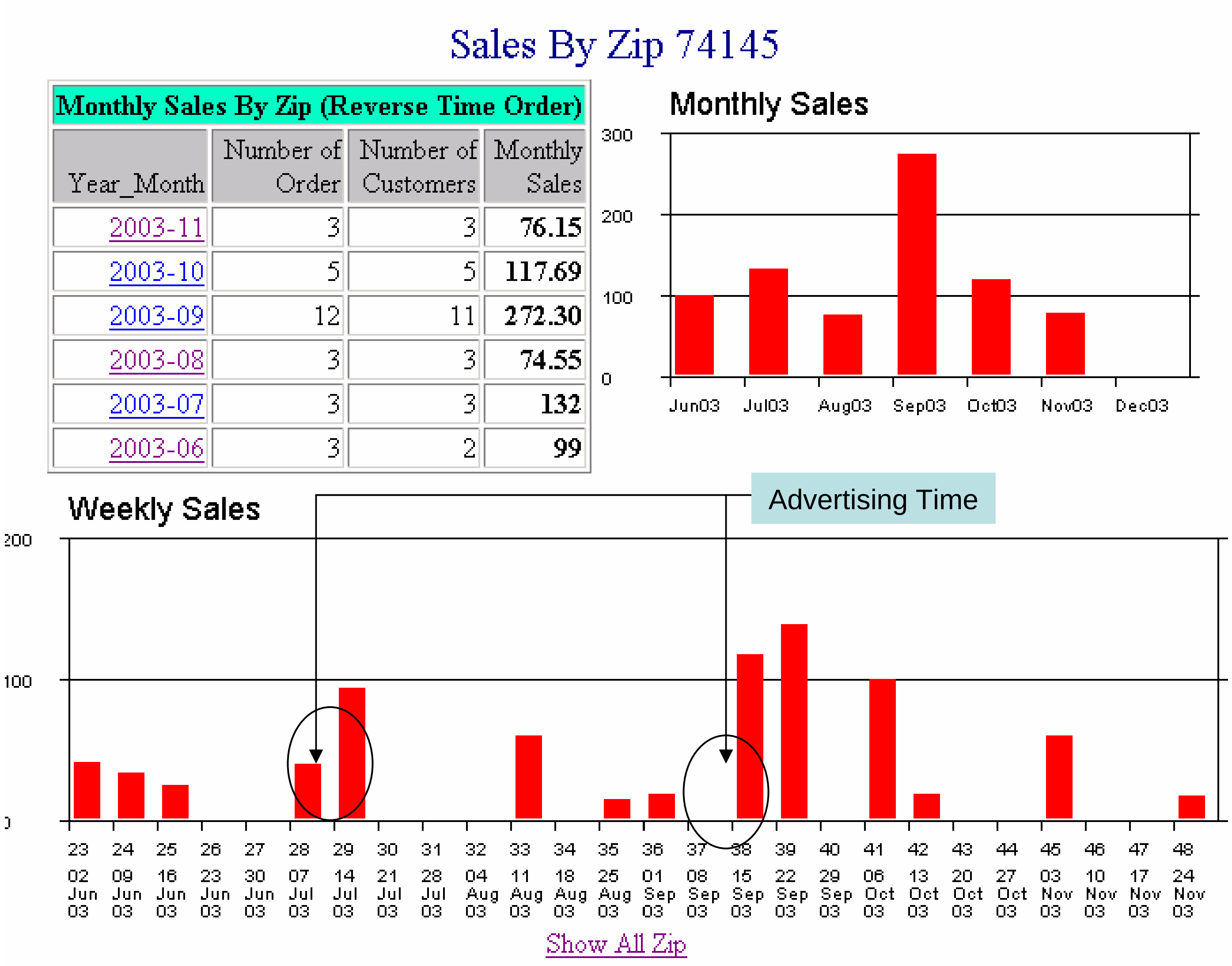


Sales By Zip 74008

Monthly Sales By Zip (Reverse Time Order)			
Year_Month	Number of Order	Number of Customers	Monthly Sales
2003-11	44	34	**1286.08**
2003-10	3	2	**111.04**
2003-09	1	1	**24.70**
2003-08	1	1	**32.60**
2003-06	1	1	**22.75**

Higher Sales in Nov 2003

Figure 3.7 Monthly Sales By Zip report - Zip Code 74008. The weekly graph (middle) shows the sales had been mostly in November. There have been missed sales in the earlier months. If advertising had been done in the earlier months in this zip code, much more sales would have been made from zip code.



Sales By Zip 74145

Monthly Sales By Zip (Reverse Time Order)

Year_Month	Number of Order	Number of Customers	Monthly Sales
2003-11	3	3	**76.15**
2003-10	5	5	**117.69**
2003-09	12	11	**272.30**
2003-08	3	3	**74.55**
2003-07	3	3	**132**
2003-06	3	2	**99**

Show All Zip

Figure 3.8 Monthly Sales By Zip report - Zip Code 74145. This figure shows the advertising is not worthwhile. There were two advertising time points, each time costing approximately $400 for advertising. But the return in sales is less than the advertising dollars spent. Therefore, such zip codes should be dropped from future advertising.

From the “Monthly Sales by Zip” report, the user can find the sales patterns for any given zip code and to find which dates and weeks are better in sales than other time period, or whether advertising is not effective, therefore this report helps to determine which zip code needs more advertising and marketing work, which zip code needs to be dropped from future advertising.

Customer Frequency

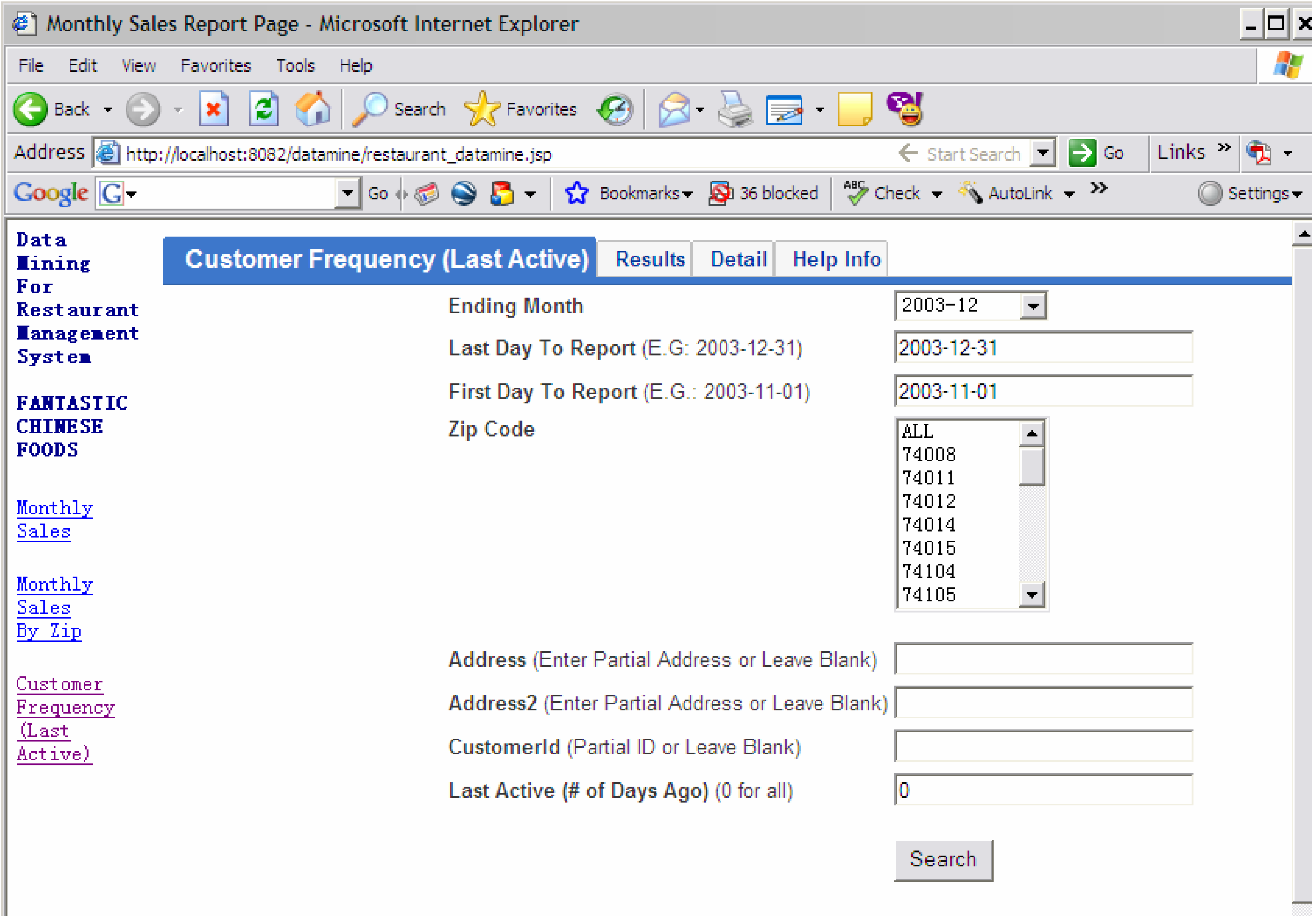
The “Customer Frequency” report (Figure 3.9) has the URL:

http://localhost:8082/datamine/monthlySalesTopCustomer.jsp

The report is loaded from the main page into the <IFrame> through JavaScript:

```
     pageurl =   “http://localhost:8082/datamine/monthlySalesTopCustomer.jsp”;
     document.all.spanPageDetail.innerHTML = "<IFRAME width=870 height=650 frameborder=0
name=ifr src=\"" + pageurl + "\"></iframe>";
```

Figure 3.9 shows the “Customer Frequency (Last Active) Report” search screen. Figure 3.9 Customer Frequency (Last Active) Report. The default display is the "Search Tab" for the search parameters. The user can enter various parameter to search for last active customer. The “Ending Month” specifies that last day of the last month to be considered for sales data. The “Last Day To Report” and “First Day To Report” specify the time period to be displayed. Zip code and address can be searched. The ”Last Active” parameter specified how many days the customer(s) being “Inactive”. If this parameter is 0, then this field is not used as criteria. In the “Search Tab”, user can enter various


Monthly Sales Report Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8082/datamine/restaurant_datamine.jsp
Start Search Go Links
Google Go Bookmarks 36 blocked Check AutoLink Settings
Data Mining For Restaurant Management System
FANTASTIC CHINESE FOODS
Monthly Sales
Monthly Sales By Zip
Customer Frequency (Last Active)
Customer Frequency (Last Active)
Results
Detail
Help Info
Search

Ending Month	2003-12
Last Day To Report (E.G: 2003-12-31)	2003-12-31
First Day To Report (E.G.: 2003-11-01)	2003-11-01
Zip Code	ALL 74008 74011 74012 74014 74015 74104 74105
Address (Enter Partial Address or Leave Blank)	
Address2 (Enter Partial Address or Leave Blank)	
CustomerId (Partial ID or Leave Blank)	
Last Active (# of Days Ago) (0 for all)	0

Figure 3.9 Customer Frequency (Last Active) Report. The default display is the "Search Tab" for the search parameters. The user can enter various parameter to search for last active customer. The “Ending Month” specifies that last day of the last month to be considered for sales data. The “Last Day To Report” and “First Day To Report” specify the time period to be displayed. Zip Code and Address can be searched. The ”Last Active” parameter specified how many days the customer(s) being “Inactive”. If this parameter is 0, then this field is not used as criteria.

parameters to search for the last time the customer was active. User can enter any combination of the search. For example, if we want to search for customers who have not ordered in the past 21 days, then we enter “21” in the “Last Active” field. This way, we find customers who have not orders in the recent past and research on the customers, and find possible causes, and decide whether to mail individual menus, coupons, and other offers.

The results of the search (Figure 3.10) are displayed as a graph for customer order history for each customer id. Each customer id occupies one horizontal line in the graph. Each dot represents an order from a customer. The size of the dot represents the dollar amount of the order -- A larger dot means greater dollar amount. More dots mean more orders. For example, customer “angieri4704” orders quite frequently every few days; Customer “birch5604s” has large orders. Customer “birch5604s” has been inactive after Nov. 26, 2003. Customer “broadw3320” has been inactive after Nov. 02, 2003.

From the graph in Figure 3.10, the owner can determine which customers to review, especially those customers who have not orders in a month or more.

In the “Details” tab, the owner can do further search by selecting the customers and then clicking on “Get Detail Chart” and “Get Detail Listing” (Figure 3.11) to see the detailed graphs and listings for the selected customers.

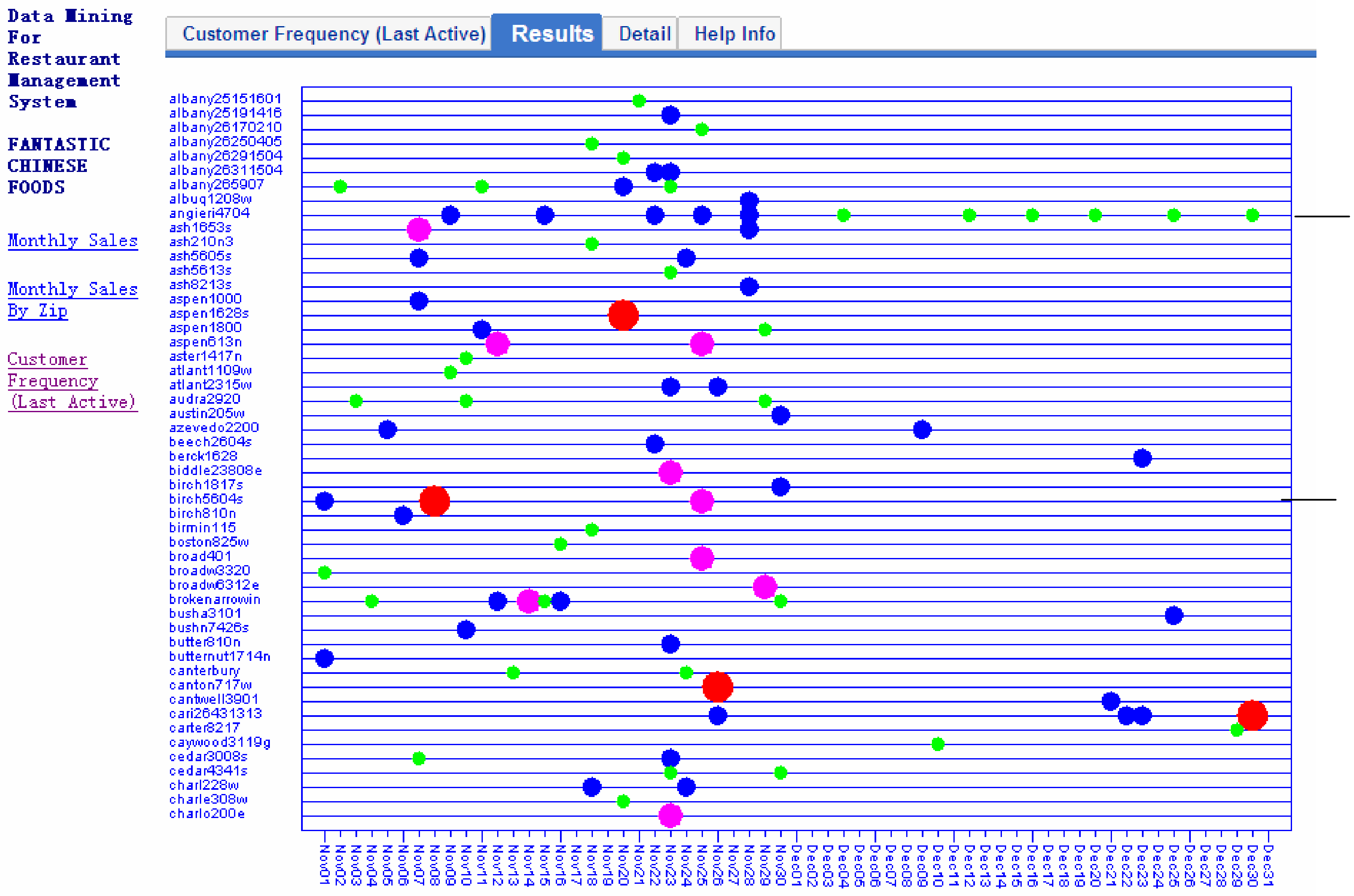
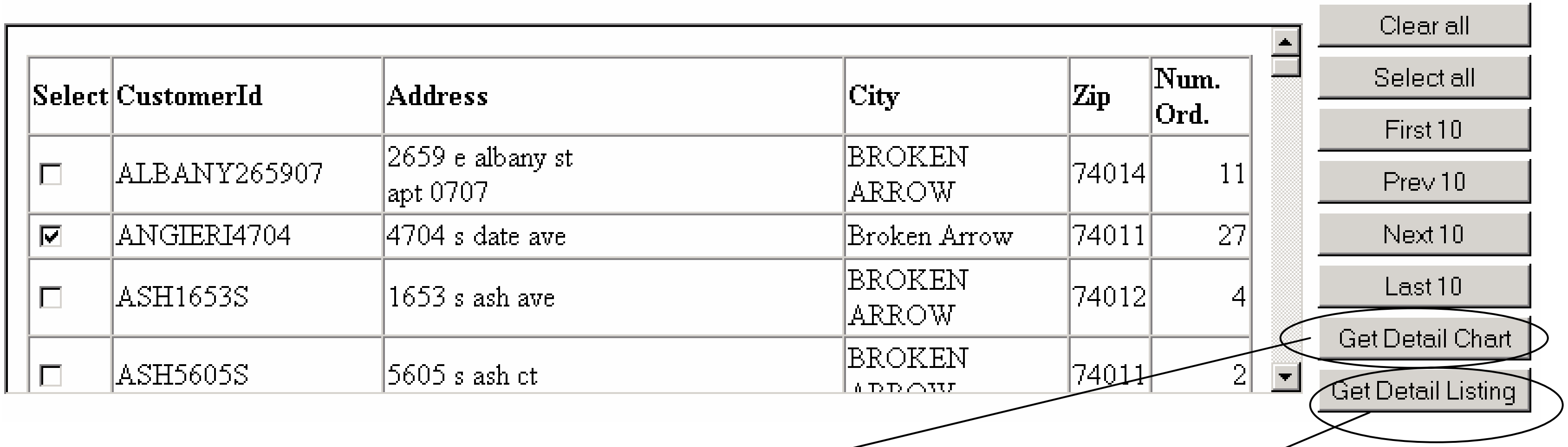


Figure 3.10 Customer Frequency (Last Active). The results of the search is displayed as a graph for user order patterns for each user id. Each user's order occupy one horizontal line in the graph. Each dot represents an order from a customer. The side of the dot represent the dollar amount of the order -- A larger dot means more dollar amount.



Select	CustomerId	Address	City	Zip	Num. Ord.
□	ALBANY265907	2659 e albany st apt 0707	BROKEN ARROW	74014	11
☑	ANGIERI4704	4704 s date ave	Broken Arrow	74011	27
□	ASH1653S	1653 s ash ave	BROKEN ARROW	74012	4
□	ASH5605S	5605 s ash ct	BROKEN	74011	2

Click on the button to see further detailed graph results on Figure 3.12

Click on the button to see further detailed listing results on Figure 3.13

Figure 3.11 Detail Tab to select individual customers for further detail graphs and listings for analysis. Click on the buttons to see graph and listing details on Figure 3.12 and Figure 3.13

Figure 3.12 shows the results of “Detail Chart” of customers being selected. Figure 3.12A shows a bar-chart for a customer’s order history with all dates displayed evenly along the horizontal axis. From this chart we can see clearly how much the customer spent for each order. Figures 3.12B shows a pie-chart for the customer’s top 10 most-ordered food items by total dollar. From this we can analyze the favorite and/or frequently ordered food items. Figure 3.12C shows the frequency diagram for the customer for the entire order history of interest. From this chart we can see the date intervals for the customer’s orders and decide whether we need to offer something special from time to time.

Figure 3.13 shows the results of “Detail Listing”. Figure 3.13 Detailed customer listing analysis. Key indicators for all selected customers are listed here, including address type, zip code, count of totals, number of delay counts, and number of days customer has not been active. Restaurant management can further get details of each customer by clicking on the CustomerId link to get a detailed Customer report, thus determine whether to mail menu and make special offers to each individual customer.

Conclusion for Data Mining and Web Reporting

It is shown that we can use Java (JSP) to do data mining for restaurant management. The JSP generates HTML pages for web reporting (listings and graphs), which is useful to show sales and order patterns.

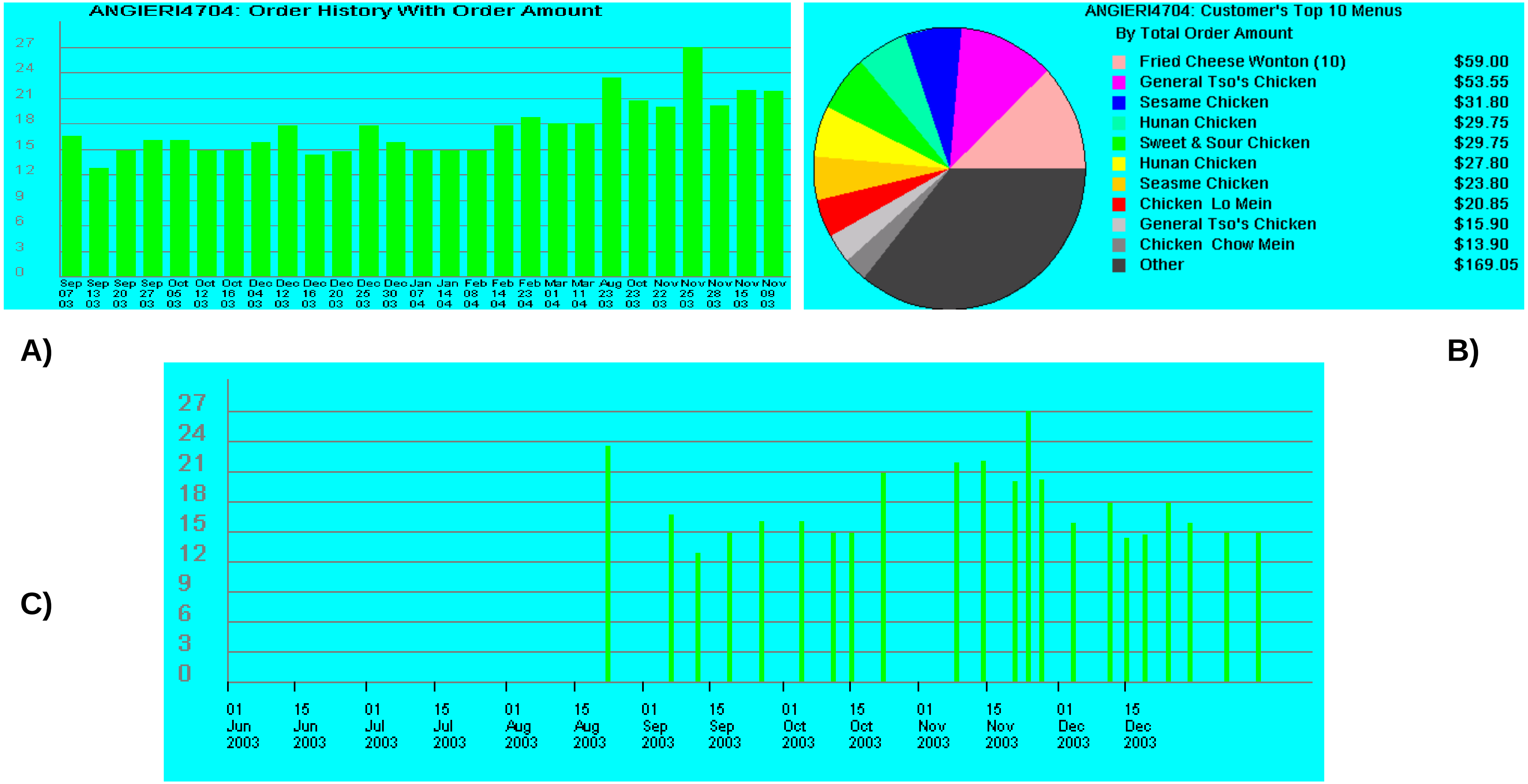
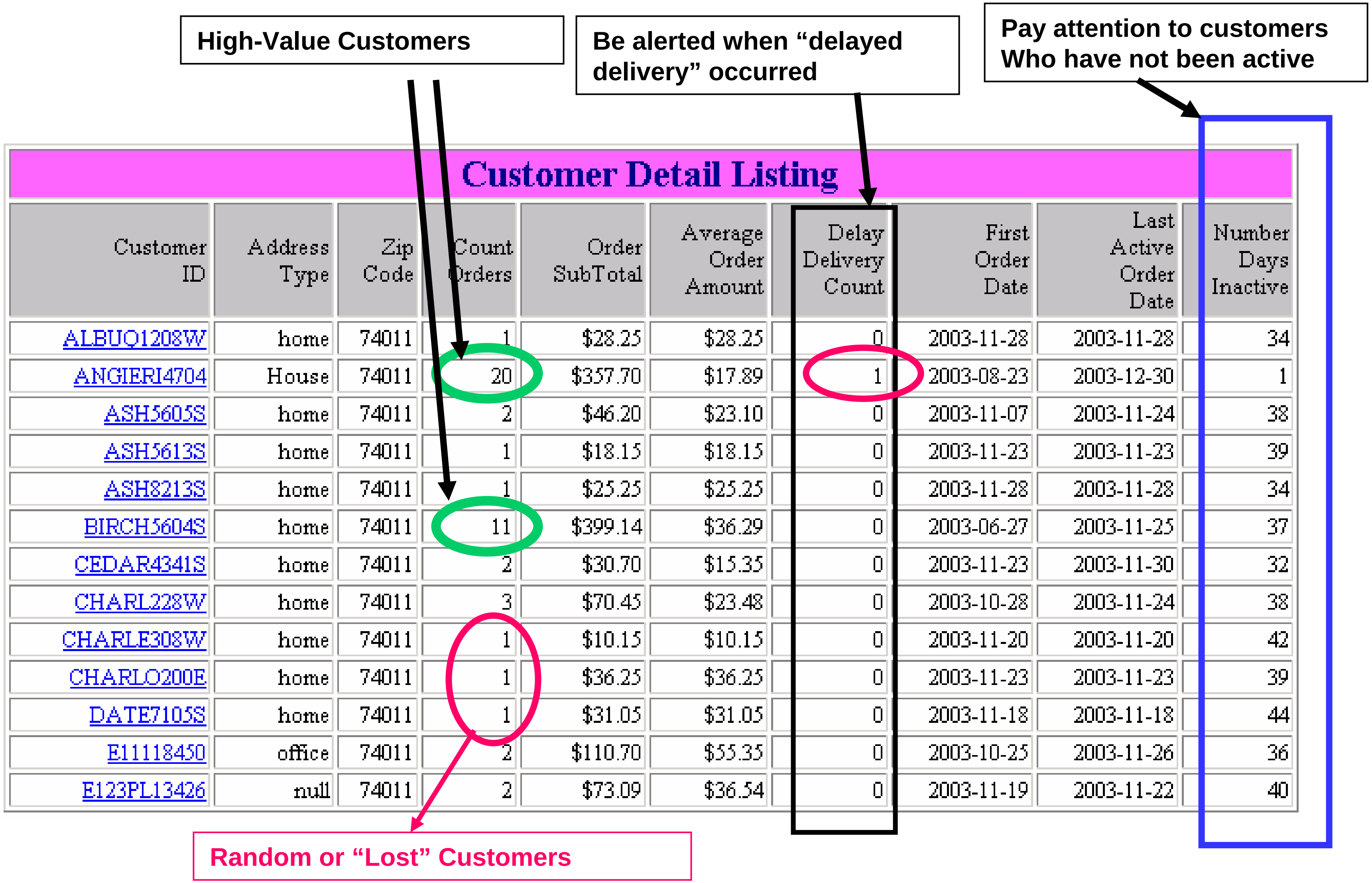


Figure 3.12 Customer detail Graph - Figure 3.12A - Detail Bar-Chart for Selected Customer. Each order is displayed with order date spaced evenly and labeled clearly at the bottom of the horizontal axis. Figure 3.12B – Pie Chart showing top 10 most order food items. Customer's most frequently ordered food items are represented by the size of the slices in the pie chart. Figure 3.12C is the frequency bar chart showing amount of order and true gaps for the orders. This is similar to the dots in Figure 3.10, yet this also shows the order amount in true scale. From this graph, it can be easily seen how often the customer orders and how long has the customer been active/inactive.



High-Value Customers

Be alerted when “delayed delivery” occurred

Pay attention to customers Who have not been active

Random or “Lost” Customers

Customer Detail Listing

Customer ID	Address Type	Zip Code	Count Orders	Order SubTotal	Average Order Amount	Delay Delivery Count	First Order Date	Last Active Order Date	Number Days Inactive
ALBUQ1208W	home	74011	1	$28.25	$28.25	0	2003-11-28	2003-11-28	34
ANGIERI4704	House	74011	20	$357.70	$17.89	1	2003-08-23	2003-12-30	1
ASH5605S	home	74011	2	$46.20	$23.10	0	2003-11-07	2003-11-24	38
ASH5613S	home	74011	1	$18.15	$18.15	0	2003-11-23	2003-11-23	39
ASH8213S	home	74011	1	$25.25	$25.25	0	2003-11-28	2003-11-28	34
BIRCH5604S	home	74011	11	$399.14	$36.29	0	2003-06-27	2003-11-25	37
CEDAR4341S	home	74011	2	$30.70	$15.35	0	2003-11-23	2003-11-30	32
CHARL228W	home	74011	3	$70.45	$23.48	0	2003-10-28	2003-11-24	38
CHARLE308W	home	74011	1	$10.15	$10.15	0	2003-11-20	2003-11-20	42
CHARLO200E	home	74011	1	$36.25	$36.25	0	2003-11-23	2003-11-23	39
DATE7105S	home	74011	1	$31.05	$31.05	0	2003-11-18	2003-11-18	44
E11118450	office	74011	2	$110.70	$55.35	0	2003-10-25	2003-11-26	36
E123PL13426	null	74011	2	$73.09	$36.54	0	2003-11-19	2003-11-22	40

Figure 3.13 Detailed customer listing analysis. Key indicators for all selected customers are listed here, including address type, zip code, count of totals, number of delay counts, and number of days customer has not been active. Restaurant management can further get details of each customer by clicking on the CustomerId link to get a detailed Customer report, thus determine whether to mail menu and offer special to each individual customer.

Chapter 4 Conclusions

Data mining has progressed significantly in the past decade. Most large companies have used data mining in various aspects to improve their products and their services. However, for small businesses, data mining is less researched and developed. This thesis demonstrates that small businesses can accomplish similar goals to those of large companies if the small businesses make the effort.

There are many methods and programming techniques for data mining. The thesis is most relevant to the visualization of data mining. The thesis presents examples for building an information system for restaurant using Java programming language and Java Server Page Technology. Drillable HTML reports with tabular listing and graphs are implemented using JSP pages. Restaurant management can research and explore the information in the HTML reports to produce more sales and create more satisfied customers.

First, Oracle 9I database server software is installed and a database for data mining is created to store the data for the restaurant management system. Oracle database software is reliable and efficient to store business data and the database can be accessed through Java Programming and Java Database Connectivity (JDBC).

Second, Netbeans IDE has been used as a GUI tool to develop a Java application for the data entry into the data mining database. The developer uses a drag and drop method to create the User Interface that includes Labels, Textboxes, Drop-down boxes, Selection Boxes, JTables, and JFrames. The developer adds event-handling code and database access code to make the Java application to respond to user inputs. It has been demonstrated that it is easy to build GUI-based application using tools such as Netbeans IDE. In addition to data entry, data mining features such as search, order history, and additional customer-related information is included in the Java application. By using the additional information, the restaurant management can prepare better and fresher food, make better recommendation for choices of food items, and deliver to customers in timely manner.

Third, a Web-based application is built using Apache Tomcat and Java Server Pages (JSP). The coding of JSP primarily is done in a text editor and there is not much help using a GUI-based IDE. Thus understanding HTML and JSP syntax, and embedding Java code to produce HTML pages are essential. The web-based application does not require distributing a lot of software to the end user’s computer.

Fourth, in the Web-based application, HTML reports with tabular listings and Java 2D graphs were created in the JSP pages. Various reports are created to visualize and mine the data for the restaurant management system. A monthly sales report is listed in tabular form as well as in bar charts. The user can drill into the detail for any given month, thereby seeing the high and low points – the months, the weeks, and the days of week -- of the sales and try to determine what is affecting the sales. In the monthly-sales by zip code report, the monthly sales by all zip codes are initially displayed. When the user clicks on the link of a zip code, a detailed report for the zip code is displayed, allowing the user to see the monthly, weekly and daily sales for that zip code. When the sales in a certain zip code is dropped significantly, it is time to analyze the causes of the slow down and to consider whether to do advertising by an advertising company or by using direct mail from the restaurant. From the data mining results we can determine which zip code area we need to explore to develop more business by expanding to new neighboring zip codes advertising and/or which zip area we need to drop the advertising depending on the low response rate in the past; thus, we can save advertising expenses effectively. In a customer frequency report, the user can search customers based on various criteria such as the report date range, zip code, partial address and the number of days that customers have not been ordering (i.e., are not active). If the user searches for customers whose last activity was 20 days ago, then a graph of all customers who have been inactive for 20 days and more will be displayed. In the graph, the dots represent the amount of each order and the spacing of the dots represents the interval for the orders. Further information about each customer is available in the details tab, allowing the user to see the customer’s order history – dates and amount -- in a bar chart, the customer’s top 10 favorite food items in a pie chart, and the customer’s order history with date spacing to see customer’s historical order frequency. A detailed listing of customer’s order history, customer preferences information, historical customer service and delivery notes are also displayed. From the customer frequency and detailed customer history, the restaurant management can decide which customers should receive new menus; which customers may have lost their menus and should receive a menu from time to time; which customers should receive special offers, especially in seasons with slow sales; and which customers can provide feed back to improve the overall restaurant customer service.

The focus of this thesis is to demonstrate how to use Java as a data mining tool to implement a visualization and data mining system. All of the code of the thesis has been developed by the author. This is a good example to demonstrate that Java programming is useful in solving practical business problems.

The thesis demonstrates data mining for small businesses using Java Programming and Java Server Pages is relatively cost-effective due to license free server software such as Apache Tomcat and the efforts of developing the data mining application.

The thesis should contribute to the understanding and application of the Java Programming and Java Server Techniques to visualization and data mining for small businesses.

The thesis is not so much about any special algorithm as it is about using Java programming in the real world of basic statistics and visualization to the data mining for small businesses. It will be interesting to do a comparison of the current method of data mining using the Java Language and Java Server Technologies with the algorithms from data mining tools such as the Weka software package.

Further study of the data mining in a restaurant management system can be done by adding more drill-down reports and other statistical results. For example, the map display and animation of active customers within a certain time period and certain zip codes can be useful to see the active and inactive customer areas. We can study the overall preferences of menu items by time and by zip codes. We can study the average turnover (churn) of customers being retained or lost. We can implement the map display and animation through Java programming without much difficulty. What will be the equivalent and/or even more advanced ways to study such topics through data mining algorithms such as clustering analysis? What other data mining algorithms can be implemented and applied to the real business situation. It will be an interesting study topic about the application of data mining algorithms in the future.

References

1. Berry, Michael J. A. and Linoff, Gordon S. Mastering Data Mining. New York, NY: John Wiley & Sons, Inc., 2000.

2. Buono, Paolo; Costabile, Maria F.; Jaeschke, Gerald and Hemmje, Matthias. Analysing data through visualizations in a web-based trade fair system. Proceedings of the 14th international conference on Software engineering and knowledge engineering(SEKE '02). ACM. July 2002.

3. Hao, M.C.; Dayal, U.; Hsu, M.; Baker, J. and D'Eletto, R. A Java-based visual mining infrastructure and applications. Proceedings 1999 IEEE Symposium on Information Visualization(InfoVis'99). San Francisco, CA, USA. Oct. 24-29 1999 Page(s):124 - 127, 153.

4. Dasu, Tamraparni and Johnson, Theodore. Exploratory Data Mining and Data Cleaning. Hoboken, New Jersey: John Wiley & Sons, Inc., 2002.

5. Data Mining Web Casts, in www.dmreview.com, 2005 (last accessed).

6. Dunham, Margaret H. Data Mining Introductory and Advanced Topics. Upper Saddle River, New Jersey: Pearson Education Inc., 2003.

7. Han, Jiawei and Kamber, Micheline. Data Mining Concepts and Techniques. San Diego, CA: Academic Press: Morgan Kaufman. 2001.

8. Information Builders. Business Intelligence in Action -- live WebFOCUS demo, in www.informationbuilders.com, 2005 (last accessed).

9. Thearling, Kurt. An Introduction to Data Mining – Discovering hidden value in your data warehouse. White Papers, www.thearling.com, 2008 (last accessed).

10. Wang, John. Data Mining Opportunities and Challenges. Hershey, PA: Idea Group Inc., 2003.

11. Witten, Ian H. and Frank, Eibe. Data mining: practical machine learning tools and techniques with Java implementations. San Francisco, CA: Morgan Kaufmann Publishers Inc., 2000.

12. Zdravko, Markov, Ingrid and Russell. An introduction to the WEKA data mining system. ACM SIGCSE Bulletin. Volume 38(3). 2006.

13. www.amazon.com. Personal experience of buying books on the web site.

14. www.visualmining.com, “Dashboard Software”, 2008 (last accessed).

15. “Oracle/Java FAQ”, in http://www.orafaq.com/faqjdbc.htm, Feb., 2008 (last accessed).

16. “Oracle ODBC Connectivity FAQ”, in http://www.orafaq.com/faqodbc.htm, Feb., 2008 (last accessed).

17. http://www.oracle.com/technology/software/index.html

18. http://www.quest.com

19. http://java.sun.com/j2se/1.4.2/download-netbeans.html

20. http://tomcat.apache.org/, Apache Tomcat.

21. http://www.yorku.ca/jhuang/examples/tomcat-install.html

22. http://java.sun.com/docs/books/tutorial/2d/TOC.html

Appendix A: Configuration of JDK and TOMCAT

Java Devlopment Kit (JDK) and Java Runtime (JRE)

Any Java Application and Java Servlet Programs need to be compiled into Java Class Code (.class) before they can be executed. The Java Development Kit (JDK) from Sun Micro Systems is used to compile the Java Programs. The JDK and the Java Runtime (JRE) were downloaded from Sun Microsystems's website:
http://java.sun.com

The JDK and JRE together are also called JSE for "Java Standard Edition".

The specific JDK that I downloaded is "J2SDK1.4.2_16" and it is installed in the "C:\j2sdk1.4.2_16" directory. The JRE downloaded is "jre1.4.2_16" and it is installed in "C:\program files\Java\jre1.4.2_06". The JDK and JRE versions are not critical, as long as they are j2sdk1.4.x or later, in order for the Java components to work together with the Apache Tomcat Server.

Apache Tomcat Server

The Apache Tomcat Server was downloaded from the Apache Web Site:
http://tomcat.apache.org

The version of the Tomcat I used is 5.0.28. Save the file "Jakarta-tomcat-5.0.28.exe" into a temporary directory and then run the program. Follow the step-by-step instructions to install the "Apache Tomcat" as a windows service.

Environment Variables for Apache Tomcat Server

The Apache Tomcat Server needs environment variables in order to compile a JSP file into Java Servlet class file and/or to compile the Servlet Java program into a Servlet class file. In "Control Panel / System", click on "Advanced" tab, then click on "Environment" button to Add/Review that the following environment variables are set.

		Comment
JAVA_HOME	c:\j2sdk1.4.2_16	Where JDK intalled
CATALINA_HOME	c:\Tomcat5.05	Apache Tomcat
CLASSPATH	c:\Tomcat5.05\common\lib\ojdbc14.jar;	Oracle 9.2I
	c:\Tomcat5.05\common\lib\ojdbc14_g.jar;	Oracle 9.2I
	c:\Tomcat5.05\common\lib\classes12.jar;	Oracle 9.2I
	c:\Tomcat5.05\common\lib\classes111.jar;	Oracle 9.2I
	c:\j2sdk1.4.2_16\lib\tools.jar;	Java Tools
	c:\j2sdk1.4.2_16\lib\dt.jar;	DATE
	c:\oracle\ora92\jdbc\lib\classes.jar;	JDBC

	c:\j2sdk1.4.2_16\lib\jawt.lib;	Java AWT
	c:\Tomcat5.05\common\lib\servlet.jar;	Servlet
	.;	
	...	
PATH	c:\j2sdk1.4.2_16\bin;	JDK (Javac, Java)
	…	

After the installation of the JDK, JRE and Apache Tomcat, verify that the Tomcat server can be started/stopped from the Control Panel / Administrative Tools / Services / Apache Tomcat. Type http://localhost:8082 into the web browser (internet explorer). If the Apache Tomcat default page is displayed, then the Tomcat Server is installed correctly. In the test URL, the “localhost” can be substituted by the computer_name.domain_name. The listening port is configured as “8082” in the file

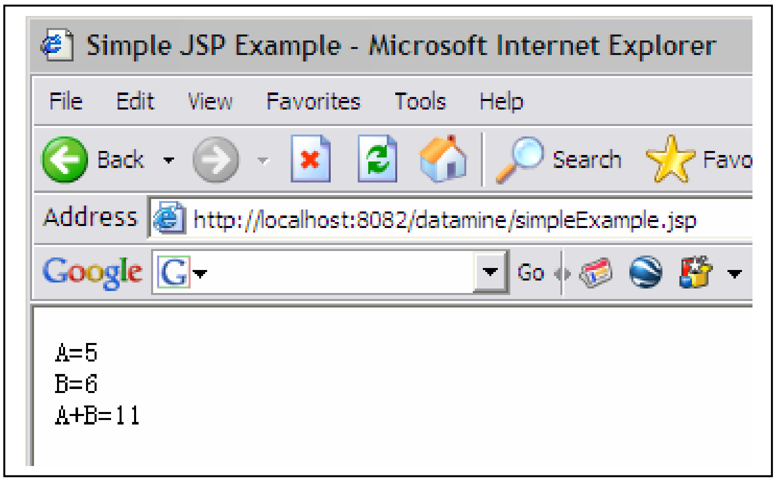
“C:\tomcat 5.05\conf\server.xml”

and look for the port=nnnn (a 4-digit numerical string).

Appendix B: Simple JSP Example

```
<% // APPENDIX B   -- A Simple JSP Example
    //   FILE NAME:    c:\datamine\simpleExample.jsp
    //   URL:              http://localhost:8082/datamine/simpleExample.jsp
   %>

%@ page import="java.io.*" %>
HTML>
 <HEAD>
  <TITLE> Simple JSP Example</TITLE>
/HEAD>
<% // Simple JSP: http://localhost:8082/datamine/simpleExample.jsp
     int A=5;
     int B=6;
     int C=A+B;
     out.println("A=" + A + "<br>");
    out.println("B=" + B + "<br>");
    out.println("A+B=" + C);
%>
```

When the URL:
   http://localhost:8082/datamine/simpleExample.jsp
is typed into the WebBrowser (Internet Explorer), the result is displayed below.



Simple JSP Example - Microsoft Internet Explorer

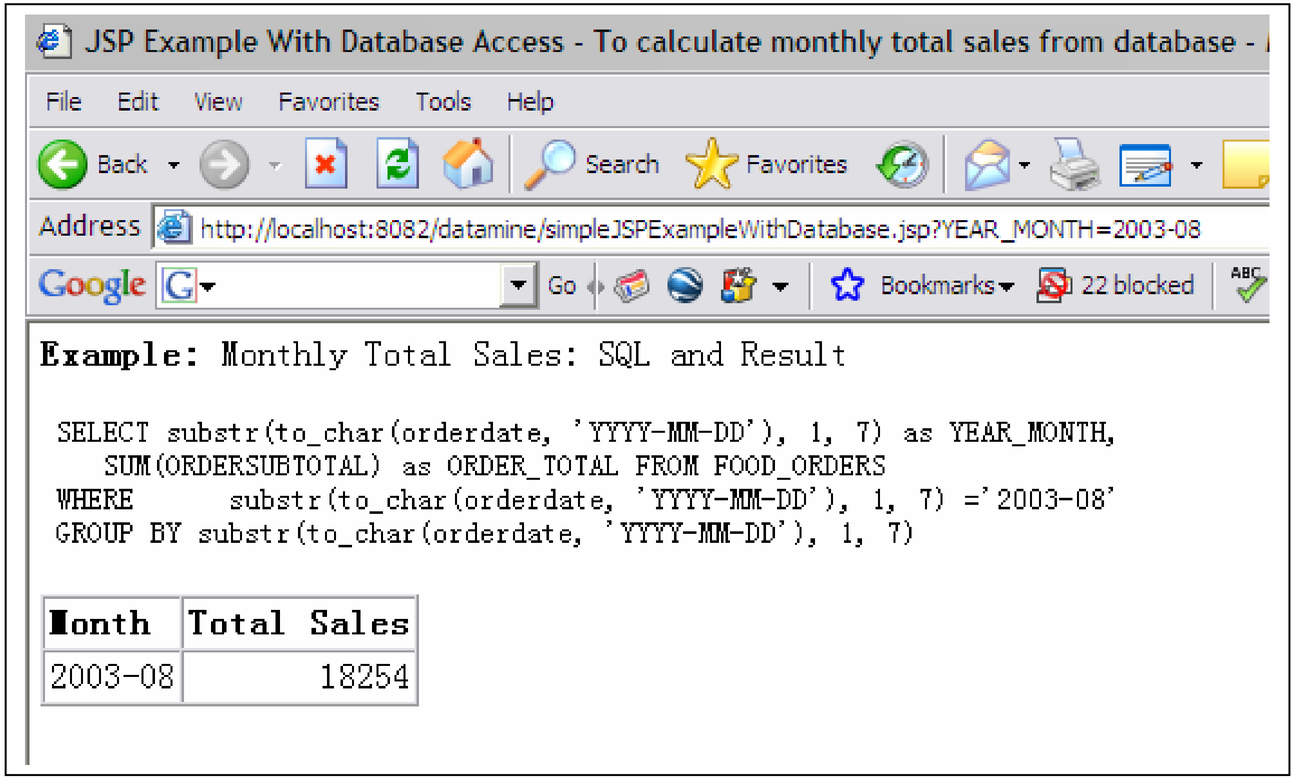
http://localhost:8082/datamine/simpleExample.jsp

```
A=5
B=6
A+B=11
```

Appendix C: Simple JSP Example with Database Access

```
<% //  A JSP Example With Database Access
   // FILE NAME:  c:\datamine\simpleJSPExampleWithDatabase?YEAR_MONTH=2003-08
   // URL:        http://localhost:8082/datamine/simpleJSPExampleWithDatabase.jsp?YEAR_MONTH=2003-08
%>

<%@ page session="true" %>
<%@ page import="java.io.*" %>
<%@ page import="java.sql.*" %>


<html>
<head>
   <title>JSP Example With Database Access - To calculate monthly total sales from database</title>
</head>
<body bgcolor="white" TOPMARGIN=5 LEFTMARGIN=5>
 <%  // http://localhost:8082/datamine/simpleJSPExampleWithDatabase.jsp?YEAR_MONTH=2003-08
     String  YEAR_MONTH = request.getParameter("YEAR_MONTH");

     String strSQL="";
     strSQL = strSQL + " SELECT substr(to_char(orderdate, 'YYYY-MM-DD'), 1, 7) as YEAR_MONTH, \n";
     strSQL = strSQL + "    SUM(ORDERSUBTOTAL) as ORDER_TOTAL FROM FOOD_ORDERS \n";
     strSQL = strSQL + " WHERE ";
     strSQL = strSQL + "     substr(to_char(orderdate, 'YYYY-MM-DD'), 1, 7) ='" + YEAR_MONTH + "' \n";
     strSQL = strSQL + " GROUP BY substr(to_char(orderdate, 'YYYY-MM-DD'), 1, 7) \n";

     out.println("<b>Example: </b> Monthly Total Sales: SQL and Result<br>");
     out.println("<pre>" + strSQL + "</pre>");
     try
      {
         Class.forName("oracle.jdbc.driver.OracleDriver");
         Connection conn =
DriverManager.getConnection("jdbc:oracle:thin:datamine/datamine987@localhost:1521:DATAEXP");
```

```
         // Execute SQL, get recordset
         Statement stmtMonthly = conn.createStatement();
         ResultSet RS_monthly;
         RS_monthly = stmtMonthly.executeQuery(strSQL);

         out.println("<TABLE border=1 cellspacing='0' cellpadding='2'>");
         out.println("   <TR>");
         out.println("        <TD><b>Month</b></TD>");
         out.println("        <TD><b>Total Sales</b></TD>");
         out.println("   </TR>");

         // Loop Through Recordset and store data in arrays
         while(RS_monthly.next())
         {
              out.println("<TR>");
              out.println("    <TD>" + RS_monthly.getString("YEAR_MONTH") + "</TD>");
              out.println("    <TD align=\"right\">" + RS_monthly.getInt("ORDER_TOTAL") + "</TD>");
              out.println("</TR>");
         };
         out.println("</TABLE>");
         RS_monthly.close();
       } catch(ClassNotFoundException e){
         response.setContentType("text/html");
         out.println("Error: "+ e);
      } catch(SQLException s){
         response.setContentType("text/html");
         out.println("Error: "+s.getSQLState()+","+s.getMessage()+" ("+s+")");
      } catch(Exception s1){
         response.setContentType("text/html");
         out.println("Error: " +s1.getMessage());
      };
%>

</body>
</html>
```


JSP Example With Database Access - To calculate monthly total sales from database -
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://localhost:8082/datamine/simpleJSPExampleWithDatabase.jsp?YEAR_MONTH=2003-08
Go
Bookmarks
22 blocked

Example: Monthly Total Sales: SQL and Result

```
SELECT substr(to_char(orderdate, 'YYYY-MM-DD'), 1, 7) as YEAR_MONTH,
   SUM(ORDERSUBTOTAL) as ORDER_TOTAL FROM FOOD_ORDERS
WHERE      substr(to_char(orderdate, 'YYYY-MM-DD'), 1, 7) ='2003-08'
GROUP BY substr(to_char(orderdate, 'YYYY-MM-DD'), 1, 7)
```

Month	Total Sales
2003-08	18254

Appendix D: Simple Servlet Example

```
/*
FILE LOCATION:
   C:\datamine\WEB-INF\classes\SimpleServletExample.java
URL:
http://localhost:8082/datamine/SimpleServletExample.java
*/


import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

// Simple Servlet Example
http://localhost:8082/datamine/servlet/SimpleServletExample?param1=aaa&param2=bbb

public class SimpleServletExample extends HttpServlet {


   public void doGet(HttpServletRequest request,
                 HttpServletResponse response)
      throws IOException, ServletException
   {
       response.setContentType("text/html");
      PrintWriter out = response.getWriter();

      out.println("<html>");
      out.println("<head>");
      out.println("<content=\"no-cache\" http-equiv=\"pragma\">");

      out.println("</head>");

      out.println("This is a simple servlet<br>");
      String valuePassed = request.getParameter("param1");

      String valuePassed2= request.getParameter("param2");

      out.println("parameter1=" + valuePassed + "<br>");
      out.println("parameter2=" + valuePassed2 + "<br>");

      out.println("</body>");
      out.println("</html>");
   };
   public void doPost(HttpServletRequest request,
                  HttpServletResponse response)
     throws ServletException, IOException {
     doGet(request, response);
   };

}
```

Appendix E: Simple Servlet Example With Database Access

```
/*
An example Servlet page with database access to the Oracle 9I database DATAEXP:
File Location:
C:\datamine\WEB-INF\classes\SimpleServletExampleWithDatabase.java
URL:
http://localhost:8082/datamine/servlet/SimpleServletExampleWithDatabase
*/

import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

import java.sql.*;


// Simple Servlet Example    http://localhost:8082/datamine/servlet/SimpleServletExampleWithDatabase?YEAR_MONTH=2003-08

public class SimpleServletExampleWithDatabase extends HttpServlet {


    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws IOException, ServletException
    {
         response.setContentType("text/html");
        PrintWriter out = response.getWriter();
```

```
// http://localhost:8082/datamine/servlet/SimpleServletExampleWithDatabase?YEAR_MONTH=2003-08
   String  YEAR_MONTH = request.getParameter("YEAR_MONTH");

   String strSQL="";
   strSQL = strSQL + " SELECT substr(to_char(orderdate, 'YYYY-MM-DD'), 1, 7) as YEAR_MONTH, \n";
   strSQL = strSQL + "    SUM(ORDERSUBTOTAL) as ORDER_TOTAL FROM FOOD_ORDERS \n";
   strSQL = strSQL + " WHERE ";
   strSQL = strSQL + "     substr(to_char(orderdate, 'YYYY-MM-DD'), 1, 7) ='" + YEAR_MONTH + "' \n";
   strSQL = strSQL + " GROUP BY substr(to_char(orderdate, 'YYYY-MM-DD'), 1, 7) \n";

   out.println("<b>Example: </b> Monthly Total Sales: SQL and Result<br>");
   out.println("<pre>" + strSQL + "</pre>");
   try
    {
      Class.forName("oracle.jdbc.driver.OracleDriver");
      Connection conn = DriverManager.getConnection("jdbc:oracle:thin:datamine/datamine987@localhost:1521:DATAEXP");

      // Execute SQL, get recordset
      Statement stmtMonthly = conn.createStatement();
      ResultSet RS_monthly;
      RS_monthly = stmtMonthly.executeQuery(strSQL);

      out.println("<TABLE border=1 cellspacing='0' cellpadding='2'>");
      out.println("   <TR>");
      out.println("       <TD><b>Month</b></TD>");
      out.println("       <TD><b>Total Sales</b></TD>");
      out.println("   </TR>");

      // Loop Through Recordset and store data in arrays
      while(RS_monthly.next())
```

```
     {
        out.println("<TR>");
        out.println("    <TD>" + RS_monthly.getString("YEAR_MONTH") + "</TD>");
        out.println("    <TD align=\"right\">" + RS_monthly.getInt("ORDER_TOTAL") + "</TD>");
        out.println("</TR>");
     };
     out.println("</TABLE>");
     RS_monthly.close();
    } catch(ClassNotFoundException e){
     response.setContentType("text/html");
     out.println("Error: "+ e);
   } catch(SQLException s){
     response.setContentType("text/html");
     out.println("Error: "+s.getSQLState()+","+s.getMessage()+" ("+s+")");
   } catch(Exception s1){
     response.setContentType("text/html");
     out.println("Error: " +s1.getMessage());
   };

  };

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

}
```

Appendix F: Simple Graph Example With Java2D API

```
/*
An Simple Graph example With Java2D API:
File Location:
C:\datamine\ simpleJava2DGraph.jsp
URL:
http://localhost:8082/datamine/simpleJava2DGraph.jsp
*/

<%@ page session="true" %>
<%@ page import="java.util.Date" %>
<%@ page import="java.util.Calendar" %>
<%@ page import="java.io.*" %>
<%@ page import="java.sql.*" %>
<%@ page import="java.text.SimpleDateFormat" %>
<%@ page import="java.text.*" %>

<%@ page import="java.util.*" %>


<%@ page import="java.io.OutputStream" %>
<%@ page import="java.awt.image.BufferedImage" %>
<%@ page import="java.awt.*" %>
<%@ page import="com.sun.image.codec.jpeg.*" %>
<%@ page import="java.awt.geom.*" %>
<%@ page import="javax.imageio.*" %>

<% // http://localhost:8082/datamine/simpleJava2DGraph.jsp
        BufferedImage bi = new BufferedImage(350, 250, BufferedImage.TYPE_INT_RGB);
        Graphics2D biContext = bi.createGraphics();
```

```
// Fill Rectangle
biContext.setColor(Color.cyan);
Rectangle rect0 = new Rectangle(0, 0, 400, 250);
biContext.fill(rect0);

// Fill Rectangle
biContext.setColor(Color.pink);
Rectangle rect1 = new Rectangle(25, 25, 315, 200);
biContext.fill(rect1);


// Draw line
biContext.setColor(Color.green);
GeneralPath p = new GeneralPath();
p.moveTo(40, 40);
p.lineTo(250, 40);
    p.lineTo(250, 180);

// p.closePath();
//   Use closePath only if to draw a closed line graph
    biContext.draw(p);

// Draw a triangle
GeneralPath p1 = new GeneralPath();
p1.moveTo(280, 50);
p1.lineTo(330, 50);
    p1.lineTo(330, 200);
    p1.lineTo(280, 50);
    biContext.draw(p1);
```

```
        // Draw Text String
        biContext.setColor(Color.black);
        java.awt.Font font;
        font = new java.awt.Font("times", 0, 20);
        biContext.setFont(font);
        biContext.drawString("Sample Java2D Text", 30, 100);

        // http://www.oreilly.com/catalog/java2d/chapter/ch04.html#35815
        // http://java.sun.com/products/java-media/2D/samples/suite/

        // draw Arc2D.Double
        // g2.draw(new Arc2D.Double(x, y,
        //              rectwidth,
        //              rectheight,
        //              90, 135,
        //              Arc2D.OPEN));
        // Arc2D.OPEN   Arc2D.PIE

        biContext.setColor(Color.red);

        biContext.fill(new Arc2D.Double(200.0, 150.0,
                    80,
                    80,
                    90, 135,
                    Arc2D.PIE));

        String ifile = "samplejava2d.jpg";
        File f = new File("c:\\datamine\\graphs\\" + ifile);
        ImageIO.write(bi, "png", f);

%>
```

```
Sample JAVA2D Graph <br>
<IMG SRC="/datamine/graphs/samplejava2d.jpg" ALT="Sample Java2D Graph">
<!-- // http://localhost:8082/datamine/simpleJava2DGraph.jsp -->
```

VITA

XIAN S. TAN

Candidate for the Degree of
Master of Science

Thesis: VISULIZATION OF A RESTAURANT MANAGEMENT DATA ANALYSIS USING JAVA TECHNOLOGIES

Major Field: Computer Science

Biographical:

Personal Data: Born in DaZhu, SiChuan Province, P. R. China, November, 18, 1963, the daughter of Cheng Liang Yao and Zhi-Wen Tan.

Education: Graduated 10th High School of DaZhu, SiChuan, China, July 1981; Received Bachelor of Science, Major in Chemistry From Nan Jing University, JianSu Province, China, July 1985; completed the requirements for the Master of Science degree at Oklahoma State University, Tulsa, Oklahoma in May, 2008.

Professional Experience: Teaching Assistant, Department of Applied Chemistry, Chengdu Institute of Technology, China, July, 1985 to July 1987; Lecturer, Department of Applied Chemistry, Chengdu Institute of Technology, China, August 1987 to April 1995.

Name: Xian S. Tan Date of Degree: May, 2008

Institution: Okalahoma State University Location: Stillwater, Oklahoma

Title of Study: VISULIZATION OF A RESTAURANT MANAGEMENT DATA ANALYSIS USING JAVA TECHNOLOGIES

Page in Study: 77 Candidate for the Degree of
Master of Science

Major Field: Computer Science

Scope and Method of Study: The purpose of this study was to help small restaurants to analyze their data to find the hidden information and patterns to increase sales, to improve customer service and customer satisfaction, and to increase business profitability. In this thesis a data mining system for restaurant management is built using Java Programming Techniques. Oracle database technology is selected as the RDBMS to store the data for the restaurant management system. A GUI-based Java Application is developed to input the data into the Database. Search functions and data mining features are built into the data entry application. A Web-based data mining application is developed using JSP pages with Apache Tomcat Server. The web-based data mining dynamically gets the data from the Oracle Database through JDBC. Tabular reports and graphs are created with JSP and HTML code, and are displayed in the Web-browser to allow easy navigation and drill-down of the data mining.

Findings and Conclusions: The results of monthly and daily sales, sales by zip codes, and customer order frequency charts, and top 10 customers order patterns are tabular reports and graphs presented to restaurant management to allow easy navigation and visualization. It helps them to make decision and overall management. An analysis of these reports shows how to use the information and hidden patterns to increase sales, to improve customer service and customer satisfaction, and to increase business profitability.

Advisor Approval:
George E. Hedrick