Implementing A Web-Based System Using Open-Source Software

Robert W. Ingram, (Email: ringram@cba.ua.edu), University of Alabama

Dale L. Lunsford, (Email: dlunsfor@cba.ua.edu), University of Alabama

Abstract

This paper describes a case to illustrate analysis and design issues involving a multi-tier e-commerce system. The case is designed for use in accounting systems and systems analysis and design courses. The case involves analysis of a sales order system that will be implemented using a web interface and relational database, conceptual design of the system, and implementation of the system. A variety of tasks are involved in the case, but an instructor can select the tasks of relevance in a particular course. Detailed teaching notes are provided with examples of primary deliverables and guidance on implementing the system using open source software.

Introduction

This paper describes a case that can be used in accounting information systems and systems analysis and design courses to illustrate analysis and design issues involving a multi-tier e-commerce information system. The case involves three sets of tasks: analysis of a sales order system to be implemented using a web interface and a database backend, conceptual design of the system, and physical design and implementation of the system. An instructor may choose to use all or any subset of the tasks.

The case provides for a variety of learning experiences. If all tasks are performed, students will obtain experience in:

- the analysis and design of multi-tier information systems,
- working with UML (Unified Modeling Language) and CASE (Computer Aided Systems Engineering) tool concepts,
- development of web-based systems,
- development of relational database systems,
- linking web and database systems through ODBC,
- use of SQL (Structured Query Language) to create, read from, and write to databases,
- programming business logic,
- identifying control issues in web-database applications.

Which tasks are emphasized depends on the instructor's goals for the case and course in which it is used. Basic familiarity with relational databases, CASE tools, html, and SQL are assumed. Some knowledge of programming also is useful, though the case can be adapted to use as a programming tutorial.

Readers with comments or questions are encouraged to contact the authors via email.

In the analysis stage, the case requires an examination of a company's need to develop a web-based sales ordering system to expand its market. In the conceptual design stage, the case requires students to develop an implementation plan that describes the components of the system and how they interact. In the implementation stage, the case requires development of a user interface, a database backend, and a business logic component to link the user interface and the database. Teaching notes describe issues students should consider and provide a tutorial on implementing the system using Open Source software components: Apache web server, Perl programming language for CGI (Common Gateway Interface) scripts, and MySQL relational database system.¹ Instructions for obtaining and installing the Open Source software are available from the authors at <http://grapeape.cba.ua.edu/csapubs/ringram/rbis1.html>.

Web-based applications are an important means of accessing business information. E-commerce represents an increasingly important part of the economy and is an obvious source of demand for web-based systems. In addition, companies may use the Internet for internal communications because of the ability of employees to access data from locations throughout the world. Companies like Oracle and SAP are increasingly focusing on the Web as a means of deploying their software products.

Vendors have developed integrated development environments to facilitate the creation of web applications that connect to databases (examples include Macromedia's Dreamweaver UltraDev and Microsoft's FrontPage). An advantage of these tools is that they simplify the programming of connections between web pages and databases, though the programs created may require modification to meet user needs. A disadvantage of these tools for educational purposes is that they can conceal the logical processes that are required to create meaningful business applications. The following case exposes students to the processes of analyzing and designing web-database applications. It focuses on the interactions of customers with a business and the development of a multi-tier system to automate those interactions.

Case Description

The Baba Blanket Company produces and sells high-quality wool blankets. It has operated for over 50 years from a single retail store. It accepts mail orders, but over 90 percent of sales or directly from the store. Though the company has been successful, its owners realize that their market is limited by location and that competition is increasing as more producers make their products available over the Internet. To increase its market, the company has decided to expand to an e-business operation. It plans to provide information about its products on the web and to provide a means for customers to order the products from its web site. The company will need to receive and record sales orders and maintain customer and product information.

A company employee will administer the system. Cost is a primary issue for the company. The system must be inexpensive to implement and manage. At the same time, the system must be flexible to allow for updates and expansion if the company decides it wants to add features or connect the order system to other parts of its information system.

Your employer, Delbert Consulting Company, has been hired to create a web-based information system for Baba to provide product information and to permit customer orders. You have been assigned the task of developing a prototype system for the project. Your tasks include:

¹ Our objective is not to foster use of these tools but to illustrate how realistic systems can be developed using freely available software that is platform independent, easily learned, and easily scaled for use on desktops or multiple server configurations.

- Preparing an analysis of the system that describes company and customer needs and how the system will meet these needs.
- Creating a conceptual design for the system that explains the operation of the system, specific activities the system will perform, and issues that will need to be considered in implementing the system.
- Developing an ordering system that implements your design. The implementation will involve a three-tier architecture with a web-based interface and a relational database.

Requirements of the project include a narrative report summarizing the analysis and describing, in general terms, the system that should be developed to meet the company's needs. The narrative should include a work flow diagram and a use case diagram and description. The conceptual design should include an entity-relationship diagram for the database. Also, it should identify and justify the software components that will be used to implement the system.

Your review of the company has produced the following information. Baba produces only one type of product, wool blankets. The blankets come in two types, solid and plaid. The solid blankets are available in red, green, and blue, and the plaid blankets are available with a dominant color of red, green, and blue. The company plans to charge \$150.00 for the solid blankets and \$175.00 for the plaid blankets. Shipping charges are included in the price, and the company will not collect sales taxes on Internet orders. The product numbers, descriptions, and prices are included in Table 1.

All Internet sales will be via credit cards. The company accepts American Express, Discover, Master Card, and Visa. Data to be obtained from customers include name, address, phone number, email address, credit card type, credit card number, and credit card expiration date. Customers should enter the quantity of each product they want to order and be able to review their order information before submitting the final order.

For purposes of the prototype, you do not need to be concerned about web site design, formatting, colors, images, and similar features. You should focus on functionality and ensuring that the system meets information requirements. Management wants the system to produce a sequential order number to uniquely identify each order. The order number and total amount charged to the customer should be reported to the customer as a final step in the order process.

Teaching Notes

Analysis

The analysis phase of the case requires students to examine company and customer needs. They should include the primary components of the system in their description and explain how these components will permit the system to meet the needs of the company and users. The analysis should identify issues that will be considered in designing and implementing the system. Some of these issues are:

- What information should be provided on the web site home page? This question should consider the information needs of customers about the company and its products and about using the web site.
- What data should be presented and obtained from the order page? This question also should consider the information needs of users and those of the company.
- What data should be included in the database? Database specifics are part of the design phase, but students should consider what data items will need to be stored in the database and where

these data items will come from. Will the customer enter them? Are they provided by the database? Will they be generated by the system?

The case requires students to prepare a work flow diagram. This diagram can be as formal or informal as the instructor desires. CASE tools often provide a work flow or activity diagram. These diagrams are flowcharts that identify tasks to be performed and decisions to be made, arranged in sequential order.²

A particular type of work flow diagram is the Line of Vision Diagram developed by IBM (Harmon and Watson, 1997, 100). It describes actions (represented by boxes) and decisions (represented by diamonds) across different participants in (or parts of) a process. Arrows are used to connect the boxes and diamonds to show the flow of activities. The vertical axis of the diagram is divided into panels for each participant group. Customers are shown in the top panel and the diagram focuses on how customers interact with an organization. Thus, the diagram is particularly useful for describing a sales order system. The horizontal axis in the diagram shows passage of time.

Figure 1 provides a work flow diagram for the sales order system. Because the participants in the system, other than customers, are parts of the computer system, the diagram describes the workflow in terms of system components. Activities provided by each component are described in the panel for that component. The activities appear in sequence from left to right. Variations in the activities are possible. Important issues for students to consider are what activities will occur in each panel. The three panels identify the three tiers of the system architecture. Consequently, the diagram is a useful starting point for students to consider the functions and advantages of a multi-tier system.

A use case diagram defines a sequence of events or transactions within a system. It depicts actors who are external to the system and interact with it. Use cases are represented by ovals. Each use case is a subset of the behaviors that occur in a system. Lines or arrows are used to connect actors with the use cases.

Figure 2 provides a use case diagram for the web order system. The actor in the system is the customer. The diagram illustrates primary ways in which the customer interacts with the system. Accordingly, it is useful for helping system developers identify primary activities that the system will perform.

Extended use case diagrams can be used to identify specific activities that occur within a system (Harmon and Watson, 1997, 117). Figure 3 provides an extended diagram that focuses on system components and describes the interaction among these components. It is useful for developing the system architecture. The three tiers of the system are evident in Figure 3.

The case also requires students to prepare a use case description. The description lists and explains steps in a process (Harmon and Watson, 1997, 119) associated with a use case diagram. Table 2 identifies the steps in the web order use case scenario. In addition to listing normal steps in a process, the description identifies other events that could occur. The description helps system developers think through how a system will be used and to identify functions that are required in the system.

² The modeling tools illustrated in this paper are examples of commonly used tools. The instructor may choose alternatives. A purpose of the tools is to require students to consider the tasks the system must perform as a basis for designing the system.

Conceptual Design

In the conceptual design stage, students should begin to identify how they will develop an information system to meet Baba's needs. From the analysis stage (and from case requirements), it should be evident that a three-tier architecture is appropriate. Accordingly, the system will include creation of a web site that will be called by a customer's web browser. The browser will provide the client interface. The web site will run on a web server and connect to a relational database system on a database server. Typically, the web server and database server would exist on separate computers and could even use different operating systems. They could reside on the same computer, however, and for purposes of this case, there is no particular reason that students need to focus on more than one platform.

Most of the design phase should focus on web server and database components. A variety of methods could be used to develop a web server interface and database connection. Typical methods are CGI scripts, Active Server Pages, and Java Server Pages.³ Each of these methods requires a means of connecting the web server with the database system. A common method for accomplishing this task is ODBC (open database connectivity). ODBC involves installing software on the web server that provides the capability of interacting with a particular type of database using SQL. In addition, many programming languages include libraries designed to facilitate database connectivity. These libraries include database independent (DBI) and database dependent (DBD) components. The DBI provides modules for programming database interaction without reference to a specific database vendor or ODBC program. The DBD component is specific to the ODBC and database implemented. It converts the generic program commands written by the developer to language understood by the ODBC and the specific database. Therefore, a developer can create a program that will run on a variety of database systems without having to modify the program for each system. Figure 4 illustrates the steps involved in connecting a program script with a database.

Business logic programs reside on the web server. These programs receive data from the client, process the data, and interact with the database by sending SQL commands. The programs return data from the database to the browser. The programs may include controls for verifying data before they are sent to the database.

Thus, students should identify the primary components of the system: a web server, programs to interact with the client and database, connection through ODBC, and a relational database system. Given the concerns of Baba about cost, students might consider the costs of various components. Some research by students might be revealing about the expense involved in purchasing, installing, and maintaining systems, especially relational database systems.

Another aspect of conceptual design is developing a database schema. Students should consider the structure of the relational database system they will recommend. Though variations are possible, the schema should include tables for maintaining product data, for recording order data, and for relating the products and orders.

Figure 5 contains an entity-relationship diagram required by the case. Diagrams can be produced in a variety of formats. The format depicted in the figure shows the cardinality of the relationships and identifies relationships as lines connecting the entities. Primary keys are identified by a "P" tag.

³ For a detailed description of using Visual Basic, Active Server Pages, and Access to develop web-database systems, see Morrison and Morrison (2000).

The Number table is included in the diagram as a means of creating unique order numbers. The table includes only one field. The value of the field is read from the database and incremented each time an order is placed. The value is then updated by adding one to the current value. This procedure is commonly used to create unique incremental id values.

From a database modeling perspective, the Customers table could be split into two tables, one for the customer data (agent) and one for the order data (event). The separation would allow the customer data to be stored as a profile that could be retrieved if the customer placed other orders by entering an identifier for the customer (phone number, email address). This feature is not implemented in the design described in the teaching notes. Because transactions are for cash, there is no particular advantage to separating the data for purposes of this case.

Another issue that students might consider in designing the system is the need for security. Security is especially important for e-commerce where credit card numbers are transferred across the internet and stored in a company database. Transmission security usually is achieved by encrypting the data before they are submitted. Systems that implement Secure Sockets Layer (SSL) provide a digital signature to verify their identity and provide the browser with a public key for encrypting data. The data are decrypted by the web server using a private key. Implementing encryption requires use of a web server with SSL capability. Netscape servers, Microsoft Internet Information Server (IIS), and Apache servers on Unix (among others) have this capability.⁴

Security of the database also is an important issue. Several companies have reported the theft of credit card data from their databases. The direct and reputation costs of these thefts can be enormous. Security is one reason that databases should be stored on separate computers from web servers. Web servers must be open to the Internet to provide access by customers. Databases should be placed behind firewalls that provide much higher levels of access control than web servers. Additionally, sensitive data could be stored in encrypted form in the database. Many database systems now provide this capability. Consideration of these issues is important, even if they are not designed into the system used in the case.

Physical Design and Implementation

The physical implementation of the system can involve a variety of software applications. This section describes a particular implementation that uses freely available open source software that is platform independent. Accordingly, the system can be created on almost any computer system without cost, a primary consideration in many classroom and business environments. Further, the system can be implemented on a desktop so that individual students can create and manage all aspects of the system.

The components of the system are Apache web server,⁵ Perl (practical extraction and reporting language), along with its DBI module, the MySQL relational database system, and its ODBC driver. Useful references for the software components include Descartes and Bunce (2000), Laurie and Laurie (1999), and Yarger, Reese, and King (1999).

Tables and fields should be created in the database. These may be added by connecting to the database server using a command window and entering SQL DDL (data definition language) commands. For example, the command to create the Number table is:

CREATE TABLE Number (OrderNum INT NOT NULL PRIMARY KEY);

⁴ Actual implementation of SSL is beyond the scope of this case.

⁵ Microsoft Personal Web Server could be used and is freeware: www.microsoft.com/windows/ie/pws/

The command creates a table named Number with a field named OrderNum that is an integer, that cannot contain a null value, and is the primary key. Additional commands would be entered to create the remaining tables.

An alternative approach is to use one of the GUI administrative tools available for MySQL. Readers familiar with SQL Server 7.0 Enterprise Manager will find the DBTools GUI to be very similar. These tools provide for the creation of a database server, connection to the server, and creation of databases, tables, and fields. Values are added to the tables using the SQL insert command. For example, the command:

INSERT INTO Numbers (OrderNum) VALUES (1);

initializes the value of the OrderNum field in the Numbers table to 1. The commands can be entered from the SQL window in the GUIs. Also, data can be imported from other sources.

An ODBC connection must be created for the database. The connection requires obtaining and installing the myodbc software. The DSN (Data Source Name) created in the ODBC installation is used in scripts to connect to the database. All scripts and the database are available from the authors at <http://grapeape.cba.ua.edu/csapubs/ringram/rbis1.html>.

Once the components are available, the remaining task is writing the scripts to create the web linkages. Though variations are possible, four programs are needed to implement the system described in Table 2. The first program, baba.html, is written in html and provides a home page for the web site. This page can provide whatever information the company wants and a link to the order page. This program should be placed in the Apache htdocs folder and can be accessed by entering <http://localhost/baba.html> in the web browser.

The second program, order.pl, provides a form for customers to enter their identification data and to specify the quantity of each product they wish to order. Annotations are provided in the program to describe the purpose of each statement. Comments follow the # sound. Perl scalar variables begin with \$. Array variables begin with @. Perl is not a strongly typed language. The variable type is determined by context and any variable can be a string. Local variables are designated using the "my" reference. Observe that there are two functions provided by the Perl program. One is to process data through interaction with the database. The other is to create the html code for the web page. Thus, the program responds to the customer's request to place an order by presenting a web page that contains a form for data entry and includes a description of the company's products from the database.

It is useful to run the perl programs from the browser as they are being developed to check for errors. The perl programs should be saved in the Apache cgi-bin folder and run from the web browser.⁶ The error.log in the Apache logs folder identifies errors in the scripts and is a valuable tool for finding errors. Programming scripts is best viewed as an iterative process of preparing a simple script, checking it for errors, correcting errors, and adding to the script. It is much easier to find errors as lines are added to a script than to debug a completed script as a whole. Though perl scripting programs are available, a standard text editor such as Notepad is sufficient for writing perl scripts.

Once the order form is submitted, a third program, confirm.pl, receives the data and returns it to the customer in a request for confirmation. In addition to providing a web page that contains the customer

⁶ The web server must be running before the scripts will work. To start Apache, go to Start, Programs, Apache Web Server, Start Apache.

information, the program stores the data in hidden form fields. Once the customer confirms the order by clicking on the Submit Order button, the data in the form fields is submitted to the next program for processing. Should the customer want to make changes to the data, he/she would return to the second program by clicking on the Back button and make the changes in the form fields.

The fourth program, submit.pl, receives the order data and submits it to the database. In addition, the program retrieves the order number from the Number table and includes this number in the data added to the Customers and OrderProduct tables. It then updates the Number table by adding one to the OrderNum variable. Finally, the program creates a web page that contains the order number and the amount charged to the customer's credit card.

Once the database is completed, students should examine the effects of entering data. The SQL SELECT command can be used to read records to examine how the system is operating. The web server access.log in the Apache logs folder provides data about connections to the database.

Summary and Extensions

This paper describes a case involving the analysis and design of a web-based sales ordering system that uses a relational database backend. The case provides an overview of the analysis and design process using a relatively simple application. The physical design and implementation phase of the case also can be used as a tutorial for a variety of applications that are commonly used in creating information systems: web development, relational database creation and management, and web scripting with ODBC.

The case could be extended to demonstrate additional concepts. For example, a script could be added that would generate a report showing sales order information for individual order numbers. The report would require a SQL statement to join the Customers and OrderProduct tables:

```
SELECT Orders.OrderNum Orders.OrderDate Orders.lname Orders.fname order-product.prodid  
order-product.quantity FROM orders order-product WHERE orders.OrderNum=order-  
product.OrderNum;
```

Thus, students could gain additional experience with creating queries, joining tables, and producing reports.

The case also could be extended to connect to other business processes in the company such as shipping and recording sales. These processes could be implemented by extending the database to include shipping and sales tables. Additional reporting capability could be added to view information on unfilled orders and product sales by querying the database.

Various control issues are considered in the case. For example, table locking is used to prevent transactions being corrupted by more than one customer placing an order at the same time. Data fields in the database should be set to Not Null and appropriate data types should be used to ensure data integrity. In addition, students can explore server logs and database access tables to examine the use and security of their systems. User accounts can be created in MySQL to grant users permissions to read and write to and to modify database components.

For courses in which students have had little programming experience, the case could be adapted as a programming tutorial. The instructor might provide the orders and confirmation programs to students and have them develop the submission program, for example. Or, the instructor might provide excerpts from the programs and require students to complete the programs.

The case is designed to serve a variety of purposes and should be adapted as needed by the instructor. Suggestions for modification and improvement are solicited.

References

1. Descartes, A. and T. Bunce. 2000. *Programming the Perl DBI*. Sebastopol, CA: O'Reilly.
2. Laurie, B. and P. Laurie. 1999. *Apache: The Definitive Guide*. Sebastopol, CA: O'Reilly.
3. Harmon, P. and M. Watson. 1997. *Understanding UML: The Developer's Guide*. San Francisco: Morgan Kaufmann.
4. Morrison, M. and J. Morrison. 2000. *Database-Driven Web Sites*. Cambridge, MA: Course Technology.
5. Yarger, R, G. Reese, and T. King. 1999. *MySQL and MSQl*. Sebastopol, CA: O'Reilly.

TABLE 1
Product Description

Product Number	Description	Price
BL001	solid red	150.00
BL002	solid green	150.00
BL003	solid blue	150.00
BL004	red plaid	175.00
BL005	green plaid	175.00
BL006	blue plaid	175.00

TABLE 2
Use Case Description

Normal sequence:

-
1. Customer opens Baba's order page in web browser and selects Place Order option.
 2. Product information is obtained by web server from database.
 3. Order form and product information appear on customer's browser.
 4. Customer enters order information and submits order.
 5. Order information and confirmation request appear on customer's browser.
 6. Customer approves order.
 7. Order data are transmitted by web server to database.
 8. Order number and amount charged appear on customer's browser.

Alternate sequences:

The customer does not approve order (step 5) and returns to correct order information and resubmits order (repeat beginning with step 3).

The customer fails to complete order.

FIGURE 1
Work Flow Diagram

FIGURE 2
Use Case Diagram

FIGURE 3
Extended Use Case Diagram

FIGURE 4
Web-Database Connectivity Components

FIGURE 5
Entity-Relationship Diagram

Notes