

Truck Security Tool for U.S. Customs and Border Protection (Cargo City)

Catalin Constantin
Purdue University
Calumet

Jerry Dekker
Purdue University
Calumet

Bonnie Rajchel
Purdue University
Calumet

Richard Sasko
Purdue University
Calumet

Ruijian Zhang
Purdue University
Calumet

Contact:

CONSTANTINC1@netmail.calumet.purdue.edu, DEKKERJRJ1@netmail.calumet.purdue.edu,
RAJCHELB1@netmail.calumet.purdue.edu, SASKOR1@calumet.purdue.edu, Zhang@calumet.purdue.edu

?

Abstract

As a result of terrorist attacks, national security has become more worried with the possibility of future occurrences and what actions can be taken to help prevent future occurrences. The U.S. Department of Homeland Security has requested proposals that provide a security tool to assist the U.S. Customs and Border Protection (CBP) to supervise shipments that enter into the United States. Cargo City will help protect the nation's water supply by tracking dry and liquid bulk shipments from origin to final destination.

At this time only 10% of shipments that enter the United States are randomly inspected as they cross the U.S. border. As an outcome of tracking bulk shipments from origin to final destination the chance of an improper shipment, which could be used to pollute the nation's water supply, slipping past the border is drastically reduced.

These improper shipments, whether they be intentional contaminants or toxic solvents, cause destruction to the water supply as well as the creatures that depend on that water supply. Any discarding of toxic solvents, from misuse or criminal intent, causes damage which is indistinguishable from that caused by those individuals or factions who have the objective of causing harm to the nation's water supply.

Cargo City is a centralized, Internet-based database that will serve as a security instrument to assist in the tracking of tankers and identification of the drivers. When a tanker is loaded at its origin a manifest, driver and tanker information have already been transmitted to Cargo City and approved by the U.S. Customs database (ACE). After the tanker is loaded, all openings are sealed with disposable barcodes. Cargo City communicates with the shipper, the receiver and ACE. Cargo City contains a database with the manifest, the tanker being used and the driver(s) information.

The spiral process model was chosen for this project as a result of the need to re-work and refine some of the coding already in existence for Cargo City. The languages used for this project are Java, HTML/JavaScript, JSP, JDBC/ODBC, Microsoft Access[®] for the database, and Apache Tomcat.

System Overview

The primary goal is to make available a means for the U.S. Customs and Border Protection to be able to help track the shipment of liquid and dry materials from origin to final destination. This tracking will help to reduce the chances of a terrorist attack or other illegal occurrences from contaminating the nation's water supply as an outcome of tankers that manage to slip past the current border control protection in place.

This will be accomplished by having Cargo City be a centralized server containing a database of shipping information that will be connected to ACE, shippers, carriers and receivers through the Internet (Figure 1). Cargo City collects information about the company, the cargo, the conveyance, the crew and the equipment needed for a particular shipment and passes that information on to ACE for acceptance/rejection before the shipment proceeds. This will help to centralize the shipping system, making it easier and faster to track shipments from origin to final destination.

The database will store information about shippers, drivers and equipment that has been pre-approved by ACE and that is in use on a regular basis. This will eliminate the need to retype every piece of information about the company, the crew or the equipment used when a new manifest is created. This will also help to reduce the number of errors transmitted to the ACE database that could affect the rate of approval for shipments.

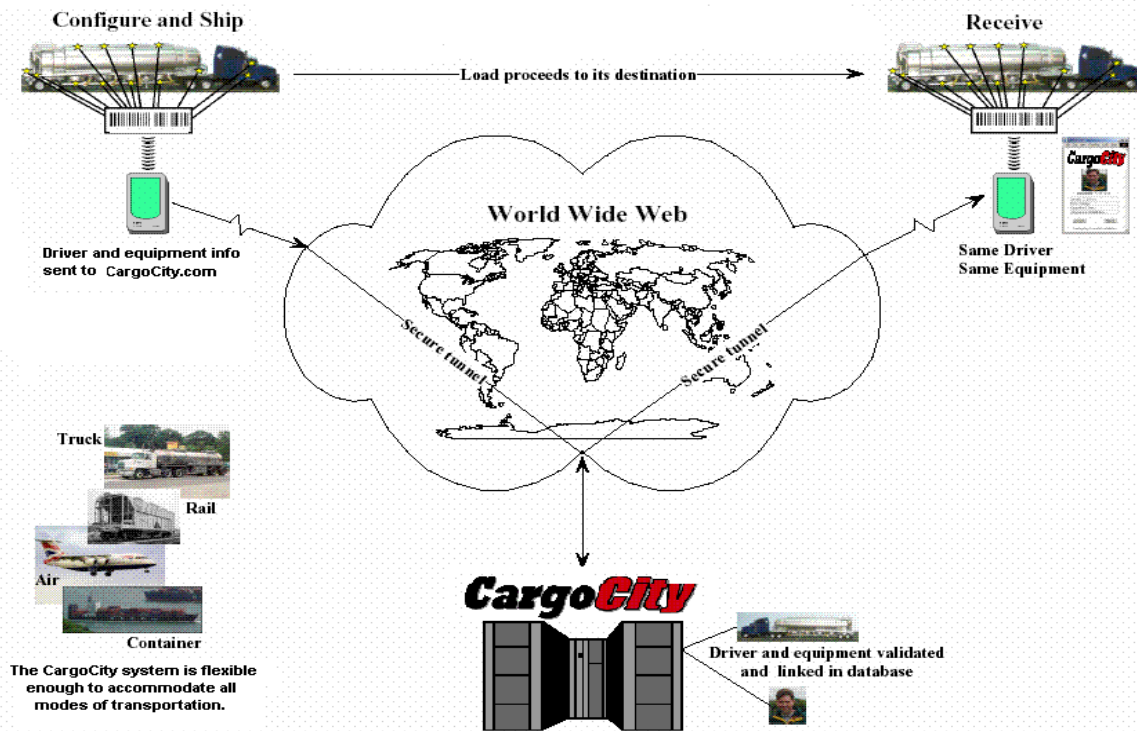


Figure 1.

The main use of Cargo City is to provide shipping security to protect the U.S. water supply from illegal and destructive contaminants. It also provides a way to maintain the shipping integrity of the private companies that deal with shipping products into the United States. The U.S. government will also be able to use Cargo City to test measures meant to help prevent future attacks from occurring as a result of contamination of the nation's water supply.

Cargo City will help to enhance the integrity level of the shipments since a tanker that does not have broken barcode seals will not have to be painstakingly inspected for possible missing or illegal items. This shipment integrity will help to reduce the work load of the inspection officials since they will not have to spend time inspecting the cargo of each tanker that is stopped, which will result in more time spent on tankers that are being monitored for possible illegal activities.

The companies using Cargo City will be able to accelerate the steps needed to ship or receive items. They will also be able to simplify the billing and invoicing process since the forms will be uniform for every shipment. Future tracking requirements that would be required by the U.S. government will be able to be easily integrated into the existing system thus making it easier and faster for the companies to meet any new guidelines that would be introduced.

The U.S. government will also be able to use Cargo City to interact with ACE to help in dealing with border control as well as for verifying the integrity of the shipment and validating the shipment container along with special notice if a shipment is lost or overdue in arriving at its specified destination.

The design methodology chosen for this project is the spiral model. This was chosen because the project was started by another team and the current team began with some code already in existence for Cargo City. The spiral model allowed the current team to refine or rework the existing code as well as refine and rework any new code created by revisiting the different stages of the design process. This provided a useful way to evaluate and reorganize the existing code for portability and efficiency.

This project is unique because there is currently no commercial program in existence that offers the services that Cargo City will provide. Cargo City will help to simplify the security checking process performed at the U.S. borders as well as by the receiver of the shipment. The U.S. border inspectors and the receivers will not have to examine the contents of the tanker in as much detail if the barcode seals have not been broken thus reducing the amount of time each inspector spends on a particular tanker.

Implementation

Cargo City is founded on the concepts of “central information storage” and “blind confirmation”. This involves storing all information about the cargo, transportation and driver in a central location which is verified by the end users on a need to know basis. The transportation carrier is able to log on to Cargo City to create, edit or view a manifest (Figure 2). When a carrier is contacted to transport a shipment, the carrier will log on to Cargo City to create a manifest containing information about the shipment, the conveyance of the shipment, the type of equipment that will be used, the crew member information and information about any other passengers that will be part of the trip.

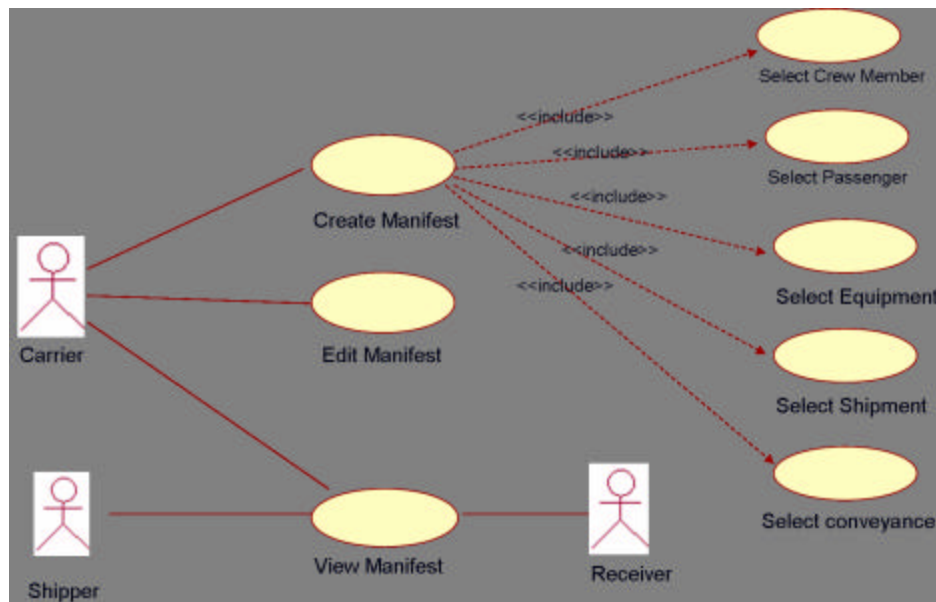


Figure 2. Manifest Creation

The carrier then submits the manifest to Cargo City which then communicates that information to ACE and awaits ACE’s acceptance or rejection of the shipment. When the shipment is accepted by ACE a notice is sent through Cargo City to the carrier that the shipment of the cargo can proceed. The carrier then arranges for the shipment of the cargo to begin.

The shipper is able to log on and view the manifest for items being shipped. The receiver is able to log on to Cargo City to view the manifests, scan the barcode into the system upon delivery and make note of the acceptance/rejection of a shipment (Figure 3). The Cargo City database administrator (DBA) will oversee the database.

The DBA will synchronize the Cargo City database with the ACE database, manage the transportation data, and monitor the transactions between the shipper, carrier, receiver and ACE with Cargo City’s database to correct any inaccuracies that may occur. Access to Cargo City will be controlled by username and password. The DBA will provide authenticated companies with the username and password. This will prevent unauthorized individuals from accessing sensitive information.

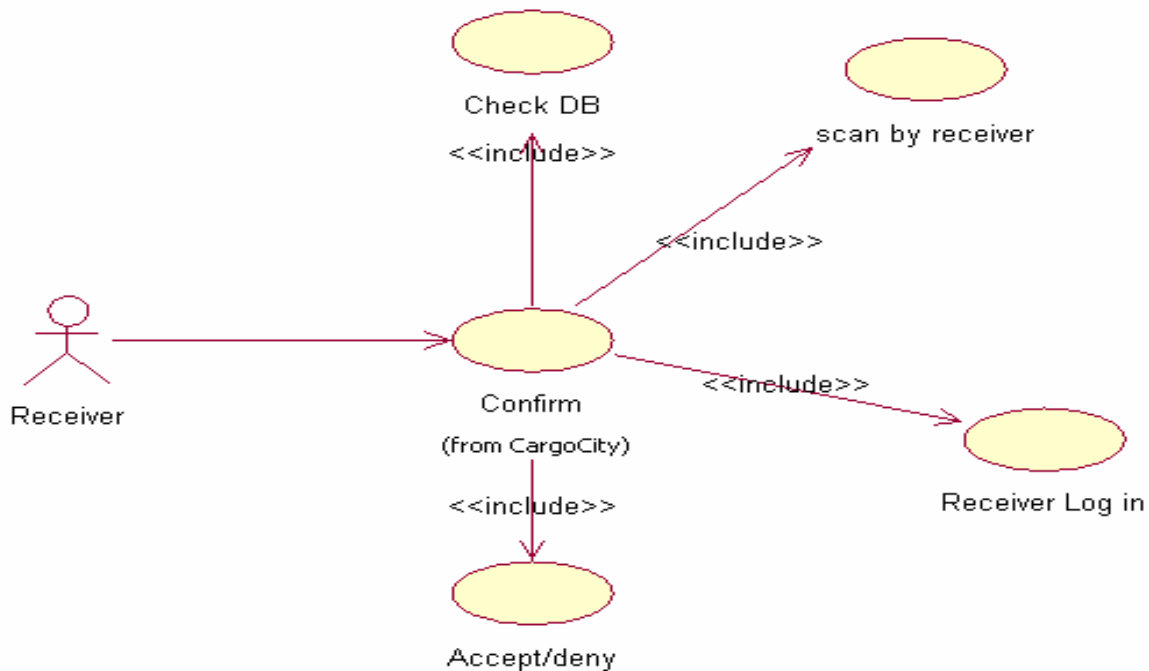


Figure 3. Receiver Actions

The database has a barcode field that is used as a key field to distinctively identify all information associated with a shipment. The databases for the company, the shipment, the conveyance, the equipment and the crew member information follow the requirements set forth by ACE for compatibility to permit ease of communication with ACE. Cargo City contains more fields in its database than is required by ACE to store information needed by Cargo City such as information about successful trips for the carriers and drivers. This helps to keep track of carriers and drivers that need to be monitored more closely than other carriers or drivers.

To ensure a secure and reliable connection to the ACE database from the front end system of Cargo City, a virtual private network (VPN) connection will be used. In busier ports a VPN concentrator may be deployed that allows for full time secure connections to the main

database. In locations that are not very busy, VPN client software or a standard web browser capable of connecting to the secure socket layer (SSL) will be loaded onto the Cargo City terminal that will be used for accessing the database. The use of either the VPN concentrator or the VPN client software will allow for security concerns to be alleviated due to the encryption involved.

The VPN chosen for Cargo City is the Cisco SSL VPN. The use of the Cisco SSL VPN allows for hundreds of simultaneous user connections (thousands if a cluster of Cisco devices is used on the receiving end) as well as high security ¹¹. Layer two tunneling protocol (L2TP) is used by the VPN to make a secure connection through the Internet by encapsulating the transmitted data using IPSEC.

The use of SSL technology allows for the use of the triple data encryption standard (3DES) which provided 192 bit encryption of the data. The use of this encryption inside of the L2TP that is used by the VPN architecture ensures a secure data transfer between the locations. The security of the data being transferred is of the utmost importance due to its sensitive nature and the team has taken this into account in choosing how the data will be transmitted between the remote location and the ACE database.

A test environment pseudo-remote database connection to simulate this functionality has been implemented. Although the current ACE database uses an X.12 EDI implementation for transactions, the team has chosen to implement a “counter-example” to this system based on XML.

This is a result of the belief that the X.12 standard is starting to show its age and that a more modern, Internet friendly common document exchange format makes more sense. This decision was made after research showed that various “ACE approved” software companies offer XML to X.12 EDI conversion services specifically aimed at making ACE document submission easier. The X.12 EDI codes for filing a manifest are (Figures 4 and 5):

- ✍ Messages transmitted to the U.S. Customs and Border Protection (CBP).
 - ✍ 309 Complete Manifest or Preliminary Shipments
 - ✍ 358 Truck Manifest (Trip/Consist)
 - ✍ 353 Event Advisory Message (Trip Commit)

- ✍ Messages transmitted from the U.S. Customs and Border Protection (CBP).
 - ✍ 997 Functional Acknowledgement
 - ✍ 355 CBP Acceptance/Rejection
 - ✍ 350 CBP Status Information

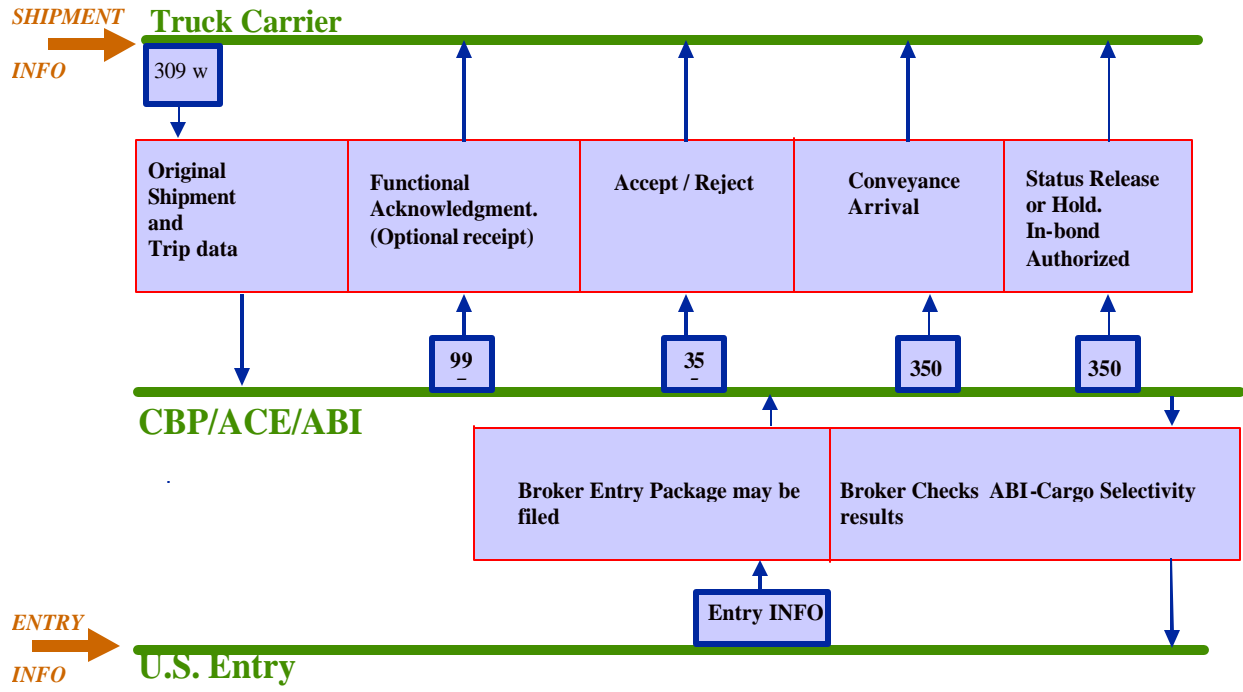


Figure 4.

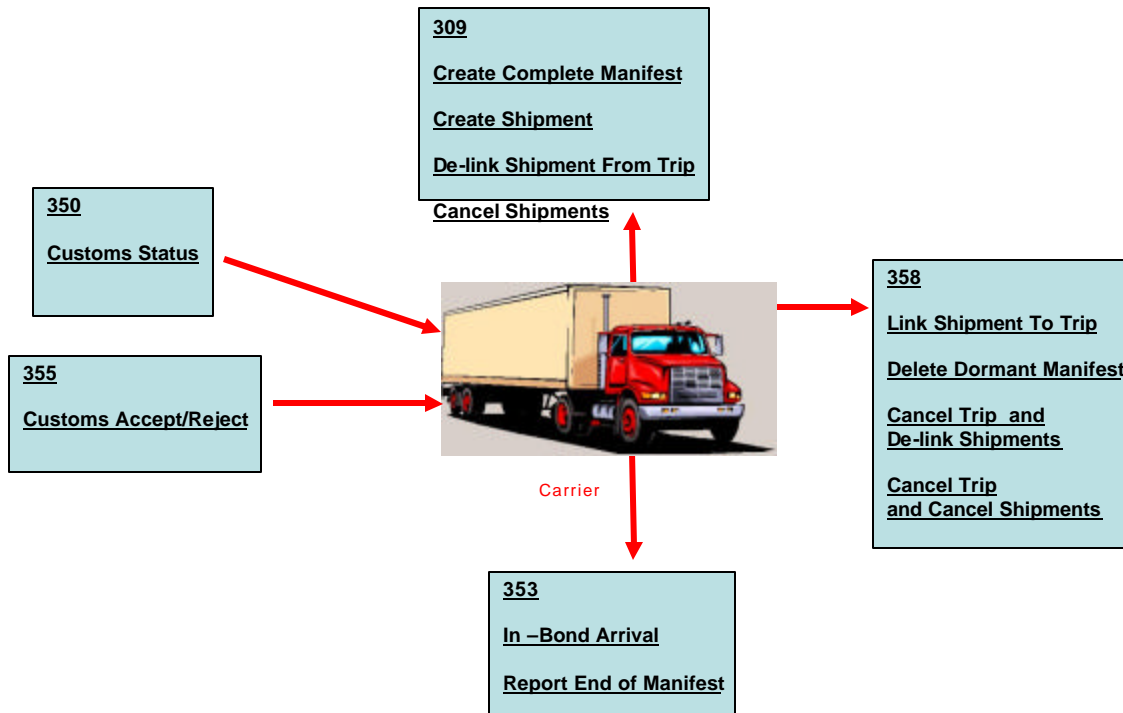


Figure 5.

Cargo City has been optimized to work well as an Internet application. As part of the optimization, the handling of multiple threads has been examined to prevent data corruption by multiple threads trying to access the same information at the same time. An extensive effort has been made to balance server memory usage against application responsiveness. This involved caching repetitive database requests of static data in memory as well as redesigning the way data is handled within a web session.

Much of the new interface design involves incremental updating without a full page reload from the server. The goal is to utilize the built in functionality available in the user side web browser to decrease the load on the Cargo City server. This assumes a relatively modern browser used by the client with respect to the fact that the client's browser must support HTTPS protocol for security reasons.

Cargo City is partitioned into multiple security sections to allow for various levels of access depending on the type of user. This multiple security scheme uses the existing Apache Tomcat security features and assumes a connection is made using HTTP protocol, which is a function of the parent web server associated with Cargo City.

Specific features have been made available only to some user groups while common elements have been made available to all. This involved extensive modification of the existing single-user interface including new code for the added functionalities and associated interfaces. For example, a carrier would need to have access to only the shipments for its company in Cargo City while local law enforcement officials would only need to be able to verify that the shipment is valid.

To verify that Cargo City works, each module of the system is tested separately to find and correct any errors. Then the module is incorporated into the system and further checked for any occurrences of problems. The grey box process, where the known input is observed through parts of the coding and the final output is verified for correctness, is used to validate that the modules and the system as a whole is working properly.

The languages chosen for this project are Java, HTML/JavaScript, JSP, JDBC/ODBC, Microsoft Access[®] for the database, and Apache Tomcat. A barcode scanner, which will rely on user side available drivers, can be used to input the barcode values into Cargo City. This will allow the shipper, carrier, receiver and local law enforcement officials to quickly and correctly enter the barcode into Cargo City for creating or verifying a shipment. There will also be a readable identifier on the barcode that can be used to input the barcode manually.

After being loaded at the origin, the tanker is sealed with a disposable barcode. The seal for the barcode, not the barcode itself, will have to be of a form that is hard to reproduce. This will help in the prevention of a tanker being opened and resealed by unauthorized individuals. These unique seals will help to reduce the chances of a tanker stopping somewhere along its route to dispose of any contaminants into the water way that it may be carrying.

The construction of the unique seal will be taken on by either the U.S. government or a private company with expertise in creating unique seals. The barcode itself has to be easy to produce so that either the carrier company or the U.S. government will be able to produce the quantity of barcodes needed per tanker without significantly raising the cost of production. For the barcode and seal security to be successful, the shipment and carrier companies will have to provide a level of integrity that is outlined in the guidelines set forth by the U. S. government.

Summary

Cargo City is approximately 75% complete with the working prototype being approximately 85% complete. To move forward with Cargo City, the code will continue to be upgraded and tested. Also, work on the web interface layout to create a more compelling look for Cargo City will need to be addressed as completion on the internal structure nears the finishing point.

The shipping industry is a very large and complex business so it is impossible to guarantee that Cargo City will be able to stop 100% of attempts to contaminate the nation's water supply. Cargo City is designed to help deter potential occurrences of contamination of the nation's water supply to the fullest extent possible. This is accomplished by sealing and tracking the shipping container as well as tracking driver information. As long as the shipper, carrier and receiver maintain a level of integrity and security required by the U.S., Cargo City will provide a high-quality way to help prevent those individuals or groups from contaminating the nation's water supply.

References

1. Barsolo, M., Compton, J. A., Johnston, K., Potts, S. and Redding, G. *HTML Complete*. San Francisco: Sybex, Inc.; 2003.
2. Braude, E. J. *Software Engineering: An Object-Oriented Perspective*. John Wiley & Sons, Inc.; 2001.
3. *Core JavaScript Guide 1.5*. Netscape Communications Corp.; 2000. Retrieved March 20, 2006 from <http://www.netcode.net/manuals/javascript/>
4. *Core JavaScript Reference 1.5*. Netscape Communications Corp.; 2000. Retrieved March 20, 2006 from <http://www.webreference.com/javascript/reference/core/>
5. Flanagan, D. *Java In A Nutshell; A Desktop Quick Reference*. Cambridge: O'Reilly; March 2002.
6. McLaughlin, B. *Mastering Ajax*. Retrieved March 20, 2006 from <http://www-128.ibm.com/developerworks/java/library/wa-ajaxintro2/index.html>
7. Murach, J. and Steelman, A. *Murach's Java Servlets & Jsp*. Fresno, CA: Mike Murach & Associates, Inc.; 2003.
8. *MySQL 5.1 Reference Manual*. MySQL AB; 2006. Retrieved April 1, 2006 from <http://dev.mysql.com/doc/refman/5.1/en/>

9. *Servlet and JavaServer Pages API Documentation*. The Apache Software Foundation; 2001.
Retrieved February 10, 2006 from <http://tomcat.apache.org/>
10. *Sun Java Documentation*. Retrieved January 30, 2006 from <http://java.sun.com>
11. Vogel, D., Davis, P., and Bornstein, M. *100 Questions and Answers on SSL-VPN*. Retrieved March 29, 2006 from http://www.cisco.com/application/pdf/en/us/guest/netsol/ns347/c381/cdccont_0900aecd800e493b.pdf.