Smart Sign Enhancement - Phase 2

State Job No. 134218

FINAL REPORT

Prepared in cooperation with the Ohio Department of Transportation and the U.S. Department of Transportation, Federal Highway Administration

September, 2007

by

Ping Yi, Chun Shao

The University of Akron

DISCLAIMER STATEMENT

ACKNOWLEDGEMENTS

We want to express our sincere appreciations to the SSOS project team, led by ODOT project liaison, Mr. Paul Trapasso, for their active participation in the development, testing, and training throughout the project. Our deep gratitude goes to Mr. Zack Mancini from the Division of Information Technology, for his effort to implement the SSOS improvements in ODOT's server and intranet system.

Special thanks go to ODOT's Research Office for providing the funding support and other assistance to the project. Support to the project from the College of Engineering and the Office of Research and Development at The University of Akron in student wages, tuitions, and lab facilities is acknowledged.

ABSTRACT

An on line ordering system, called Smart Sign Ordering System (SSOS), was developed by The University of Akron for Ohio Department of Transportation (ODOT) in 2004. Driven by the demand of managing planning, fabrication, packaging and delivery, "Smart Sign Enhancement – Phase 2" has been conducted from 2005 to 2007. This report discuss the detail of the project, including the requirement study, system modeling, software implementation, system maintain and customer training.

TABLE OF CONTENTS

Chapter I Background

Summary of SSOS (Phase I)

The Ohio Department of Transportation (ODOT) operates a Sign Shop to fabricate traffic signs ordered by ODOT's field districts. For many years, the districts submit standard sign orders to the Sign Shop, and special sign orders to the Central Office for review before they are forwarded to the Sign Shop, where all approved orders are processed by means of production planning, fabrication, and subsequent shipping. However, due to lack of data automation and inefficient data exchange and management, unnecessary delays and even errors occur in the current ordering process.

In a research project several years ago, the University of Akron developed an on-line traffic-sign ordering system for ODOT, the Smart Sign Ordering System (SSOS Phase I). The main focus of SSOS Phase I was to provide ODOT with a fully automated and networked sign ordering system for data exchange between ODOT's field districts, the Central Office, and the Sign Shop. SSOS is JSP applications based, implemented in a three-tier structure with Client, Application, and Database Servers and supported by ODOT's Sybase database to facilitate the creation of on-line sign orders. The main objective of SSOS is to improve the efficiency of the sign ordering process by reducing errors in order preparation and handling, speeding up review time, and making on-line modifications to the orders in line with current sign production methods at the Sign Shop.

As the title of the project indicates, SSOS is designed primarily for sign ordering management. The function of the system at the end of the phase I development includes mainly automation for ODOT districts in sign order preparation and handling. However, the system does not include many critical elements needed by the Sign Shop to organize, plan, produce and deliver traffic signs. The additional functions would be the focus of the phase II development.

SSOS Enhancement (Phase II)

The Sign Shop currently manages the data generated during the planning, production, and delivery process through use of a few tools (GQL-Graphical Query Language, Quatro-Pro and Paradox database), along with some manual procedures. A Paradox database program (JIMANI) stores the design specifications of those traffic signs which can be fabricated at the Sign Shop. However, because it was a simple program written a long time ago, not only is there lack of technical support to this program, but more importantly, it cannot be integrated into the three-tier server structure of SSOS supported by ODOT's Sybase system.

The Sign Shop also uses a GQL program for making production report and sale/cost summaries. For convenient use of the program, the Sign Shop needs to build application-specific queries for fast accessing and utilizing this program. The Sign Shop needs some technical guidance and assistance in this task.

During production, the Sign Shop needs to track the production status of each order in every processing step. The Sign Shop also wants to implement the new federal sign codes for all its producible signs. This would require change of the current sign codes in the SSOS database. In addition, the Sign Shop needs to have the ability to add, delete, and make changes to the design features of its signs as new standards, for that a simple method must be developed for accessing the SSOS database and making such changes.

Another most needed feature with the SSOS model is Component Pricing, by which the Sign Shop can determine the price of each order according to the sign characteristics. In other words, this flexible method of price determination can allow the Sign Shop to implement a better organized price structure based on different materials, production methods, and production procedures.

In summary, SSOS Phase II would concentrate specifically on the internal operations of the Sign Shop and development of additional functions to facilitate production planning, production management, and related administrative reporting.

Chapter II System Specifications

Order Lifecycle

First, a pending order is initially created online at the county office and reviewed by a local District Office. With necessary modification if needed, the order can be submitted to the Central Office for approval. After that, the Sign Shop will receive the approved orders and schedule the fabrication by grouping them into different production packages based on producing method and materials used. When the fabrication has been done the orders will be packaged based on the submitting district. After transferring the orders to the shipping department, the Sign Shop will be delivered to the respective district office. The receiving district will conclude the order when the signs have arrived and archived to history. During the lifecycle of the order, the four units (county office, District Office, Central Office, Sign Shop) can communicate about the order with each other by leaving comments in the order history field. Any mistake in the order at the sign shop can be rolled back to the previous stage by the authorized personnel. Every status change to the order will also be automatically documented in the history field. For the purpose of reducing order latency, the project committee has determined that only one item will be permitted in each order. The lifecycle is shown as Figure 1.

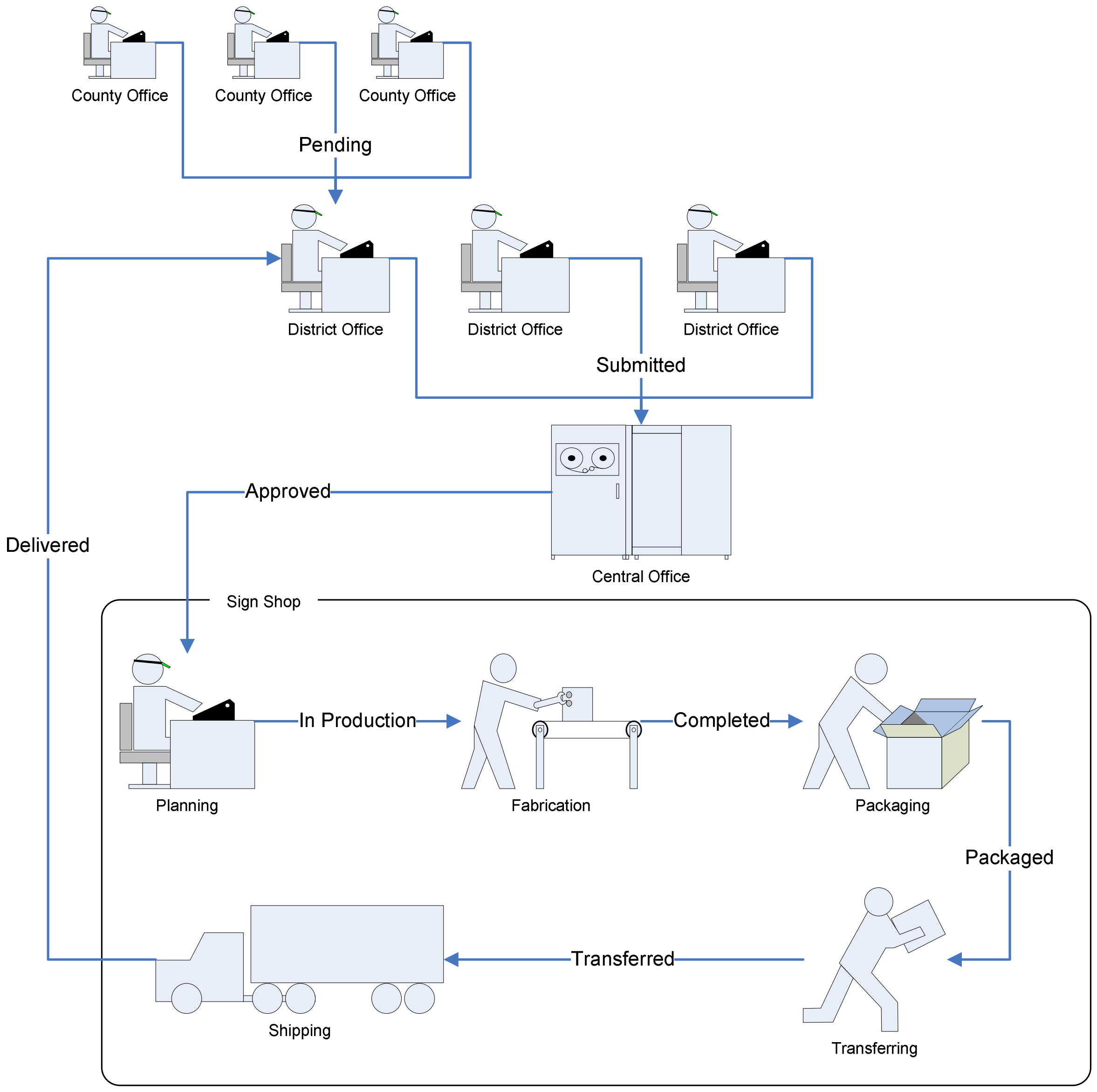


Figure 1 Lifecycle of a Sign Order

County Office

A county office has two functions, place an order and query the orders' status. When placing a new order, the users can locate the sign in the system through three ways: by the EMS number (the unique ID of a sign), by the sign code (including federal code) or by the legend on the sign. The default specifications of the sign in the system are imported from the JIMANI program at the sign shop. The county officer can either modify the specifications to make special order or keep the default to order a regular sign. What You See Is What You Get (WYSIWYG) technology is used for customizing the legend to help the user in design and editing. The cost of the order will be calculated automatically according to the material, production method, size and other characteristics of this sign.

Once the order is created, its status will be tracked and can be checked at any time. The detail information of each process step, such as create, submit, approve and etc. can also be displayed by checking the history of the order.

After the order is initially created it will be in a pending status for further modification by the district operator until it is finally submitted. Before the order is submitted, it can be reviewed and modified at any time.

District Office

A District Office unit has three functions. First of all, it will review all the pending orders placed by the county offices in the district or by its local warehouse. The District Office can modify the orders if necessarily and submit it to the central office. Secondly, it can track the order status while the order goes through the approval and fabrication process. Thirdly, upon the receipt of the manufactured signs, it will have the responsibility to confirm the reception and archive the orders through the online system.

After the order is initially created by county, it will be in a pending status for review and further modification if needed by the district operator until it is finally submitted (and the status of the order becomes “submitted”). Reviewing a previously “submitted” order is constrained to showing the order content with no modification permission to the district operator. When the order status changed to “transferred” and the district manager does receive the order, a confirm reception button will be shown and the user can choose to archive this order after the confirmation.

Central Office

The main function of the Central Office is to validate orders which are open for designation or special orders which modify the default settings, such as legend designable signs. The bypass signs such as warehouse signs whose attributes are all fixed will bypass the inspection by the Central Office and will be directly sent to the Sign Shop for manufacture.

Central Office can change the condition for the bypass and specify which signs can bypass and likewise which signs should be reviewed by the Central Office. Central Office will have the ability to review all "submitted" orders and have an approve button to process thereafter. For the sake of convenience, there is also a group approval function which can approve selected orders with one click. The Central Office will also have the permission to modify problematic orders before approve them.

For the purpose of material and resource management, the Central Office needs a generic search function that can return a certain group(s) of orders by the combination of district, material type, manufacture method, etc. This function is very useful when searching documented orders or creating summaries upon various attributes such as certain district or within certain dates.

Sign Shop

When the approved orders arrive at the Sign Shop, they need to be grouped by types of signs, materials used, and fabrication methods, etc. to facilitate production planning and material usage estimation. The Sign Shop has four sub-groups: production planning group, producing group, packaging group and transferring group.

Planning Group

Planning group will handle approved orders and make a production package of them based on material and production method. This group can also insert an order into existing production. It will also confirm the completion of the production process. There are six different types of production, Silk Screen, CBH (EX), CBH (FS), Purchase, Warehouse and Others, for planning group to choose to create a production plan.

Production Group

Production group will perform the fabrication and mark the corresponding step complete. As shown in Figure 2, there are different processes for each production according to its production type. For Silk Screen, there are three processes, Finishing, Stencil and Silk Screen. For Copy By Hand (CBH) type, there are two sub types, CBH (EX) and CBH (FS). CBH (EX) has two processes, Design and Shop; while two processes in CBH (FS) are Design and Layout. For other three types, Purchase, Warehouse and Others, there is only one step to complete the order. Once the final step of the order is completed, the order's status will automatically changed to completed.

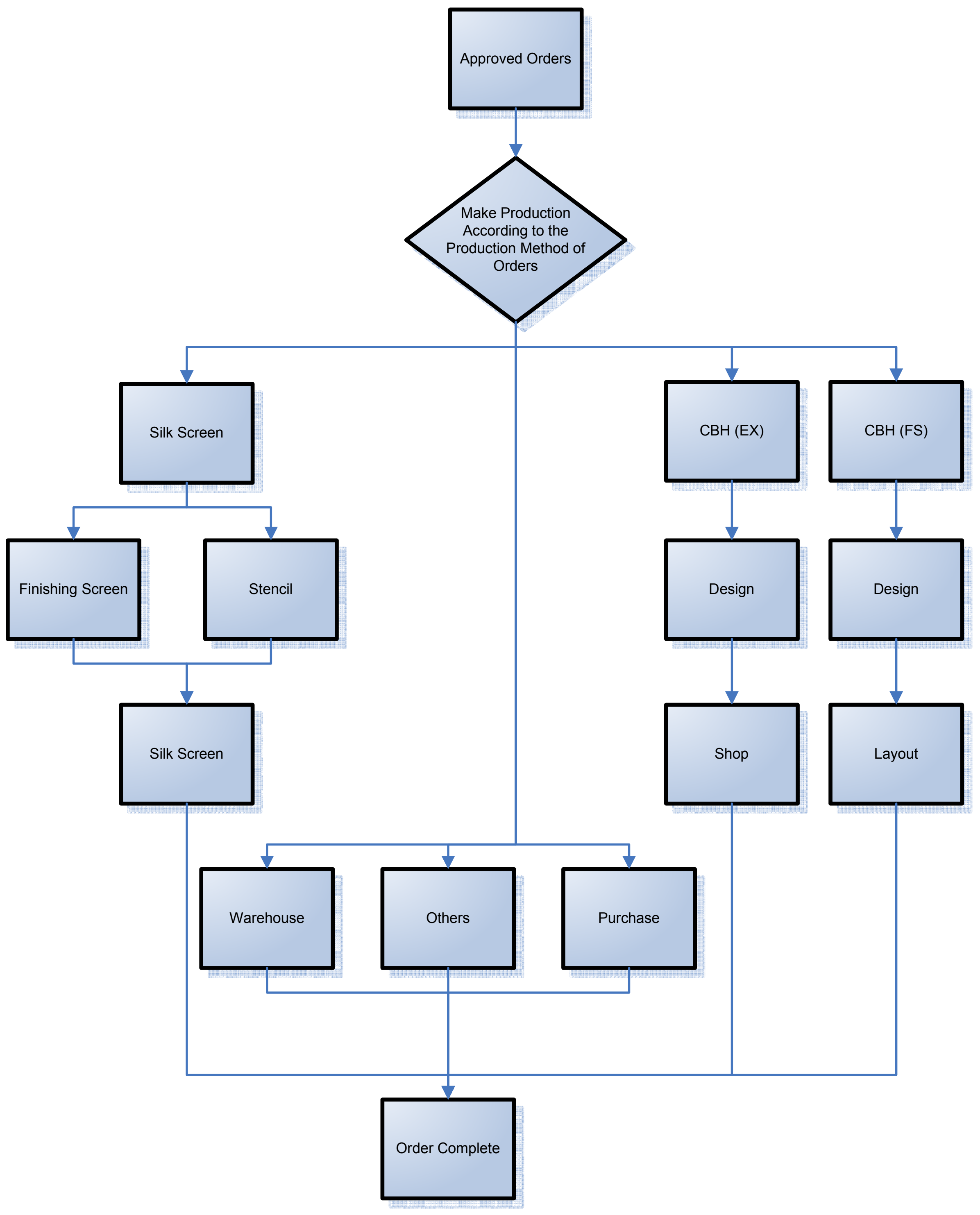


Figure 2 Sign Production Flow Chart

Packaging Group

Packaging group will gather all completed orders from production/warehouse and package them for the delivering to the districts. Orders can be packed differently from the production package generated by planning group.

Transferring Group

Transferring group has two major functions, transferring and delivering. Once receiving the package from packaging group, this group will mark the package as Transferred. After the signs have been sent to district or the third-party delivery company has picked up the package, the transferring group will mark the orders in the package as Delivered.

All of the four sub-groups will have the view-order function and the generic searching function. Those functions are critical to all of them. For example, Planning will need to group the orders cross districts based on the production methods and Packaging will need to regroup the orders by districts for delivering.

Data Manager

Data manager is an advanced group that will maintain system consistency and ensure correctness. This group mainly has three functions. First, the data manager can manage the signs information stored in the database. He can add, modify or delete the signs in the system with a provided operation interface. The second function is maintaining the sign cost table. This table saves the cost of each feature of the sign, such as background color, production method, material and etc. The data manager has the responsibility to keep the cost information up-to-date. Once the cost table is changed, the cost of newly placed orders will change automatically. Last but not least, data manager has the power to correct any mistake in the system. For instance, if an order with spelling error in customized legend has been approved, the data manager can undo the approval operation and thus the central office user can modify the order. Obviously, the data manager authority will be given only to a limited number of users to maintain system integrity.

Chapter III Tasks in Phase II

The SSOS Phase II enhancements consist of several tasks including requirement study, system design, functional implementation, testing, training, and deployment and support. We will discuss the detail efforts in each task in the following part.

Task 1 – Requirement Study

This task is to discuss and understand user needs at the Sign Shop. It is the foundation to understand and implement the enhancement of SSOS. According to software development procedure, the requirement study should be conducted before all the other steps, such as system design and functional implementation. However, it will also last to almost the end of the project because the change in requirement will always happen during the development of the system. For SSOS Phase II, a lot of time is used to revisit the production needs at the Sign Shop at the beginning, including planning and scheduling requirements, production procedures and order status tracking. Activities including touring the workshops and conducting individual discussions with technicians have been performed all along with the development. Additional meetings and group discussions were held during the functional implementation stage.

Task 2 – System Design

System design consists mainly of modifications to the system architecture (Figure 3) by changing the graphical user interface (GUI) design and modifying database. According to the requirement study, this project is the enhancement of SSOS, thus the basic architecture will be remained. It is still an Object-Oriented, three-tier (client, application and database) architecture, web server based system. In the application tier, we have two sub-tiers to separate data operation and user interface. However, the contents in each tier have been modified to support the requirements in Phase II. In database tier, new tables have been created and existing tables have been modified to save more information, such as sign cost table, order fabrication status and etc. Database operation sub-tier is also changed according to the modification in database structure. A lot of new classes have been added into the system and some new functions have been appended in existing classes. For user interface sub-tier, a number of web pages have been designed for the user to operate the system easily and properly. Detailed system design, such as database structure and GUI files will be described in Chapter V.

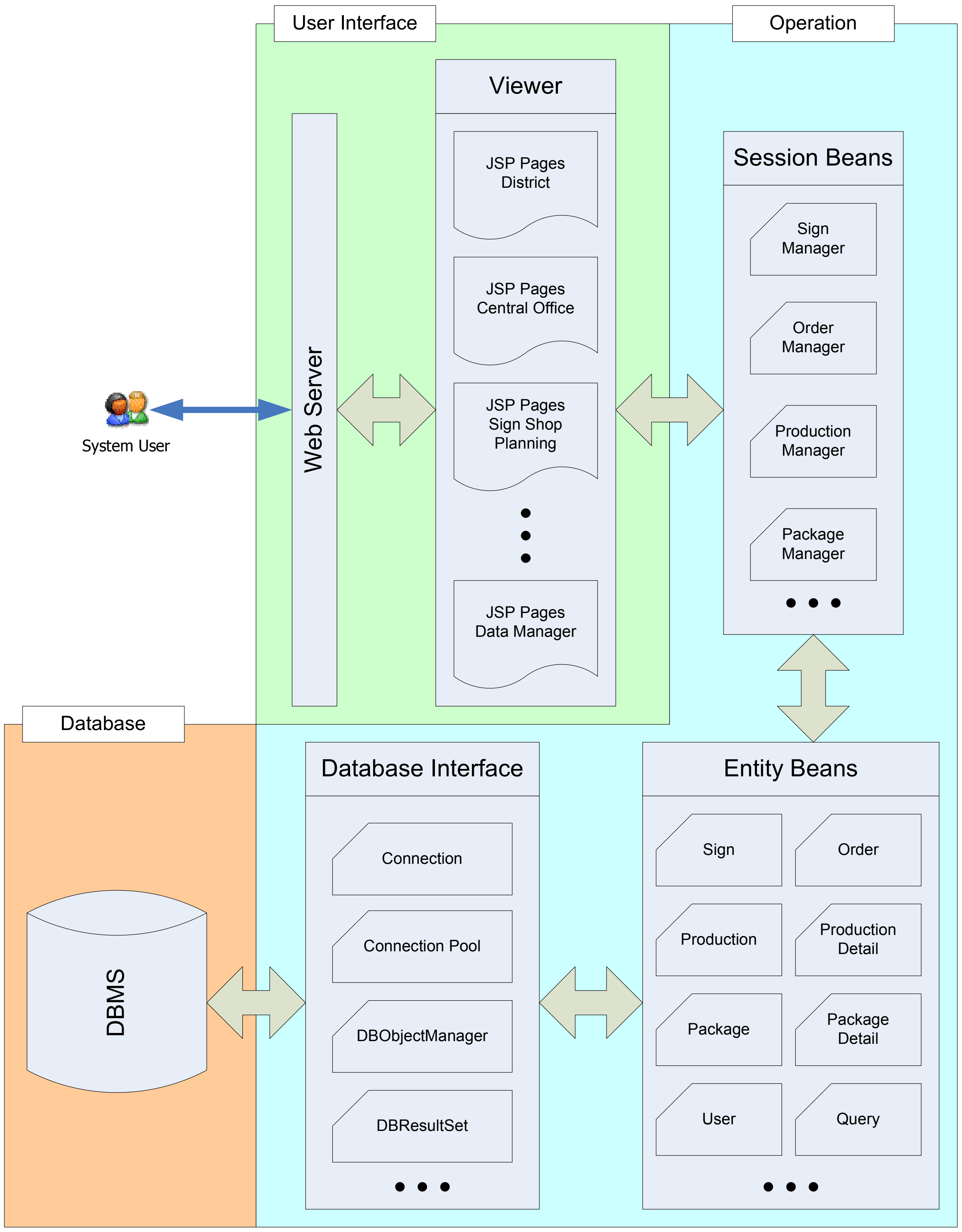


Figure 3   System Architecture

Task 3 – Functional Implementation

For consistency with the existing SSOS model, Java programming language will be used to implement the additional functions in the Phase II work. Since the Java programming language is an ideal tool for implementing Internet-based enterprise applications, we took advantage of the built-in Internet features, such as Java Servlet, JSP (Java Server Page) and EJB (Enterprise Java Bean), to make the final SSOS tool Internet friendly and user friendly.

The implementation is the major part of the effort of the project. Multiple related major subtasks will be discussed in the following.

Subtask 3.1 – Database Implementation

Database is the backbone of a three-tier (client, application and database) application. It stores application-critical data and links various modules of the application together. New data tables will be created and existing tables modified to support all the requested enhancements to the planning, production, and cost calculation functions. After working with Sign Shop and ODOT's DoIT, the specific format of the data tables and the data tables needed to be added or modified have been determined. Standard Query Language (SQL) files have been released to perform the update.

Subtask 3.2 – Replacement of JIMANI

JIMANI was implemented by the Sign Shop on the Paradox database many years ago. It is no longer supported since ODOT is phasing out Paradox. Due to the out-of-dated design of JIMANI, it cannot support many of the operations of SSOS which runs over the Sybase system. Part of the Phase II work is to replace all the needed functions from JIMANI to SSOS, such as sign specification, special queries, sign price computation, etc.

For sign specification, the Sign Shop maintains a table including more than 6,000 signs in JIMANI. We expand the sign information table in SSOS database first and transfer all the data in JIMANI. A SQL file is used to accomplish this job. Also, more than 5,600 pictures have been uploaded to SSOS web server to show the sign images when users query the sign information.

A customized-query function has been added during the SSOS Enhancement. With this function, user can create and save the preferred queries to replace the query function in JIMANI. It is very important to assist system user, especially the planning group in the Sign Shop to schedule the fabrication of orders.

Sign price computation table is also established in the SSOS. The price calculation is a component based procedure. The cost of each component is saved as one record in the price table. When there is a query of the cost of a sign, all the features (components) of this sign will be compared with the components saved in the data table. The summary of the cost of all fit components will be the unit price of this sign.

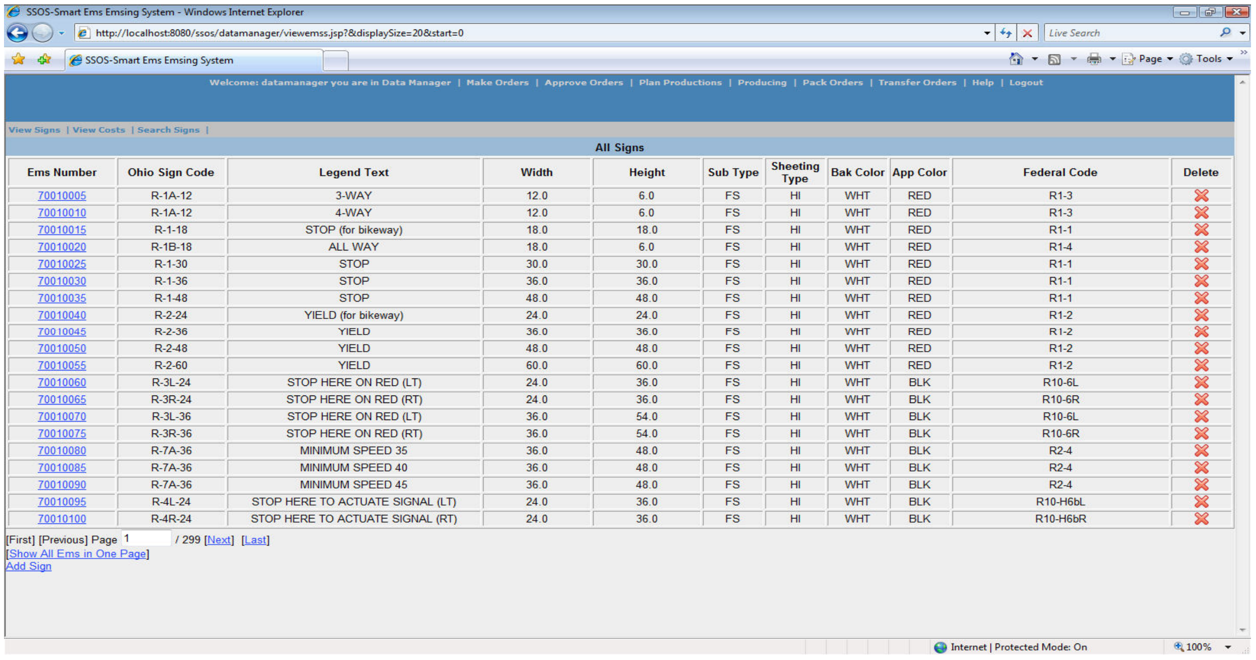
ID	PROD	SUB	GAUGE	SHEET	BG CLR	APP	APP CLR	SPECIAL	COST	UNIT
0									0.20	ft
1	SS								2.89	ft
2	CBH	FS							5.32	ft
3	CBH	PL							5.32	ft
4	CBH	EX							5.10	ft
5	BSO								2.53	ft
6	DC								2.00	ft
7		FS	63						1.26	ft
8		FS	80						1.57	ft
9		FS	100						1.93	ft
10		FS	125						3.03	ft
11		PL	50						1.25	ft
12		PL	75						1.50	ft
13		EX							5.40	ft
14				HI					1.16	ft
15				PZ					1.16	ft
16				DG					4.64	ft
17				DG	WHT				-0.79	ft
18				DG	FLO				-1.89	ft
19				NR					0.53	ft
20				EG					0.78	ft
21						HI			0.29	ft
22						PZ			0.29	ft
23						DG			1.16	ft
24						DG	WHT		-0.20	ft
25						DG	FLO		-0.47	ft
26						NR			0.15	ft

27						EG			0.20	ft
28						EC			1.00	ft
29								HDW	0.25	ft
30								HG	0.15	ft
31								C2	0.25	ft
32								C3	0.50	ft
33								C4	0.75	ft
34								PDC	0.90	ft
35								DF	0.75	per

Table 1 Sample Pricing Table

Subtask 3.3 – Building Editing Capabilities of Data Tables

For data application consistency and maintenance convenience, the Sign Shop needs to have the capability to fully manipulate data in some data tables, including addition, deletions and value changing in the data cells. Sign information table and price table are the most frequently maintained tables in SSOS. Through the JSP interface, the capability of editing data table is limited to data manager group. The interface of manage sign information is shown in Figure 4.


SSOS-Smart Ems Emsing System - Windows Internet Explorer
http://localhost:8080/ssos/datamanager/viewemss.jsp?&displaySize=20&start=0
Live Search
SSOS-Smart Ems Emsing System
Page
Tools
Welcome: datamanager you are in Data Manager | Make Orders | Approve Orders | Plan Productions | Producing | Pack Orders | Transfer Orders | Help | Logout
View Signs | View Costs | Search Signs |

All Signs

Ems Number	Ohio Sign Code	Legend Text	Width	Height	Sub Type	Sheeting Type	Bak Color	App Color	Federal Code	Delete
70010005	R-1A-12	3-WAY	12.0	6.0	FS	HI	WHT	RED	R1-3	✗
70010010	R-1A-12	4-WAY	12.0	6.0	FS	HI	WHT	RED	R1-3	✗
70010015	R-1-18	STOP (for bikeway)	18.0	18.0	FS	HI	WHT	RED	R1-1	✗
70010020	R-1B-18	ALL WAY	18.0	6.0	FS	HI	WHT	RED	R1-4	✗
70010025	R-1-30	STOP	30.0	30.0	FS	HI	WHT	RED	R1-1	✗
70010030	R-1-36	STOP	36.0	36.0	FS	HI	WHT	RED	R1-1	✗
70010035	R-1-48	STOP	48.0	48.0	FS	HI	WHT	RED	R1-1	✗
70010040	R-2-24	YIELD (for bikeway)	24.0	24.0	FS	HI	WHT	RED	R1-2	✗
70010045	R-2-36	YIELD	36.0	36.0	FS	HI	WHT	RED	R1-2	✗
70010050	R-2-48	YIELD	48.0	48.0	FS	HI	WHT	RED	R1-2	✗
70010055	R-2-60	YIELD	60.0	60.0	FS	HI	WHT	RED	R1-2	✗
70010060	R-3L-24	STOP HERE ON RED (LT)	24.0	36.0	FS	HI	WHT	BLK	R10-6L	✗
70010065	R-3R-24	STOP HERE ON RED (RT)	24.0	36.0	FS	HI	WHT	BLK	R10-6R	✗
70010070	R-3L-36	STOP HERE ON RED (LT)	36.0	54.0	FS	HI	WHT	BLK	R10-6L	✗
70010075	R-3R-36	STOP HERE ON RED (RT)	36.0	54.0	FS	HI	WHT	BLK	R10-6R	✗
70010080	R-7A-36	MINIMUM SPEED 35	36.0	48.0	FS	HI	WHT	BLK	R2-4	✗
70010085	R-7A-36	MINIMUM SPEED 40	36.0	48.0	FS	HI	WHT	BLK	R2-4	✗
70010090	R-7A-36	MINIMUM SPEED 45	36.0	48.0	FS	HI	WHT	BLK	R2-4	✗
70010095	R-4L-24	STOP HERE TO ACTUATE SIGNAL (LT)	24.0	36.0	FS	HI	WHT	BLK	R10-H6bL	✗
70010100	R-4R-24	STOP HERE TO ACTUATE SIGNAL (RT)	24.0	36.0	FS	HI	WHT	BLK	R10-H6bR	✗

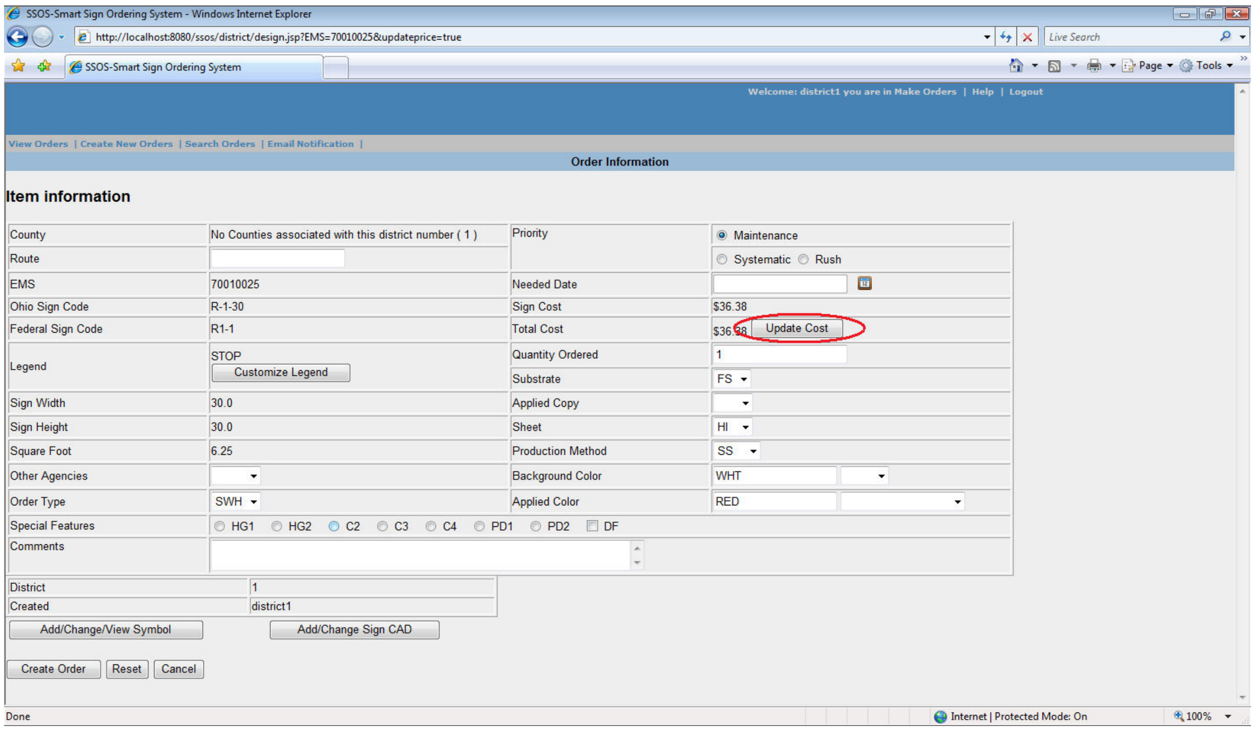
[First] [Previous] Page 1 / 299 [Next] [Last]
[Show All Ems in One Page]
Add Sign

Figure 4 Sign Information Management

Subtask 3.4 – Building the Component Pricing Scheme

The SSOS model is enhanced with the capability of calculating sign prices based on sign characteristics. This feature is also known as Component Pricing where each component of a traffic sign (i.e., substrate, production method, reflective sheeting, etc.) is assigned a default cost in the price table. SSOS will compile, calculate and assign prices based on the characteristic values used in each individual traffic sign. Based on the sample price table shown in Table 1, if there is a sign which production method is SS (Silk Screen) and the substrate is FS (Flat Sheet), the gauge is 63, sheeting type is HI (High Intensity), background color is WHT (White) and applied color is RED, and no special features; then the rule 0, 1, 7, 14 will fit the criteria and the unit cost of this sign will be \$0.20+\$2.89+\$1.26+\$1.16=\$5.51. If the size of this sign is 0.5 ft^2, the cost of each sign will be \$2.755.

Figure 5 show the access of this component pricing scheme in the user interface. The button in the red cycle called "Update Cost" will update the order/sign's cost based on the scheme described above.


SSOS-Smart Sign Ordering System - Windows Internet Explorer
Live Search
SSOS-Smart Sign Ordering System
Page
Tools
Welcome: district1 you are in Make Orders | Help | Logout
View Orders | Create New Orders | Search Orders | Email Notification |
Customize Legend
Update Cost
Add/Change/View Symbol
Add/Change Sign CAD
Create Order
Reset
Cancel
Done
Internet | Protected Mode: On
100%

http://localhost:8080/ssos/district/design.jsp?EMS=70010025&updateprice=true

Order Information

Item information

County	No Counties associated with this district number (1)	Priority	[✓ Maintenance]
Route			Systematic Rush
EMS	70010025	Needed Date	
Ohio Sign Code	R-1-30	Sign Cost	\$36.38
Federal Sign Code	R1-1	Total Cost	\$36.38
Legend	STOP	Quantity Ordered	1
		Substrate	FS
Sign Width	30.0	Applied Copy	
Sign Height	30.0	Sheet	HI
Square Foot	6.25	Production Method	SS
Other Agencies		Background Color	WHT
Order Type	SWH	Applied Color	RED
Special Features	HG1 HG2 [✓ C2] C3 C4 PD1 PD2 DF		
Comments			

District	1
Created	district1

Figure 5 Price Calculation Scheme

Subtask 3.5 – Building Order Production Monitoring Modules

This is to build an Order Monitoring Module to facilitate tracking of the different status involved in all the stages in an order. The status diagram of an order is shown in Figure 6. Any process to the order by any one will be recorded in the order history table and can be check by anyone who has the access to that order. So not only the district user can track their orders' status, but also the manager in the Sign Shop can check the progress of the fabrication of each order.

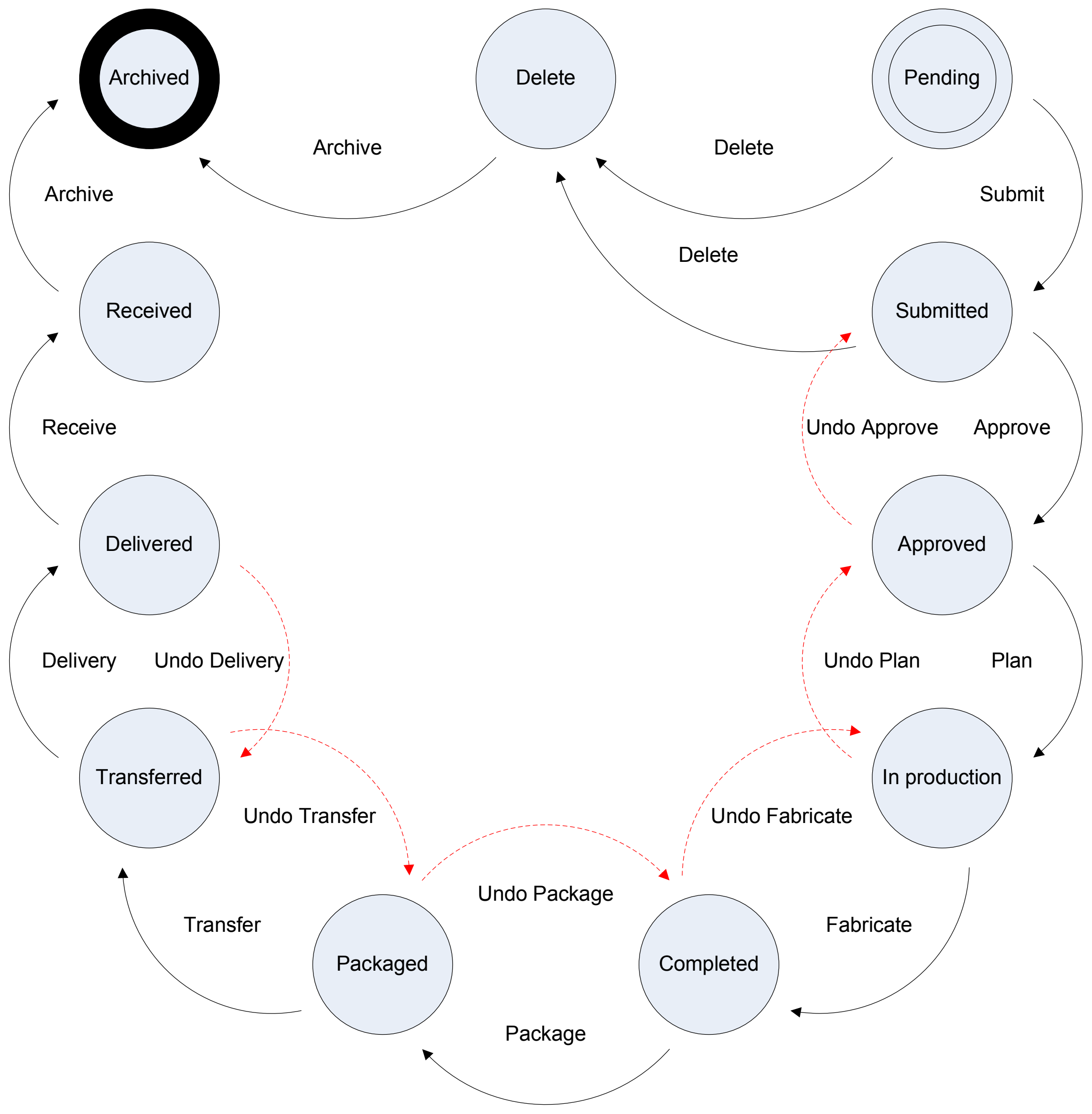


Figure 6 Order Status Diagram

Functional implementation includes much more subtasks which will not be discussed in detail in this report. For detailed database and application implementation information, please reference to Chapter V.

Task 4 – Testing

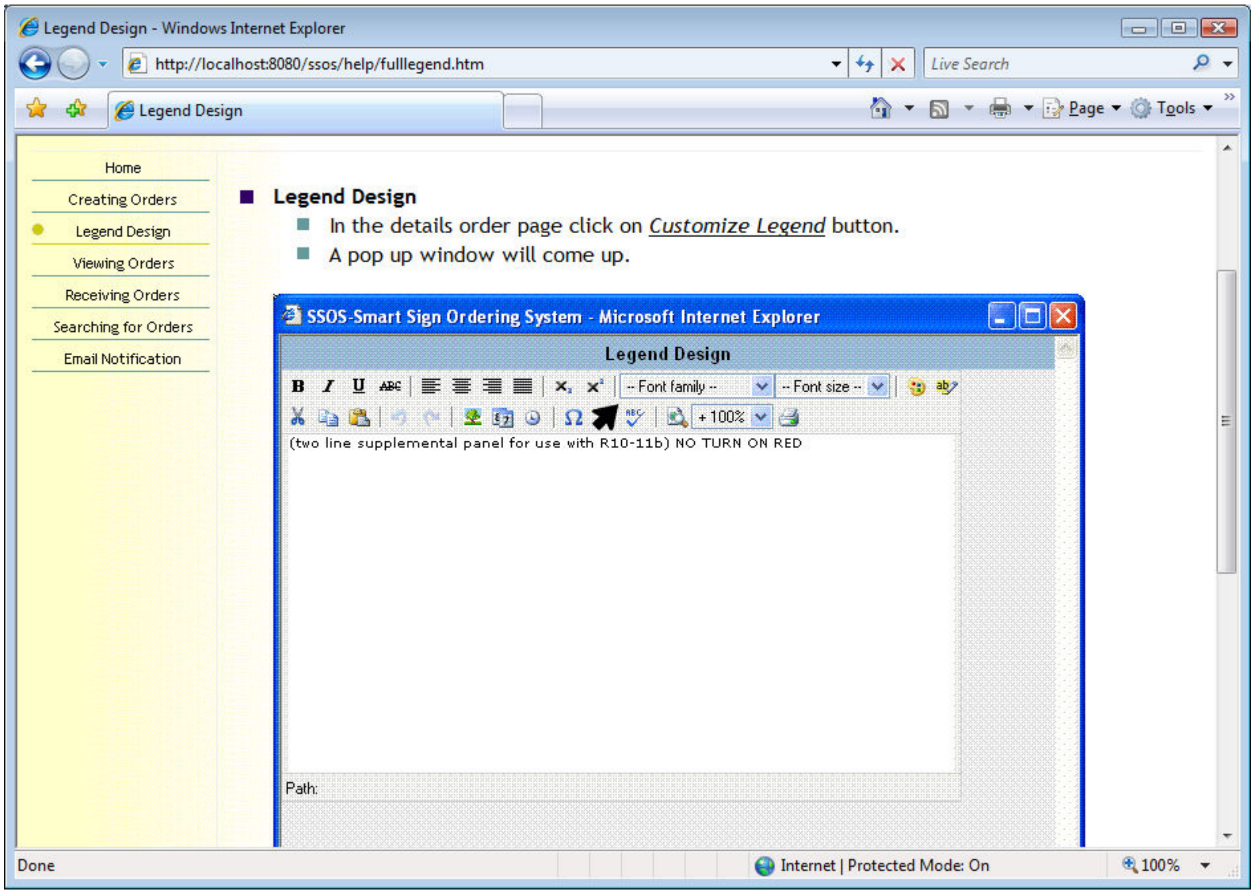
Two types of tests are conducted to guarantee the correctness and robustness of the SSOS model: the programmers' internal test and the users' acceptance test.

During the SSOS Phase II development, our programmers internally performed unit tests, modular tests and system tests to the program. A unit test is performed when a functional unit of the program is implemented, and a modular test is conducted when all of the functional units of a module are completed. System tests verify the functionality of the complete system when all of its modules are finished and connected together.

The user acceptance test is a full phase of testing of the program by the Sign Shop. The primary objective of the user testing is to evaluate functional capability of the overall program from the users' perspective. It is focused on correct workflow, improved efficiency in sign production and inventory management, and application friendliness as well as human error reduction. The project research team and the Sign Shop exchanged ideas to answer questions and fix bugs in a timely fashion.

Task 5 – Training

The project development team provided on-site user training to district office staff. More training sessions will be given in the extend period of this project. At the mean time, to facilitate the training, we developed a user's help documents online as part of the SSOS system. Multimedia methods such as image and video are used to build this online help document. The help documents include step-by-step instructions on how to use the program and case studies designed to showcase how to perform specific operations. The screen snap of help system is shown in Figure 7.


Legend Design - Windows Internet Explorer
http://localhost:8080/ssos/help/fulllegend.htm
Home
Creating Orders
Legend Design
Viewing Orders
Receiving Orders
Searching for Orders
Email Notification
SSOS-Smart Sign Ordering System - Microsoft Internet Explorer
Done
Internet | Protected Mode: On

Legend Design

- In the details order page click on *Customize Legend* button.
- A pop up window will come up.

Legend Design

(two line supplemental panel for use with R10-11b) NO TURN ON RED

Figure 7 SSOS Online Help Documents

Task 6 – System Development and Operational Support

Now the SSOS is deployed in a test server of ODOT for system testing. As important as we understand this is, we will provide a 90-day onsite (connection at the Sign Shop or through a field office), telephone and e-mail support of the system. In the event that a major systems glitch occurs or a flaw is detected while operating the software, we will begin repairs to the system within 2 working days.

During the Phase I development, we provided support to the Sign Shop over changes to the system features and continued the support to facilitate sign production after the completion of the Phase I project. As the new functions are being tested now, we will provide continued enhancements to the existing functions (such as spelling errors, column widths and title errors, issues with printing and page sizes, etc.).

Chapter IV Applied Technologies

Java Technology

A platform is the hardware or software environment in which a program runs. Some of the most popular platforms are Windows, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components: The Java Virtual Machine (Java VM) and the Java Application Programming Interface (Java API). The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages. Figure 8 depicts the Java platform architecture. As the figure shows, the Java API and the virtual machine insulate the program from the hardware [1]. Thus the system build in Java is platform free.

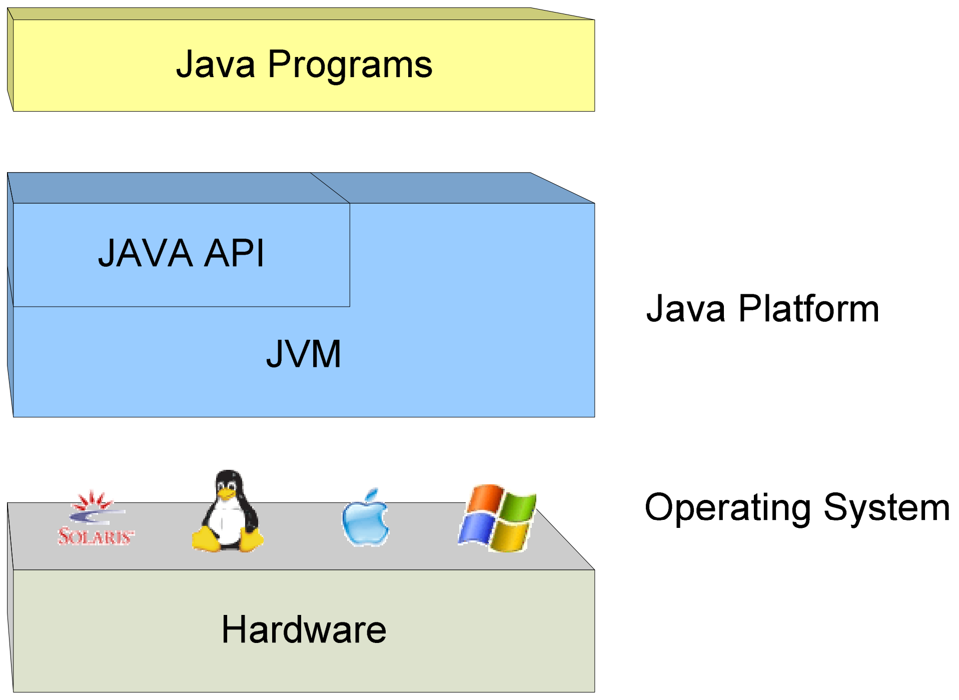


Figure 8 Java Platform

Servlet/JSP Technology

A servlet can be thought of as a server-side applet at a superficial level. Servlets are loaded and executed by a web server in the same manner that applets are loaded and executed by a web browser. Compared to most of the server side languages, servlets have the following advantages [2]:

1. Servlets are persistent.

Servlets are loaded only once by the web server and can maintain services (such as a database connection) between requests. CGI scripts, on the other hand, are transient. Each time a request is made to a CGI script, it must be loaded and executed by the web server. When the CGI script is complete, it is removed from memory and the results are returned to the client. All program initialization (such as connecting to a database) must be repeated each time a CGI script is used.

2. Servlets are fast.

Since servlets only need to be loaded once, they offer much better performance over their CGI counterparts.

3. Servlets are platform independent.

Servlets are written in java, which inherently brings platform independence to your development effort.

4. Servlets are extensible.

Since servlets are written in java, this brings all of the other benefits of java to your servlet. Java is a robust, object-oriented programming language, which easily can be extended to suit your needs.

5. Servlets are secure.

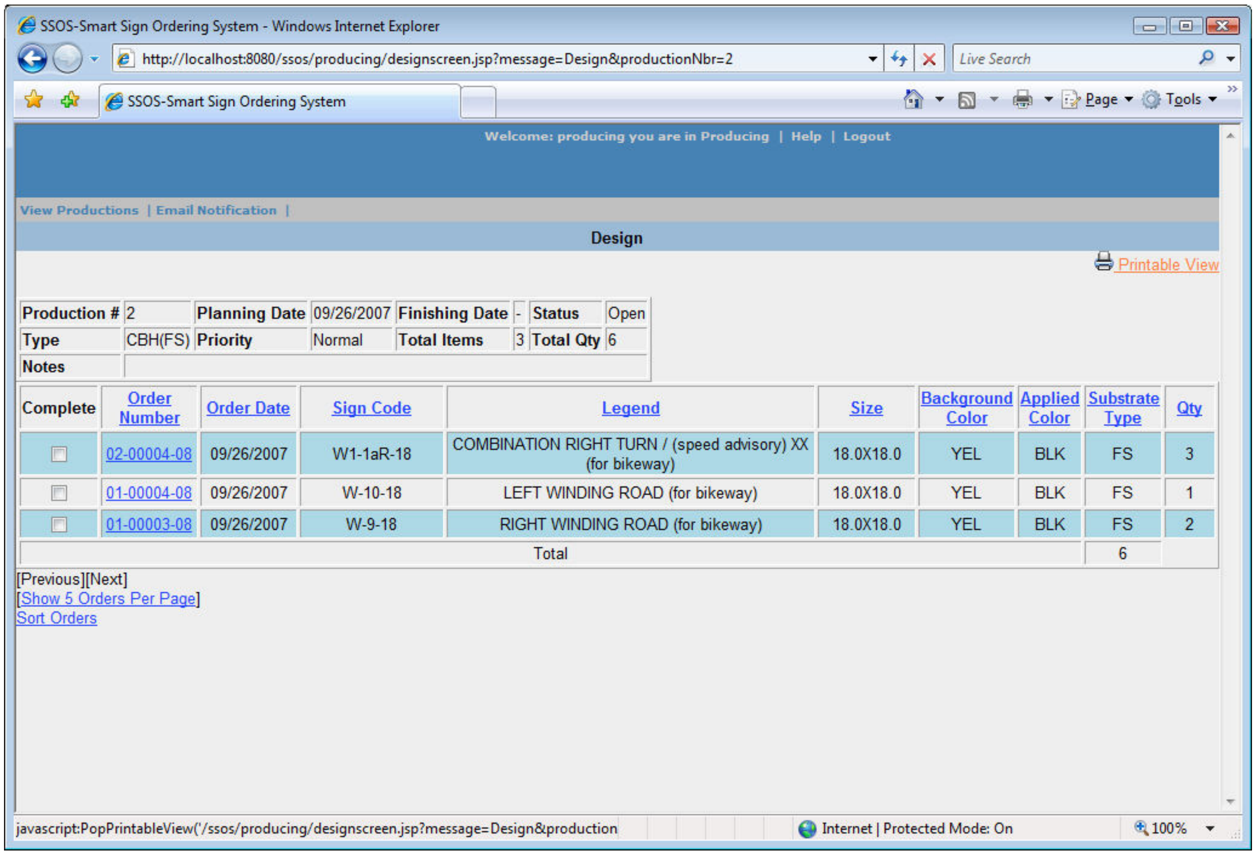
The web browser does not communicate directly with a servlet. The servlet is loaded and executed by the web server. This brings a high level of security. This means that if the web server is secure behind a firewall, then your servlet is secure as well.

DHTML

DHTML stands for the Dynamic Hypertext Markup Language. It is a collection of technologies used together to create interactive and animated web sites by using a combination of a static markup language (such as HTML), a client-side scripting language (such as JavaScript), a presentation definition language (Cascading Style Sheets, CSS), and the Document Object Model. [3]

A DHTML webpage is any webpage in which client-side scripting changes variables of the presentation definition language, which in turn affects the look and function of otherwise "static" HTML page content, after the page has been fully loaded and during the viewing process. Thus the dynamic characteristic of DHTML is the way it functions while a page is viewed, not in its ability to generate a unique page with each page load.

DHTML technology is used in SSOS Phase II to facilitate users in some advanced view options, such as user defined printable view shown in Figure 9. With the help of DHTML technology, production group can select part of the orders in one production and print them out.


SSOS-Smart Sign Ordering System - Windows Internet Explorer
http://localhost:8080/ssos/producing/designscreen.jsp?message=Design&productionNbr=2
Live Search
SSOS-Smart Sign Ordering System
Page
Tools
Welcome: producing you are in Producing | Help | Logout
View Productions | Email Notification |
Printable View
javascript:PopPrintableView('/ssos/producing/designscreen.jsp?message=Design&production
Internet | Protected Mode: On
100%

Design

Production #	2	Planning Date	09/26/2007	Finishing Date	-	Status	Open
Type	CBH(FS)	Priority	Normal	Total Items	3	Total Qty	6
Notes							

Complete	Order Number	Order Date	Sign Code	Legend	Size	Background Color	Applied Color	Substrate Type	Qty
☐	02-00004-08	09/26/2007	W1-1aR-18	COMBINATION RIGHT TURN / (speed advisory) XX (for bikeway)	18.0X18.0	YEL	BLK	FS	3
☐	01-00004-08	09/26/2007	W-10-18	LEFT WINDING ROAD (for bikeway)	18.0X18.0	YEL	BLK	FS	1
☐	01-00003-08	09/26/2007	W-9-18	RIGHT WINDING ROAD (for bikeway)	18.0X18.0	YEL	BLK	FS	2
				Total				6	

[Previous][Next]
[Show 5 Orders Per Page]
Sort Orders

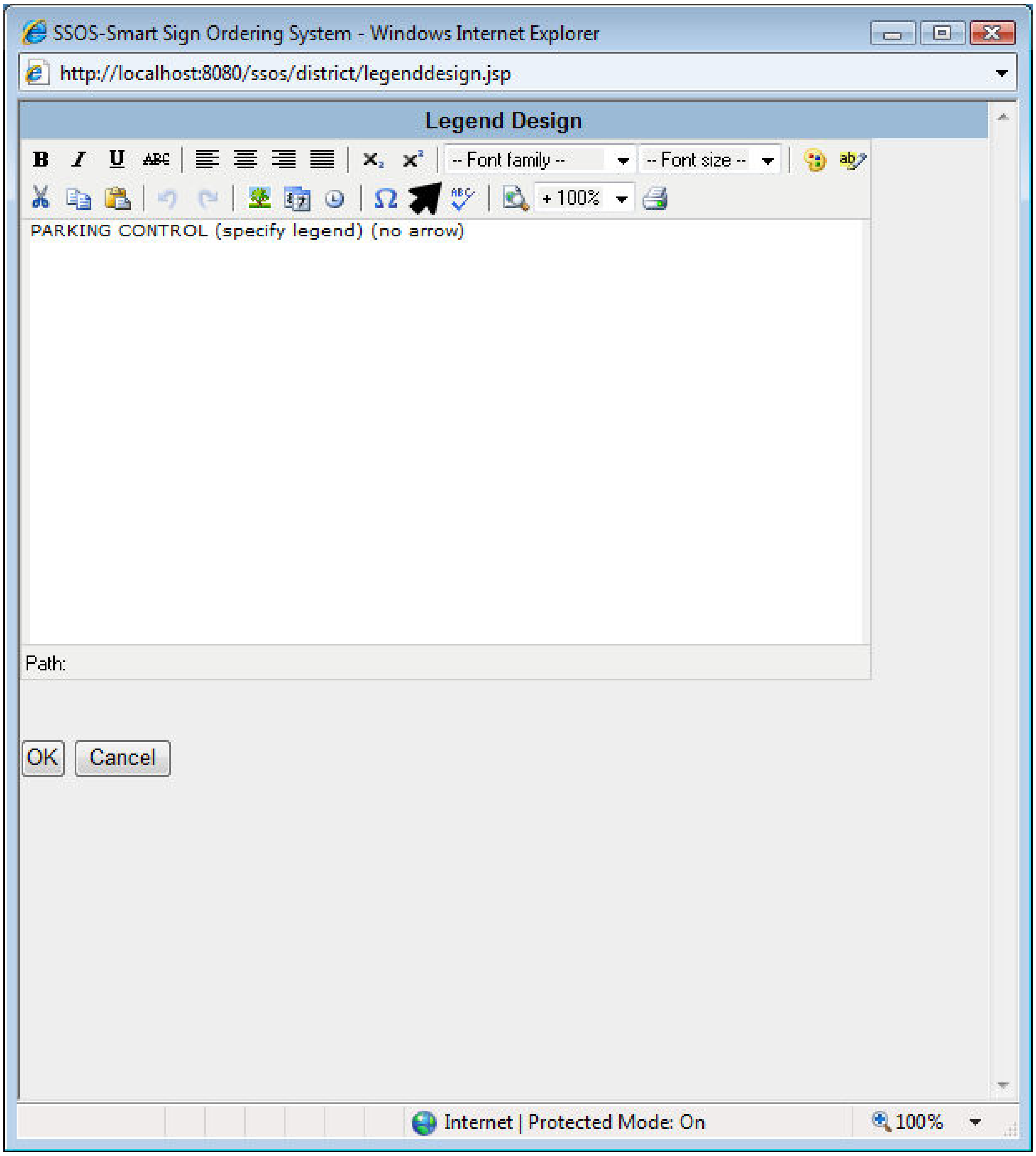
Figure 9 Application of DHTML in Customized Printable View

TinyMCE Editor

TinyMCE [4] is a platform independent web based Javascript HTML WYSIWYG editor control released as Open Source under LGPL by Moxiecode Systems AB. It has the ability to convert HTML TEXTAREA fields or other HTML elements to editor instances. TinyMCE is very easy to integrate into other Content Management Systems. It has the following features:

1. Easy to integrate, takes only two lines of code.
2. Customizable through themes and plugins.
3. Customizable XHTML 1.0 output. Block invalid elements and force attributes.
4. International language support (Language packs)
5. Multiple browser support, Mozilla, MSIE, FireFox, Opera and Safari (experimental).
6. PHP/.NET/JSP/Coldfusion GZip compressor, Makes TinyMCE 75% smaller and a lot faster to load.
7. You can easily use AJAX to save and load content!

TinyMCE is used in SSOS Phase II to enhance the customized legend function and provide a WYSIWYG editor. The screen of the TinyMCE editor in SSOS is shown in Figure 10.


SSOS-Smart Sign Ordering System - Windows Internet Explorer
http://localhost:8080/ssos/district/legenddesign.jsp
Legend Design
-- Font family --
-- Font size --
+100%
Path:
OK
Cancel
Internet | Protected Mode: On
100%

PARKING CONTROL (specify legend) (no arrow)

Figure 10 TinyMCE Editor

Chapter V Implementation and Code Explanation

Database Implementation

The database is implemented on Sybase database system, version 12.5. This DB server supports JDBC IV connection and standard SQL query. The table property and field names are listed below and they can be found in the file "SSOS Table Definition.pdf" on the SSOS CD under root directory. Scripts for setting up SSOS database tables can also be found in the same directory.

1. CreateTables.sql script will create all the necessary tables for SSOS.
2. InitialData.sql script will insert the initial data to the tables.
3. ems.sql script will insert the sign information which has been exported from Jimani. (This script will take longer to run than the others)

DISTRICT_COUNTIES

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
District Number*	DISTRICT_NBR	smallint	District Number
County Abbreviation Code	COUNTY_ABREV3_CD	char(3)	3-characters abbreviation associated with the district number

SSOS_AGENCIES

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Agency Number*	AGENCY_ID	varchar(15)	Unique number assigned to each Agency in SSOS
Agency Text	AGENCY_TXT	varchar(63)	Agency Full Name
Default Indicator	DEFAULT_IND	varchar(15)	Indicates whether this is a default record
Sequence Number	SEQ_NBR	numeric(2,0)	Sequence number used to order the records in the table

SSOS_APPLIED_COPYS

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Applied Copy ID	APPLIED_COPY_ID	varchar(15)	Unique abbreviation assigned to each Applied Copy in SSOS
Applied Copy Text	APPLIED_COPYS_TXT	varchar(63)	Applied Copy Full Name
Cost of Applied Copy	COST_NBR	numeric(8,2)	Cost of each Applied Copy
Default Indicator	DEFAULT_IND	varchar(15)	Indicates whether this is a default record
Sequence Number	SEQ_NBR	numeric(2,0)	Sequence number used to order the records in the table

SSOS_BYPASS

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Bypass Number*	BYPASS_NBR	numeric(4,0)	Unique number assigned to each Bypass definition
SQL Statement	SQL_STATEMENT_TXT	varchar(255)	SQL Statement which select the signs which bypass the district
Description	DESCRIPTION_TXT	varchar(255)	Description of the bypass definition

SSOS_CADS

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
CAD Number*	CAD_NBR	numeric(6,0)	Unique number assigned to each CAD
CAD File Name	CAD_FILE_NAME_TXT	varchar(31)	File name of the cad, not including the path name
CAD Description	CAD_DESCRIPTION_TXT	varchar(255)	Description of the CAD

SSOS_EMAILS

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Ssos Group*	SSOS_GROUP	varchar(31)	User Group
District Number	DISTRICT_NBR	smallint	District Number
Email	EMAIL_TXT	varchar(255)	Email addresses where the notification will be sent to
Deleted	DELETED_IND	varchar(15)	Email will be sent when order status become Deleted
Pending	PENDING_IND	varchar(15)	Email will be sent when order status become Pending
Approved	APPROVED_IND	varchar(15)	Email will be sent when order status become Approved
Archived	ARCHIVED_IND	varchar(15)	Email will be sent when order status become Archived
Received	RECEIVED_IND	varchar(15)	Email will be sent when order status become Received

Completed	COMPLETED_IND	varchar(15)	Email will be sent when order status become Completed
Delivered	DELIVERED_IND	varchar(15)	Email will be sent when order status become Delivered
Submitted	SUBMITTED_IND	varchar(15)	Email will be sent when order status become Submitted
In Production	IN_PRODUCTION_IND	varchar(15)	Email will be sent when order status become In Production

SSOS_EMS

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
EMS Number*	EMS_NUM	varchar(15)	EMS number of the sign
Ohio Sign Code	OHIO_SIGN_CD	varchar(31)	Ohio Sign Code of the sign
Legend Text	LEGEND_TEXT_TXT	varchar(255)	Legend on the sign (text only) no HTLM tag
Sign Width	SIGN_WIDTH_NBR	decimal(8,2)	Sign Width
Sign Height Number	SIGN_HEIGHT_NBR	decimal(8,2)	Sign Height
Square Foot	SQ_FT_NBR	decimal(8,2))	Square Foot
Sign Cost	SIGN_COST_NBR	decimal(8,2)	Sign Cost
Notes	NOTES_TXT	varchar(255)	Notes on the sign
Substrate	SUBSTRATE_TYPE_ID	varchar(15)	Substrate type of the sign
Sheet	SHEETING_TYPE_ID	varchar(15)	Sheeting type of the sign
Background Color	BACKGROUND_COLOR_TXT	varchar(21)	Background color of the sign
Applied Color	APPLIED_COLOR_TXT	varchar(41)	Applied color of the sign
Is Permanent	IS_PERMANENT_IND	varchar(5)	Is Permanent indicator
Warehouse Location	WAREHOUSE_LOCATION_TXT	varchar(31)	Warehouse location of the warehouse signs
Production Method	PRODUCTION_METHOD_ID	varchar(15)	Production method of the sign
Screen Location	SCREEN_LOCATION_NBR	numeric(4,0)	Screen Location of the sign

Screen Condition	SCREEN_CONDITION_TXT	varchar(15)	Screen condition of the sign
Screen Frame Size	SCREEN_FRAME_SIZE_TXT	varchar(15)	Screen Frame Size
Sign Positive Location	SIGN_POSITIVE_LOC_TXT	varchar(5)	Sign Positive Location
Design Date	DESIGN_DT	datetime	Design Date
Specification	SPECS_TXT	varchar(5)	Specification of the sign
Open Size	OPEN_SIZE_TXT	varchar(31)	Open Size of the Sign
Sign Cost Per Square Foot	SIGN_COST_SQFT_NBR	decimal(8,2)	Sign Cost Per Square Foot
Bypass indicator	BY_PASS_IND	varchar(15)	Bypass indicator indicates whether this sign should bypass Centrol Office or not
Applied Color Number	NUM_APPLIED_COL_NUM	varchar(5)	Special Features of the sign, such as Applied Color Number
Assignment Number	ASSIGNMENT_NBR	varchar(30)	Assignment Number of the sign
Overlay Decal Positive Location	OVERLAY_DECAL_POSITIVELOC	varchar(10)	Overlay Decal Positive Location
Overlay Decal Location	OVERLAY_DECAL_LOC	varchar(30)	Overlay Decal Location
Overlay Decal Frame Size	OVERLAY_DECAL_FRAME_SIZE	varchar(50)	Overlay Decal Frame Size (width x height)
Overlay Decal Condition	OVERLAY_DECAL_CONDITION	varchar(40)	Overlay Decal Condition (GOOD)
Shape	SHAPE	varchar(10)	Shape of the sign
Federal Sign Code	FED_SIGN_CD	varchar(20)	Federal Sign Code of sign
Applied Copy	APPLIED_COPY	varchar(10)	Applied Copy of the sign
Sign Blank EMS	SBL_EMS	varchar(15)	Sign Blank EMS Number
Gauge	GAUGE	decimal(4,0)	Gauge of the sign
Radius	RADIUS	decimal(8,2)	Radius of the sign

SSOS_EMS_COST

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Cost Rule ID*	PRICE_ID	decimal(4,0)	Unique ID assigned to each cost rule
Production Method	PRODUCTION_METHOD_ID	varchar(15)	Production method of the sign
Substrate Type	SUBSTRATE_TYPE_ID	varchar(15)	Substrate type of the sign
Gauge	GAUGE	decimal(4,0)	Gauge of the sign
Sheeting Type	SHEETING_TYPE_ID	varchar(15)	Sheeting type of the sign
Background Color	BACKGROUND_COLOR_TXT	varchar(21)	Background color of the sign
Applied Copy	APPLIED_COPY	varchar(10)	Applied Copy of the sign
Applied Color	APPLIED_COLOR_TXT	varchar(41)	Applied Color of the sign
Special Feature	SPECIAL_COLUMN	varchar(5)	Special Feature of the sign
Cost	COST	decimal(8,0)	Cost of the rule
Unit	UNIT	varchar(4)	Unit of the sign

SSOS_IMAGES

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Image Number*	IMAGE_NBR	numeric(6)	Unique number assigned to each image
Image File Name	IMAGE_FILE_NAME_TXT	varchar(31)	File name of the image, not including the path name
Image Description	IMAGE_DESCRITION_TXT	varchar(255)	Description of the image

SSOS_LEGENDS

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Legend Number*	LEGEND_NBR	numeric(6)	Unique number assigned to each legend
HTML Legend	HTML_LEGEND_TXT	text	Legend description including HTML tags and code

SSOS_MATERIAL_TYPES

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Material Type ID*	MATERIAL_TYPE_ID	varchar(15)	Unique ID assigned to each Material Type in SSOS
Material Type Text	MATERIAL_TYPE_TXT	varchar(63)	Material Type
Default Indicator	DEFAULT_IND	varchar(15)	Indicates whether this is a default record
Sequence Number	SEQ_NBR	numeric(2,0)	Sequence number used to order the records in the table

SSOS_NEEDED_DATE

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Needed Date Rule ID*	NEEDED_DATE_NBR	numeric(4,0)	Unique ID assigned to each Material Type in SSOS
SQL Statement	SQL_STATEMENT_TXT	varchar(255)	Material Type
Days Added	DATE_ADDED	numeric(4,0)	Number of days added to the order
Description	DESCRIPTION_TXT	varchar(255)	Sequence number used to order the records in the table

SSOS_ORDERS

“PRETTY” NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Order Number*	ORDER_NBR	numeric(6,0)	Unique number assigned to each order in SSOS
Order Number	ORDER_NUM	varchar(15)	Unique number assigned to each order in SSOS. This number consists of 2-digit district numbers + '-' + 5 digit sequence number per district per fiscal year + '-' + 2- digit fiscal year. E.g. 1-00049-02 is order no. 49 in year 2002 for district 1
Order Number	ORDER_DT	datetime	Time and date on the order have created
District Number	DISTRICT_NBR	smallint	District number who created the order
Order Sequence Number	ORDER_SEQ_NBR	numeric(5,0)	Order sequence per district per fiscal year
County Number	COUNTY_NUM	varchar(3)	County number for which the order belong
Comments	COMMENTS_TXT	varchar(255)	Comments on the order
Rout	ROUTE_TXT	varchar(15)	Rout number where the order belong
Needed Date	NEEDED_DT	Datetime	Date by which the order is needed
Priority	PRIORITY_STTS	varchar(10)	Priority by the order is needed
Quantity Ordered	QTY_ORDERED_AMT	numeric(4,0)	Number of sign that is needed by that order
Status	STATUS_STTS	varchar(31)	Current status of the order
Created By	CREATED_BY_TXT	varchar(31)	User name who created the order
EMS Number	EMS_NBR	varchar(15)	EMS number of the sign that is ordered
Ohio Sign Code	OHIO_SIGN_CD	varchar(31)	Ohio Sign code of the sign that is ordered
Legend Number	LEGEND_NBR	numeric(6,0)	Reference to the legend table
Legend Text	LEGEND_TEXT_TXT	varchar(255)	Text only (no image) description of the legend
Sign Width	SIGN_WIDTH_NBR	decimal(8,2)	Sign Width

Sing Height	SIGN_HEIGHT_NBR	decimal(8,2)	Sign Height
Sign Foot	SQ_FT_NBR	decimal(8,2)	Sign Size in Square Foot
Sign Cost	SIGN_COST_NBR	decimal(8,2)	Sign Cost
Substrate	SUBSTRATE_TYPE_ID	varchar(15)	Substrate type of the sign
Sheet	SHEETING_TYPE_ID	varchar(15)	Sheeting type of the sign
Production Method	PRODUCTION_METHOD_TXT	varchar(15)	Sign Production Method
Material Type	MATERIAL_TYPE_ID	varchar(15)	Sign Material Type
Applied Color	APPLIED_COLOR_TXT	varchar(41)	Sign Applied color
Background Color	BACKGROUND_COLOR_TXT	varchar(21)	Sign Background color
Order Type	ORDER_TYPE_ID	varchar(15)	Type of the order like Standard, Special, Warehouse, etc.
Agency	AGENCY_ID	varchar(15)	Name of the agency that is associated with order
Total Cost	TOTAL_COST_NBR	decimal(8,2)	Total cost of the order usually it is Individual Cost X Number of Signs
Systematic Replacement	SYS_REPALCE_IND	varchar(15)	Determine whether the order is systematic replacement or not
Image Number	IMAGE_NBR	numeric(6,0)	Reference to the image table
CAD Number	CAD_NBR	numeric(6,0)	Reference to the Cad table
Silk Screen Complete	SS_COMP	varchar(16)	Indicate Silk Screen process is completed or not
Stencil Complete	ST_COMP	varchar(16)	Indicate Stencil process is completed or not
Finishing Complete	FI_COMP	varchar(16)	Indicate Finishing process is completed or not
Design Complete	SBL_COMP	varchar(16)	Indicate Design process is completed or not
Applied Copy	APPLIED_COPY_ID	varchar(15)	Applied Copy of the order
Layout Complete	LAYOUT_COMP	varchar(16)	Indicate Layout process is completed or not
Shop Complete	SHOP_COMP	varchar(16)	Indicate Shop process is completed or not

SSOS_ORDER_HISTORIES

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Order History Number*	ORDER_HISTORY_NBR	numeric(8,0)	Unique number assigned to each record in the History table
Order Number	ORDER_NBR	numeric(6,0)	Reference number to the SSOS_ORDERS table that refer to the ORDER that this history record belong to
Update Time Date	UPDATE_TIME_DT	datetime	Time and date on which the History record added
User	USER_TXT	varchar(31)	User name of the user who took the action for this history record
Action	ACTION_TXT	varchar(31)	Action that took place for this history record
Notes	NOTES_TXT	varchar(255)	Extra notes to elaborate on the action that took place when this record added

SSOS_ORDER_TYPES

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Order Type Number*	ORDER_TYPE_ID	varchar(15)	Unique ID assigned to each Order Type in SSOS
Order Type Text	ORDER_TYPE_TXT	varchar(63)	Order Type
Default Indicator	DEFAULT_IND	varchar(15)	Indicates whether this is a default record
Sequence Number	SEQ_NBR	numeric(2,0)	Sequence number used to order the records in the table

SSOS_PACKAGES

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Package Number*	PACKAGE_NBR	numeric(6,0)	Unique number assigned to each Package in SSOS
Package Number	PACKAGE_NUM	varchar(15)	Unique number assigned to each package in SSOS. This number consists of T + 2-digit district numbers + '-' + 4 digit sequence number

			per district per fiscal year + '-' + 2- digit fiscal year. E.g. T04-0012-02 is package no. 12 in year 2002 for district 4
Package Sequence Number	PACKAGE_SEQ_NBR	numeric(6,0)	Package sequence per district per fiscal year
District Number	DISTRICT_NBR	smallint	District number where the package was packed to
Notes	NOTES_TXT	varchar(255)	Notes on the package
Created By	CREATED_BY_TXT	varchar(31)	User name of the user who created the package
Status	STATUS_STTS	varchar(31)	Status of the package
Packed Date	PACKED_DT	datetime	Time and date on which the package was created
Transfer Date	TRANSFER_DT	datetime	Time and date on which the package was transferred

SSOS_PACKAGE_ITEMS

“PRETTY” NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Package Item Number*	PACKAGE_ITEM_NBR	numeric(6,0)	Unique number assigned to each Package Item in SSOS
Package Number	PACKAGE_NBR	numeric(6,0)	Reference number to Package table
Order Number	ORDER_NBR	numeric(6,0)	Reference number to Order Table
Quantity Packed	QTY_PACKED_AMT	numeric(4,0)	Number of signs in the package item

SSOS_PRODUCTIONS

“PRETTY” NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Production Number*	PRODUCTION_NBR	numeric(6,0)	Unique number assigned to each Production in SSOS
Planning Date	PLANNING_DT	datetime	Time and date on which the production was created
Notes	NOTES_TXT	varchar(255)	Note on the production
Created By	CREATED_BY_TXT	varchar(31)	User name of the user who created the production

Status	STATUS_STTS	varchar(15)	Status of the production
Priority	PRIORITY_STTS	varchar(31)	Priority of the production
Type	TYPE_STTS	varchar(31)	Type of the production e.g. Silk Screen, Copy By Hand, etc…
Finishing Date	FINISHING_DT	datetime	Time and date when the production status changed to finished

SSOS_PRODUCTION_ITEMS

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Production Item Number*	PRODUCTION_ITEM_NBR	numeric(6,0)	Unique number assigned to each Production Item in SSOS
Production Number	PRODUCTION_NBR	numeric(6,0)	Reference to production table
Order Number	ORDER_NBR	numeric(6,0)	Reference to order table
Production Quantity	PRODCUTION_QTY_AMT	numeric(4,0)	Number of signs in the production item

SSOS_PRODUCTION_METHODS

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Production Method ID*	PRODUCTION_METHOD_ID	varchar(15)	Unique ID assigned to each Production Method
Production Method Name	PRODUCTION_METHOD_TXT	varchar(63)	Production Method Name
Cost	COST_NBR	decimal(8,2)	Cost of this production method
Default Indicator	DEFAULT_IND	varchar(15)	Indicates whether this is a default record
Sequence Number	SEQ_NBR	numeric(2,0)	Sequence number used to order the records in the table

SSOS_QUERY_FIELDS

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Field Name*	FIELD_NAME_TXT	varchar(31)	Unique number assigned to each Field Name in SSOS
Column Name	COLUMN_NAME_TXT	varchar(31)	Name of the column that has the look up values
Data Type	DATA_TYPE_TXT	varchar(31)	Data type of the field
Table Name	TABLE_NAME_TXT	varchar(31)	The table that holds the look up values
Table Abbreviation	TABLE_ABBREVIATION_TXT	varchar(5)	Abbreviation of the table name

SSOS_QUERY_TABLES

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Query Number*	QUERY_NBR	numeric(4,0)	Unique number assigned to each Query in SSOS
Query Name	QUERY_NAME_TXT	varchar(31)	Query name
Created By	CREATED_BY_TXT	varchar(31)	User Name of the user who created the query
SQL Statement	SQL_STATEMENT_TXT	varchar(511)	SQL statement that define the query
Creation Date	CREATION_DT	datetime	Time and date when the query was created

SSOS_SHEETING_TYPES

"PRETTY" NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Sheeting Type ID*	SHEETING_TYPE_ID	varchar(15)	Unique ID assigned to each Sheeting Type in SSOS
Sheeting Type Text	SHEETING_TYPE_TXT	varchar(63)	Sheeting Type
Cost	COST_NBR	decimal(8,2)	Cost
Default Indicator	DEFAULT_IND	varchar(15)	Indicates whether this is a default record

Sequence Number	SEQ_NBR	numeric(2,0)	Sequence number used to order the records in the table

SSOS_SIGN_PRICE

“PRETTY” NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Sheeting Type ID*	SHEETING_TYPE_ID	varchar(15)	Unique ID assigned to each Sheeting Type in SSOS
Sheeting Type Text	SHEETING_TYPE_TXT	varchar(63)	Sheeting Type
Cost	COST_NBR	decimal(8,2)	Cost
Default Indicator	DEFAULT_IND	varchar(15)	Indicates whether this is a default record
Sequence Number	SEQ_NBR	numeric(2,0)	Sequence number used to order the records in the table

SSOS_SUBSTRATE_TYPES

“PRETTY” NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Substrate Type Number*	SUBSTRATE_TYPE_ID	varchar(15)	Unique ID assigned to each Substrate Type in SSOS
Substrate Type Text	SUBSTRATE_TYPE_TXT	varchar(63)	Substrate Type
Cost	COST_NBR	decimal(8,2)	Cost
Default Indicator	DEFAULT_IND	varchar(15)	Indicates whether this is a default record
Sequence Number	SEQ_NBR	numeric(2,0)	Sequence number used to order the records in the table

Table 2   Database Table

User Interface Implementation

User interfaces of SSOS Phase II are implemented in JSP with DHTML support for dynamic user interaction. As introduced in Chapter IV, JSP is a server side program that will be compiled into a Java class when called from a client. Different from CGI and ASP, JSP will be compiled once at the first access and will be loaded into the Java virtual machine for future referencing. This feature largely reduces the turnaround time when multiple accesses happen. JSP files in SSOS can be found in the SSOS CD under the "ssos" directory. The purpose of each is described in the following table.

Root JSP Files

File	Category	Summary
"authenticate.jsp"	root	Authentication based on database stored users (for testing purpose only)
"dologin.jsp"	root	Authentication based on OneLogin security system (for final deployment)
"login.jsp"	root	Welcome page for entering "username" and "password"
"logout.jsp"	root	Logout the user and close the session

Central Module JSP Files

File	Category	Summary
"adddesign.jsp"	Central	Add new legend designs with HTML tags and pictures
"addhistory.jsp"	Central	Add new customized records in the order history table
"addhtml.jsp"	Central	Add HTML designs to a legend design
"addtext.jsp"	Central	Add text designs to a legend design
"bypass.jsp"	Central	List all bypass conditions
"changeall.jsp"	Central	Change the status of a group of orders
"changestatus.jsp"	Central	Change the status of a single order
"createbypass.jsp"	Central	Add new bypass conditions
"createneededdate.jsp"	Central	Create needed date rules
“createquery.jsp”	Central	Add new query definitions to the database

“deletebypass.jsp”	Central	Delete bypass conditions
"deltectneededdate.jsp”	Central	Delete needed date rule
"deletequery.jsp"	Central	Delete a saved query from the database
"doaction.jsp"	Central	Actions (add line, add image, change alignment) for a customized HTML legend
"executebypass.jsp"	Central	Run bypass conditions
"legenddesign.jsp"	Central	View HTML legend designs
"moveline.jsp"	Central	Change the alignment of a line
"neededdate.jsp"	Central	View needed date rules
"nestedorder.jsp"	Central	Order the orders in the vieworder.jsp page based on multi columns
"queryordernumber.jsp"	Central	Lookup an order based on its order number
"savebypass.jsp"	Central	Save bypass conditions in the database
"saveneededdate.jsp"	Central	Save needed date rule in the database
"savequery.jsp"	Central	Save query definitions in the database
"updateorder.jsp"	Central	Apply order modifications in the database
"updateprice.jsp"	Central	Update price of the order
"uploadcad.jsp"	Central	Upload CAD files to the server
"uploadimage.jsp"	Central	Upload a sign image to the server
"viewcad.jsp"	Central	View an attached CAD file
"viewhistory.jsp"	Central	View the history of an order
"viewimage.jsp"	Central	View an attached sign image
"viewmodify.jsp"	Central	View order details in the modify mode
"vieworders.jsp"	Central	View summery of a list of orders
"viewqueries.jsp"	Central	View a list of saved queries

Datamanager – Data Manager Module Files

File	Category	Summary
"addems.jsp"	Datamanager	Add a new sign to database
"addemscost.jsp"	Datamanager	Add a new component of sign cost to database
"deleteems.jsp"	Datamanager	Delete a sign
"deleteemscost.jsp"	Datamanager	Delete a component of sign cost

"ems.jsp"	Datamanager	Search sign by EMS number
"legend.jsp"	Datamanager	Search sign by legend
"newems.jsp"	Datamanager	Create a new sign
"newemscost.jsp"	Datamanager	Create a new component of sign cost
"signcode.jsp"	Datamanager	Search sign by sign code
"updateems.jsp"	Datamanager	Update sign information in the database
"updateemscost.jsp"	Datamanager	Update sign cost component information in the database
"uploadimage.jsp"	Datamanager	Upload image for a sign
"viewemscost.jsp"	Datamanager	View sign cost components
"viewemss.jsp"	Datamanager	View signs
"viewimage.jsp"	Datamanager	View sign image
"viewmodify.jsp"	Datamanager	Modify sign information
"viewmodifycost.jsp"	Datamanager	Modify sign cost component information

District – District Module Files

File	Category	Summary
"adddesign.jsp"	District	Add new legend design with HTML tags and pictures
"addhistory.jsp"	District	Add new customized record in the order history table
"addhtml.jsp"	District	Add HTML design to the overall legend design
"addtext.jsp"	District	Add text design to the overall legend design
"changeall.jsp"	District	Change the status of all checked orders
"changestatus.jsp"	District	Change the status of a single order
"createquery.jsp"	District	Create customized query and save it in the database
"deletequery.jsp"	District	Delete saved query from the database
"design.jsp"	District	List sign attributes ready to make order
"doaction.jsp"	District	Actions (add line, add image, change alignment) for customized HTML legend
"ems.jsp"	District	Search for signs by typing few digits of the EMS number
"legend.jsp"	District	Search for signs in the database by providing few letters of the legend
"legenddesign.jsp"	District	View the HTML legend design
"makeorder.jsp"	District	Make an order and add it to the database

"moveline.jsp"	District	Change the alignment of a line
"nestedorder.jsp"	District	Order orders in the vieworder.jsp page based on multi columns
"queryordernumber.jsp"	District	Look up an order by giving its order number
"savequery.jsp"	District	Save a query in the database
"signcode.jsp"	District	Search for signs by typing few digits of the Sign Code
"updateorder.jsp"	District	Apply order modifications in the database
"updateprice.jsp"	District	Update the price of the order
"uploadcad.jsp"	District	Upload a CAD file to the server
"uploadimage.jsp"	District	Upload a sign image to the server
"viewcad.jsp"	District	View an attached CAD file
"viewhistory.jsp"	District	View the history of an order
"viewimage.jsp"	District	View an attached sign image
"viewmodify.jsp"	District	View order details in the modify mode
"vieworders.jsp"	District	View summery of a list of orders
"viewqueries.jsp"	District	View a list of saved queries

Packing Module JSP Files

File	Category	Summary
"addhistory.jsp"	Packing	Add new legend designs with HTML tags and pictures
"createquery.jsp"	Packing	Add new query definitions to the database
"deletequery.jsp"	Packing	Delete saved queries from the database
"doinsert.jsp"	Packing	Insert an order into existing package
"insertpackage.jsp"	Packing	Review insert an order into existing package
"makepackage.jsp"	Packing	Save a new package in the database
"mvpackage.jsp"	Packing	Confirm the creation of a package before saving it in the database
"nestedorder.jsp"	Packing	Order the orders in the vieworder.jsp page based on multi columns
"queryordernumber.jsp"	Packing	Lookup an order based on its order number
"querypacking.jsp"	Packing	Search for packages using certain parameters
"savequery.jsp"	Packing	Save a query definition in the database
"viewems.jsp"	Package	View the sign information
"viewhistory.jsp"	Packing	View the history of an order
"viewmodify.jsp"	Packing	View order details in the modify mode

"vieworders.jsp"	Packing	View summery of a list of orders
"viewpackages.jsp"	Packing	View summery of a list of packages
"viewqueries.jsp"	Packing	View a list of saved queries

Planning Module JSP Files

File	Category	Summary
"adddesign.jsp"	Planning	Add new legend designs with HTML tags and pictures
"addhistory.jsp"	Planning	Add new customized records in the order history table
"addhtml.jsp"	Planning	Add HTML designs to the overall legend design
"addtext.jsp"	Planning	Add text designs to the overall legend design
"changestatus.jsp"	Planning	Change the status of an order
"createquery.jsp"	Planning	Add new query definitions to the database
"deletequery.jsp"	Planning	Delete saved queries from the database
"designscreen.jsp"	Planning	View the printout screens for production planning
"doaction.jsp"	Planning	Actions (add line, add image, change alignment) for the customized HTML legend
"doinsert.jsp"	Planning	Insert an order into existing production
"insertproduction.jsp"	Planning	Review the insert of an order into existing production
"finishingscreen.jsp"	Planning	View the printout screens for finishing production
"legenddesign.jsp"	Planning	View an HTML legend design
"makeproduction.jsp"	Planning	Save a production in the database
"moveline.jsp"	Planning	Change the alignment of a line
"mvproduction.jsp"	Planning	Confirmation message before creating a production
"nestedorder.jsp"	Planning	Order the orders in the vieworder.jsp page based on multi columns
"queryordernumber.jsp"	Planning	Lookup an order based on its order number
"savequery.jsp"	Planning	Save a query definition in the database
"undoapprove.jsp"	Planning	Undo approve order operation
"updateorder.jsp"	Planning	Apply order modifications in the database
"uploadcad.jsp"	Planning	Upload a CAD file to the server
"uploadimage.jsp"	Planning	Upload a sign image to the server
"viewcad.jsp"	Planning	View an attached CAD file
"viewhistory.jsp"	Planning	View the history of an order

"viewimage.jsp"	Planning	View an attached sign image
"viewmodify.jsp"	Planning	View order details in the modify mode
"vieworders.jsp"	Planning	View summery of a list of orders
"viewqueries.jsp"	Planning	View a list of saved queries

Producing Module JSP Files

File	Category	Summary
"addhistory.jsp"	Producing	Add new customized records in the order history table
"changestatus.jsp"	Producing	Change the status of an order
"closeproduction.jsp"	Producing	Change the status of a production to "Closed"
"completedesign.jsp"	Producing	Complete the design process of an order
"completefinishingscreen.jsp"	Producing	Complete the finishing screen process of an order
"completelayout.jsp"	Producing	Complete the layout process of an order
"completeorder.jsp"	Producing	Change the order's status to "Completed"
"completeshop.jsp"	Producing	Complete the shop process of an order
"completesilkscreen.jsp"	Producing	Complete the silk screen process of an order
"completestencilscreen.jsp"	Producing	Complete the stencil screen process of an order
"designscreen.jsp"	Producing	View the design process of a order
"finishingscreen.jsp"	Producing	View the finishing screen process of a order
"layout.jsp"	Producing	View the layout process of a order
"shop.jsp"	Producing	View the shop process of a order
"silkscreen.jsp"	Producing	View the silkscreen process of a order
"undodesign.jsp"	Producing	Undo design complete process of a order
"undofinishingscreen.jsp"	Producing	Undo finishing screen complete process of a order
"undolayout.jsp"	Producing	Undo layout complete process of a order
"undoorder.jsp"	Producing	Remove order from the production
"undoordercomplete.jsp"	Producing	Undo order complete process of a order
"undoshop.jsp"	Producing	Undo shop complete process of a order
"undosilkscreen.jsp"	Producing	Undo silk screen complete process of a order
"undostencilscreen.jsp"	Producing	Undo stencil screen complete process of a order
"updateorder.jsp"	Producing	Apply order modifications in the database
"uploadcad.jsp"	Producing	Upload a CAD file to the server

"uploadimage.jsp"	Producing	Upload a sign image to the server
"viewcad.jsp"	Producing	View an attached CAD file
"viewems.jsp"	Producing	View the sign information
"viewhistory.jsp"	Producing	View the history of an order
"viewimage.jsp"	Producing	View an attached sign image
"viewmodify.jsp"	Producing	View order details in the modify mode
"vieworders.jsp"	Producing	View summery of a list of orders
"viewproduction.jsp"	Producing	View a list of productions

Transfer Module JSP Files

File	Category	Summary
"addhistory.jsp"	Transfer	Add new legend designs with HTML tags and pictures
"createquery.jsp"	Transfer	Add new query definitions to the database
"deletequery.jsp"	Transfer	Delete a saved query from the database
"nestedorder.jsp"	Transfer	Order the orders in the vieworder.jsp page based on multi columns
"queryordernumber.jsp"	Transfer	Lookup an order based on its order number
"querypacking.jsp"	Transfer	Lookup a package based on its package number
"savequery.jsp"	Transfer	Save query definitions in the database
"transferorder.jsp"	Transfer	Change the status of an order to "transferred"
"transferpackage.jsp"	Transfer	Change the status of a package to "transferred"
"viewhistory.jsp"	Transfer	View the history of an order
"viewmodify.jsp"	Transfer	View order details in the modify mode
"vieworders.jsp"	Transfer	View the summery of a list of orders
"viewpackages.jsp"	Transfer	View the summery of a list of packages
"viewqueries.jsp"	Transfer	View a list of saved queries

Table 3 JSP Files List

Security Implementation

Security is a very important issue in web development. There are two commonly used approaches to address the issue: system security and application security. Currently, application security is used by SSOS based on the OneLogin security system implemented by DoIT as shown in Figure 11. Each SSOS user will have one or more groups assigned to him/her. The first group will be the default one and the others will be on the toolbar so he/she can switch between them. If a user failed to pass the OneLogin security system he/she will be returned to the login page.

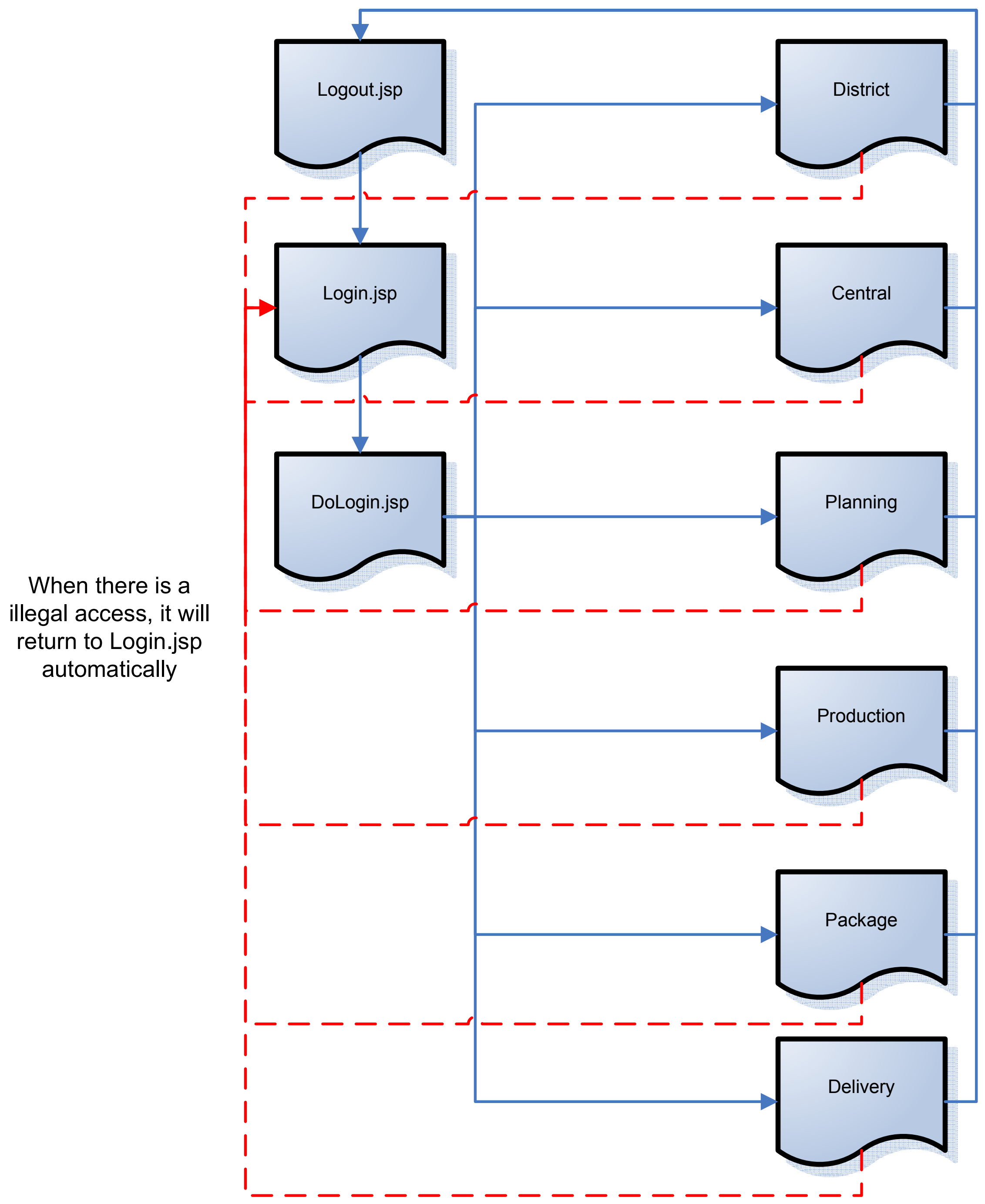


Figure 11 SSOS Security System

CONCLUDING REMARKS

The project enhances the smart sign ordering system (SSOS) to address the requirement in sign ordering, fabrication, tracking and data management. Specifically, this project has studied the production requirements in detail at the Sign Shop, replaced the existing JIMANI data program, and built up sign cost table and price calculation scheme. In addition, it has implemented the functions for order approval, production planning, operation management, package assistance and order status tracking. Furthermore, improvements over the user interface have been made and technologies such as TinyMCE editor and DHTML used to enhance easy execution of the system. The project has fulfilled all of its objectives. With the enhancement, the SSOS can be used to improve the work efficiency of the sign ordering and fabrication process, and to speed up the order review time and reduce potential human errors in order handling.

BIBLIOGRAPHY

1. Sun java tutorial (java.sun.com/docs/books/tutorial/)
2. Sun Java Servlet Introduction page (http://java.sun.com/products/servlet/)
3. Dynamic HTML – Wikipedia (http://en.wikipedia.org/wiki/DHTML)
4. TinyMCE Javascript Content Editor by Moxiecode System AB (http://tinymce.moxiecode.com/)