XML GENERATOR

Done at

LUCID TECHNOLOGIES PVT LTD

PROJECT REPORT

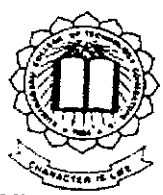
Submitted in partial fulfillment for award of Degree of M.Sc.[Applied science Software Engineering]

Done by

Y.HARIRAM YESHWANTH
9837S0047

Guided by

Mrs.S.Devaki
&
Mr.P.Saravana Kumar



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY
(Affiliated to Bharathiar University)
COIMBATORE – 641006

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to Bharathiar University)

COIMBATORE – 641006

BONAFIDE CERTIFICATE

This is to certify that this is the Bonafide Project Record Work done by **Y.HARIRAM YESHWANTH** , Reg.No **9837S0047** in Partial fulfillment for the award of Degree of MSc [APPLIED SCIENCE SOFTWARE ENGINEERING] , during the academic year 2002 –2003.

S. J... 31/3/03

Prof & HOD

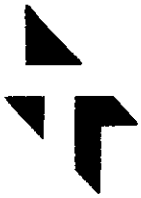
Guide

SUBMISSION

This Project entitled "XML Generator " is submitted for the X semester of MSc [APPLIED SCIENCE SOFTWARE ENGINEERING] for Bharathiar University Project Viva-voce examinations held on 4-4-03

Internal Examiner

External Examiner



Lucid Technologies Pvt. Ltd.

Microsoft
CERTIFIED

Bonafide Certificate

Certified that this thesis on "XML Generator" is the bonafide work of **Mr. Hariram Yeshwanth Y.** **Register Number:** 9837S0047 and **Roll Number**: 98SE07, who carried out the project in our organization during the period December 2002 to March 2003. Certified further that to the best of my knowledge, the work reported there-in does not form part of any other thesis or work on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Srinath Jagannathan
Director

Dedicated to my ever loving Parents and Lord almighty...

Acknowledgement

To add meaning to the perception , it is my indebtedness to honor a few who had helped me in this endeavor, by placing them on record.

With profound gratitude, I am extremely thankful to **Dr.K.K.Padmanaban B.Sc (Eng),M.Tech,Ph.D.**, Principal, Kumaraguru College of Technology, Coimbatore for providing me an opportunity to undergo the MSc[APPLIED SCIENCE SOFTWARE ENGINEERING] Course and thereby this project work also.

I extend my heartfelt thanks to my CSE department head **Prof.Dr.S.Thangasamy B.E.(Hons),Ph.D.**, for his kind guidance and encouragement to complete this project successfully.

It's my privilege to express my deep sense of gratitude and profound thanks to **Mr.Srinath Jaganathan, Managing Director**, and **Mrs.Kalyani Jaganathan,Technical Director** Lucid Technologies Pvt Ltd, Chennai for having allowed me to do my project work in their esteemed concern and for helping me in all means in successful completion of this project work.

Gratitude will find least meaning without a mention for **Asst Prof.Mrs.S.Devaki B.E., M.S.,** and **Mr.P.SaravanaKumar Software Engineer** who have guided me with their valuable suggestions and consistent Motivations during my project work. I'm happy that I was able to give shape to their novel ideas to an extent.

Words are boundless for me to express my deep sense of gratitude and profound thanks to **Mr.Meenakshi Sundaram** ,all my associates at Lucid Technologies, and all the other staff in my college for all their kind guidance and encouragement towards my project work. I've learnt quite a lot and acquired enough technical skills from their guidance without which, this project work could have not reached its successful completion.

Finally, this acknowledgement will not achieve its complete form if I don't remember my parent's sacrifices. Without their constant moral support, motivations and kind encouragements, I could have not channelised my career in the field of Computer science.

Above all, I dedicate this project to the Lord Almighty, who has been miraculously leading me throughout my life and career.

- Y.Hariram Yeshwanth

Project Abstract

This Project entitled "XML Generator" is done at Lucid Technologies Pvt Ltd, Chennai, and the main purpose of this project is to develop a product that will provide wireless service's based on the requirement of the client. The product will be customized to match the client's requirement's. This project entitled "Style Studio" is developed using Java technologies ,XML and XSLT . Oracle is used for data storing, retrieval and manipulation purposes at the backend.

The Style Studio(SS) is a XML based network application. The system facilitates a client to provide internet based services which are WAP enabled wherein he will be able to use the client's wireless services. The system contains modules for handling customer requests and generate appropriate responses based on the request. As a request from a customer is sent, it undergoes various stages of checks in the system.

There are two main module's in the application, namely XML Generator and XSLT custom file format generator. First module is meant to be at the back end wherein it will produce the appropriate XML response based on the request received. The second module will translate the XML and generate the output in the form requested by the user. The two module's operate independently of each other. Each one will be a plug-in for the other.

The system produces various types of response's as required by the user which may vary from account balance enquires to complaint's to horoscope to weather report's based on the client. The scope of the product is so wide that we will not be able to predict the type of clients for this product. But this product will mainly target the banking sector where the financial institution will provide wireless services.

Contents

INTRODUCTION

1.1 Organization Profile - LUCID TECHNOLOGIES Pvt Ltd.

Lucid Technologies was set up in 1997 with a single minded mission: To provide world-class IT services that enable clients to develop and market innovative products and services that rely on next generation technologies.... and win.

Our world class development centre in Chennai, India, is equipped with the latest technologies and solutions for enterprise networking, office productivity and collaborative software engineering. We are a Microsoft Certified Partner and a part of the Sun Developer Partner Program.

1.2 Project Overview - Style Studio

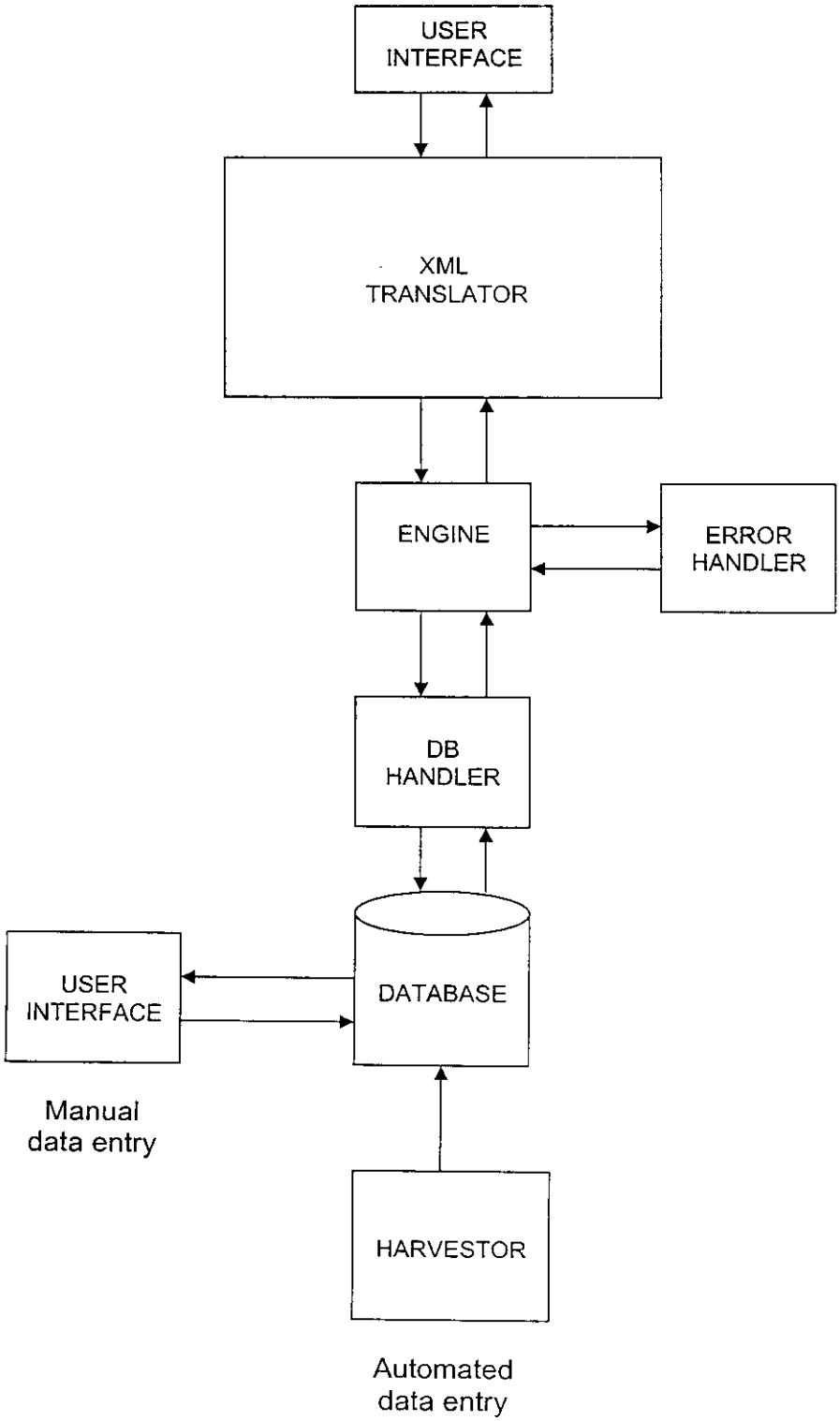
XML Generator is a part of Style Studio(SS).The Style Studio (SS) is a XML based network application. The system facilitates a client to provide internet based services which are WAP enabled wherein the customer will be able to use the client's wireless services. The system contains modules for handling customer requests and generates appropriate responses based on the request. As a request from a customer is sent, it undergoes various stages of checks in the system.

The system contains modules for handling various types of customer request's based on the client who is purchasing this product. As a request from a customer is sent, it undergoes various stages of checks in the system. There are mainly two modules namely XML Generator and XSLT custom file format generator. First module is meant to be at the back end wherein it will produce the appropriate XML response based on the request received. The second module will translate the XML and generate the output in the form requested by the user. The two module's operate independently of each other. Each one will be a plug-in for the other.

This product is unique of its kind and has not yet been attempted to the best of our knowledge. So far XML parsers and transformers have been existing as separate entities, in this product we have made an attempt to combine the features of both of these technologies to develop a product which will address far beyond the needs of either the XML parser or the transformer. The key feature being the using of wireless technologies enables the user to perform secure transactions and hence paws the way for a reliable Credit Realization and Authentication for Online Transactions.

The various important terms used in the application are : (a) Automated data population(Harvesting) - Filling the database automatically by extracting the data from the financial institution specified by the client(b) Manual data population – Filling the database manually by providing an interface which the client will use to feed data into the database (c) Connection Pooling – A technology used for maintaining the integrity of the database by restricting the no of connections at any given moment.

1.3 Business Scenario



User Sign Up

Throughout this document the sample scenario that we will be considering will be a banking application where our client will be a financial institution which will be providing wireless services to their customers .The user will first need to sign up for the Financial Institution's (FI) service. The user sign up may be through any mean's it will depend on the client. Then he will login to the FI's website then add FI's of his choice where he is maintaining accounts. He will do this by providing the username and password that he uses to access the FI's website. The website will then use a process called harvesting to extract data from the FI's website using the username and password provided by the use. The harvesting process will be described below.

User Request And Response

The customer will use the user interface of his choice to generate a request which will be first processed by the XSLT custom file format generator which will then be sent to the JAVA engine where the request will be processed to determine the nature and validity of the request. The nature of the service being that the customer will be able to view his account balances for an asset and payment due date's in case of liability in his mobile phone. The user will send a request with the keyword which will determine the nature of his request as either an account balance request or a payment due date request along with an account number which will be used to identify his account. The system will then process the user's request determine its nature and then generate an appropriate XML response and send it to the user in the requested format. If the request is wrong then the appropriate error response will be generated.

Alerts

In addition to this he will be able to set alerts for each of his account. Let us say that he sign's up for this service he will then have to login to the website and then he will have to set alerts for each of his account depending on the nature of his account. The alert may be due date alert or a balance alert. When he set's a due date alert he will be sent a alert to his mobile phone when his due

date approaches. He will also be able to set an alert if his account balance goes above or below a certain amount.

Manual Data Entry

This module is for the client where the client will be provided with a user interface to populate the database in cases where automated data population is not possible. In a different scenario where this product is customized to client which provides service's like weather report we will need to provide this module as a means for database entry.

Automated Data Entry

This is where the harvesting process is used. Here we will use the username password provided by the user to simulate user login to the bank's website and go to the summary page and we will scrape the values present in the page and we will use populate these values into the database. Then on request from the user the values will be sent to him.

SYSTEM ANALYSIS

2.1 Existing System

To the best of our knowledge currently none of the Institutions are providing this sort of service. This will be an innovation wherein he will not always need to rely on a computer or his bank to provide him with his account balances. In the current wireless world he will need to see account balances from his mobile phone. This system will enable such service's to the client.

2.2 Need for this new System

Consider this simple scenario the customer will need to view the account balance when he wants to write out a check to see if he has the required balance. As of now he should either approach the bank or login to the website to view his account balances but with this new system he can request for his balances from his mobile phone.

1. As the application will be handling many Customers , the application's security has to be maintained. Hence Java version is preferred (strong in server side programming).
2. The proposed version is very much scalable.
3. In the wireless world each institution providing wireless services will soon become a necessity rather than a luxury.
4. The Oracle 8.0 version supports more new functionalities, along with Java's advantages .
5. He will be able to have all his account balance through a single window.
6. Can be accessible to any system with a compatible mobile phone.
7. Shifting this system to different domains is very easy.
8. Addition of new functionality is very easy as we need to add the functionality only in the server.
9. Setting the reminders using his alerts option is very useful.

2.3 Proposed System - Java Version

The proposed system would address all the above listed features with main emphasis on Security and Content Customization. The main system can be split into many individual modules each of which can be considered as plug-in modules for the other modules.The main system can be split into

many individual modules each of which can be considered as plug-in modules for the other modules. The main core forms the XML Generator and XSLT custom file format generator modules These are the two modules that handle the user requests all the other modules form a support for these two modules and are built around these module's. When the user request's for an enquiry he will receive a response after sometime. He will not know what are the actions taking place in between the request and response action.

The manual data population module start's with the crucial authentication process where the client's data entry operator's will be authenticated .Then they will populate the database using the UI provided for them .They will make changes to already added entries.

All the other modules like the user sign up is all provided by the institution and he will customize his account to enable this services. The harvester module is an automated module that will be triggered on two modes. The two modes are the on-demand mode and daily mode. In on-demand mode his account details for other FI's will be updated on clicking update from the site. On daily mode his account details will be updated on a daily basis at a specific time each day. These modes are set in the FI's site for each account he can set the update mode. Based on these modes the harvester module will be triggered.

DETAIL DESCRIPTION

The system will incorporate two main modules and many sub-modules they are.

- XML Generator
    - DB Handler
    - JAVA Engine
    - Error Handler
    - Manual data population
    - Automatic data population(Harvester)
- XSLT custom file format generator

The two modules will function independently of each other and will be plug-in modules for each other. The other sub modules will support these main modules and some of them like the harvester and the manual data population module will be plug-ins.

XML Generator

This module lies in the backend and is mainly responsible for all database related activities like entry, retrieval and maintenance. This module is made up of many sub modules that will be used for various operations. The main function of this module is to get the request and process it to check its validity. Then based on the request the database will be queried or the error handler will be called a response either a normal response or a error response which will be in an XML Format will be generated which will then be sent to the XSLT custom file format generator.

Java Engine

This forms the core of the XML Generator this is the module which will interact with all the other modules other than the data population modules. This will first get the request then it will submit the request to the error handler which will check the validity of the request and authenticate the request. Then the request will again be processed by the engine which will then process the nature of the request and call appropriate DB Handlers to get the required information. Then this information extracted will then be used to generate an appropriate response which will be an XML which will then be sent to the XSLT custom file format generator.

DB Handler

This module will be called by the engine in case of a database operation this module is mainly composed of beans which are used for database queries. The engine will call the appropriate DB Handler based on the nature of the request .

Error Handler

This module is used for error handling and request authentication. Three main cases may occur the first is that the database or the server may fail. In the second case the user may send an invalid

request. In the third case an invalid user may send a request. Such errors are to be handies and appropriate error responses are to be generated as XML's and the response is to be sent to the user.

Manual Data Population

For all this requests the responses are generated by extracting the data from database. The data in the database are to be constantly updated. For this we provide two means one is the manual data population where we provide a user interface for the client to add ,edit and delete the database entries .For this we need to provide security where unauthorized database access will be restricted.

Harvester

This module is for automated data population. This is specifically for all banking sector clients where the account details will be extracted from the specific bank websites. This is an automated process which will simulate a user login into his financial website and scrape the data from the html presented there and place those account details in the database.

XSLT custom file format generator

This is the core system which takes care of all the transformation process it receives request from the client in form of HTTP requests if the request is from a web browser or WAP request if its from a WAP Browser or a mobile device , send the request to the XML Transformer. The XSL Transformer generates a XML file based on the request or generates an error code in an XML file. This file is then converted into the clients target format and send to the client. The description and functionality of the individual modules are listed below :

XSLT Engine :

This module contains the XSLT Transformer Engine which uses XSLT to structure data and extract data from the XML file. It take as input the generated XML file and expels as output an XSL file which then includes a stylesheet for data structuring.

HTML Transformer :

This module contains the code which transforms the XML document into an HTML document. Before the transformation is done the target request is identified and the target device is identified. This module then pulls in the stylesheet which is specific to the requirement and structures data based on the type of request. If the request is from Internet Explorer a different stylesheet is pulled in and if its from a Netscape Navigator then a different stylesheet is pulled in.

Stylesheet Repository :

This module is a virtual repository of stylesheets it conatins within it all the types of stylesheets in variations for various type of browsers , various types of mobile devices and hand held devices. When a transformation is needed stylesheets are pulled out of this repository. Almost all the stylesheets within this module are static entities and this contains two general templates which are called when the device or the browser type is unknown or new.

WML Transformer :

This module is similar to that of the HTML Transformer except for the fact that this module is responsible for handling WAP requests. Before the transformation is done the target request is identified and the target device is identified. This module then pulls in the stylesheet from the repository and structures data based on the type of device. If the request is from a Nokia device it gets the Model number and based on the model number and the manufacturer specification a stylesheet is pulled in from the repository . Supposing if the device is a Motorola or Siemens made then the respective model number is identified from the request and the corresponding stylesheet is pulled in.

Content Customization Module :

The main functionality of this module is to retrieve customized content for particular type of user. BizCodes are used to identify the financial institution and by using the BizCodes the data is structured to suit that particular client. Say if an associate of the main financial client has restricted access

only to view the name of an customer then this property is identified using the BizCodes and the data is structured in a way that it conatins only the name of the custome and not any other information.

SYSTEM ENVIRONMENT

3.1 Computing Environment

Server:

Web Server Software	Iplanet 4.1
Application Server Software	Weblogic 5.1
Server Operating System	Sun Solaris 2.8
Database Software	Oracle 8 RDBMS
Hardware	Sun SPARC Server ES 3000
Web Security Software	Web Security Management System

Client (Intranet):

Hardware	IBM PC Compatible Pentium III or later
Client Operating System	Windows 9x, Windows 2000, Windows NT Workstation
Web Browser	Internet Explorer 5.0, Netscape Communicator 4.6 or later

Development:

Hardware	IBM PC Compatible Pentium III or later
Client Operating System	Windows 9x, Windows 2000, Windows NT Workstation
Web Browser	Internet Explorer 5.0, Netscape Communicator 4.6 or later
Development Language/ Tools	JAVA 2, Java Server Pages (JSP), Enterprise Java Beans (EJB) , JavaScript, Servlets, HTML

3.2 Technologies - Quick Reference

The following are the important as well as main factors in choosing the specific technologies and environments for developing the software.

SUN SOLARIS

An Operating system's primary goal is to use the system's resources and hardware in an efficient manner and it's secondary goal is to make the system convenient to use. Solaris satisfies the both factors and hence it's prefered in all over the world and in this project also. It's main salient features includes the following : (a) Multi tasking (b) Multi User and (c) System Portability. 'Multitasking' means multiple tasks can be carried out by placing other tasks in the background, while the user work on one task at a time. A multi user operating system permits several users to use the same computer to carry out their computing jobs. One of the outstanding features that Solaris posses is the ability to port itself to another installation without the need to incorporate any major changes.

JAVA

Java is a powerful but lean object oriented programming language. It has generated a lot of excitement because it makes it possible to program for Internet by creating applets, programs that can be embedded in web page. The context of an applet is limited only by one's imagination.

For example, an applet can be animation with sound, an interactive game or a ticker tape with constantly updated stock prices. Applets can be just little decoration to liven up web page, or they can be serious applications like word processors or spreadsheet. But Java is more than a programming language for writing applets. It is being used more and more for writing standalone applications as well. it is becoming so popular that many people believe it will become standard language for both general purpose and Internet programming.

There are many buzzwords associated with Java, but because of its spectacular growth in the popularity, a new buzzword has appeared lubricious. Indeed, all indications are that it will soon be everywhere. Java builds on strength of C++. It has taken the best features of c++ and discarded the more problematic and error prone parts. To this lean core, it has added garbage collection (Automatic Memory Management), multi threading (The capacity for one program to do more than in time), and security capabilities. The result is that Java is simple, elegant, powerful and easy to use.

WHY JAVA IS IMPORTANT TO THE INTERNET

The Internet help catapult Java to the forefront of programming and Java intern has had a profound effect on the internet. The reason is simple, Java responds the universe of objects. That can move about freely in cyberspace. In a network, there are two broad categories of objects transmitted between the server and your personal computer, passive information and dynamic, active programs. For example, when you read your email, you are viewing passive data. Even when you down load a

program, the programmer's code are still only passive data until you execute it. However, there is a second type of object that can be transmitted to your computer, a dynamic, and self-executive program. Such program would be an active agent for the client computer, yet the server would initiate it.

As desirable as dynamic, networked programs are, they also present serious problems in the areas of security and portability. Prior to Java cyber space laws effectively closed to of the entities now live there. Java addresses these concerns and doing so, as opened the door to an exiting a new form of program.

SERVLETS

A servlet is a body of Java code that is loaded into and runs inside a network service. such as a web server. It receives and responds to requests from clients. For example, a client may neec a specific information from a database; a servlet can be written that receives the request, gets anc processes the data as needed by the client, and then returns it to the client. The Servlet API, which you use to write servlets, assumes nothing about how a servlet is loaded, the server environment in which the servlet runs, or the protocol used to transmit data to and from the user. This allows servlets to be embedded in many different web servers.

Servlets are an effective substitute for CGI scripts: they provide a way to generate dynamic documents that is both easier to write and faster to run. They also address the problem of doing server-side programming with platform-specific APIs. Servlets are developed with the Java Servlet API, a *standard Java extension*. While it is not part of the core Java framework, which must always be part of all products bearing the Java brand, it will be made available with such products by their vendors as an add-on package. A few of the many applications for servlets include,

- Processing data posted over HTTP using an HTML form, including purchase order or credit card data. A servlet like this could be part of an order-entry and processing system, working with product and inventory databases, and perhaps an on-line payment system.

- Allowing collaboration between people. A servlet can handle multiple requests concurrently; they can synchronize requests to support systems such as on-line conferencing.
- Forwarding requests. Servlets can forward requests to other servers and servlets. This allows them to be used to balance load among several servers that mirror the same content. It also allows them to be used to partition a single logical service over several servers, according to task type or organizational boundaries.
- Being a community of active agents. A servlet writer could define active agents that share work among each other. Each agent would be a servlet, and the agents could pass data among themselves.

JAVA SCRIPT

Role of Scripting

There is no definitive definition of a scripting language. Sometimes the term is used to make a distinction from compiled languages. However, some languages like C or C++ can be used for scripting as well as full applications. The term scripting is also used because a language will react to, control, or "Script" a series of events. Even macro languages built into PC applications like spreadsheets, databases, word processors, and multimedia applications are now often called scripting languages.

The purpose of most scripting languages is to extend the capabilities of applications. Just as the authors of this book cannot imagine every creative use you will make of JavaScript, software authors cannot imagine every possible use of their applications. To make their products more versatile, they add a scripting language. With JavaScript you have a scripting language to use your imagination on the Web.

What is JavaScript?

JavaScript is an easy-to-use programming language that can be embedded in the header of your web pages. It can enhance the dynamics and interactive features of your page by allowing you to perform calculations, check forms, write interactive games, add special effects, customize graphics selections, create security passwords and more.

What's the difference between JavaScript and Java?

Actually, the 2 languages have almost nothing in common except for the name. Java is an interpreted programming language, similar to C++. It is powerful enough to write applications and also insert them in a web page as a special object called an "applet." Java has been generating a lot of excitement because of its unique ability to run the same program on IBM, Mac, and Unix computers. Java is not considered an easy-to-use language for non-programmers. JavaScript is much simpler to use than Java. With JavaScript, if you want check a form for errors, you just type an if-then statement at the top of your page. No compiling, no applets, just a simple sequence. The major difference between Java and Javascript is as follows :

JAVASCRIPT	*JAVA*
Interpreted by client	Compiled by the author, run on client
Code integrated in HTML documents	Applets distinct from HTML document
Loose typing of data types	Strong typing of data types
Script limited to browser functions	Stand-alone applications
Works with HTML elements	Goes beyond HTML (Ex : Multimedia)

JAVA SERVER PAGES (JSP)

While there are numerous technologies for building web applications that serve dynamic content, the one that has really caught the attention of the development community is JavaServer Pages TM (JSP TM). And not without ample reason either. JSP not only enjoys cross-platform and cross-Web-server support, but effectively melds the power of server-side Java technology with the WYSIWYG features of static HTML pages. JSP pages typically comprise of: Static HTML/XML components. Special JSP tags I Optionally, snippets of code written in the Java programming language called "Scriptlets." Consequently, you can create and maintain JSP pages by conventional HTML/XML tools. It is important to note that the JSP specification is a standard extension defined on top of the Servlet API. Thus, it leverages all of your experience with servlets. There are significant differences between JSP and servlet technology. Unlike servlets, which is a programmatic technology requiring significant developer expertise, JSP appeals to a much wider audience. It can be used not only by developers, but also by page designers, who can now play a more direct role in the development life cycle.

Another advantage of JSP is the inherent separation of presentation from content facilitated by the technology, due its reliance upon reusable component technologies like the Java Beans™ component architecture and Enterprise Java Beans™ technology.

JSP Advantages

1. Separation of static from dynamic content

With servlets, the logic for generation of dynamic content is an intrinsic part of the servlet itself, and is closely tied to the static presentation templates responsible for the user interface. Thus, even minor changes made to the UI typically result in the recompilation of the servlet. This tight coupling of presentation and content results in brittle, inflexible applications. However, with JSP, the logic to generate the dynamic content is

encapsulating it within external JavaBeans components. These are then created and used by the JSP page using special tags and scriptlets. When a page designer makes any changes to the presentation template, the JSP page is automatically recompiled and reloaded into the web server by the JSP engine.

2. Write Once Run Anywhere

JSP technology brings the "Write Once, Run Anywhere" paradigm to interactive Web pages. JSP pages can be moved easily across platforms, and across web servers, without any major changes.

3. Dynamic content can be served in a variety of formats

There is nothing that mandates the static template data within a JSP page to be of a certain format. Consequently, JSP can service a diverse clientele ranging from conventional browsers using HTML/DHTML, to handheld wireless devices like mobile phones and PDAs using WML, to other B2B applications using XML.

4. Completely leverages the Servlet API

If you are a servlet developer, there is very little that you have to "unlearn" to move over to JSP. In fact, servlet developers are at a distinct advantage because JSP is nothing but a high- level abstraction of servlets. You can do almost anything that can be done with servlets using JSP--but more easily!

HYPER TEXT MARKUP LANGUAGE (HTML)

HTML is basically a scripting language that's mainly used to display static contents on the Internet and Intranet applications, using Browsers. As a formatting language, HTML utilizes SGML(Standard General Markup Language) declarations and the document type declarations (DTD). SGML document has three main parts. The first part defines the character set to be used and tells which characters in that set distinguish text from markup tags. Markup tags specify how the viewer application, or browser, should present the text to the user. The second part of an SGML document specifies the document type and states which markup tags are legal. The third part of an SGML document, called the document instance, contains the actual text and markup tags. Because there is no requirement that the three parts of an SGML document reside in the same physical file, we can concentrate on the document instance. The Web pages you create are document instances.

Most HTML browsers assume a common definition about the character set used, and about which characters distinguish text from markup tags. They also generally agree about a core set of legal markup tags. They then diverge on which additional new markup tags to permit. In terms of universality, HTML along with Hyper Text Transfer Protocal (HTTP) , is a base scripting language that's mainly used for creating a Client – Server applications in Internet / Intranet arenas.

ENTERPRISE JAVA BEANS (EJB)

An Enterprise Bean is a simple class that provides two types of methods: business logic and lifecycle. A client program calls the business logic methods to interact with the data held on the server. The container calls the lifecycle methods to manage the Bean on the server. In addition to these two types of methods, an Enterprise Bean has an associated configuration file, called a EJB Deployment descriptor, that is used to configure the Bean at deployment time.

As well as being responsible for creating and deleting Beans the Enterprise Java Beans server also manages transactions, concurrency, security and data persistence. Even the connections between the client and server are provided by using the RMI and JNDI APIs and servers can optionally provide scalabilty through thread management and caching.

Entity and Session Beans

There are two types of Enterprise Beans: entity Beans and session Beans. An Enterprise Bean that implements a business entity is an entity Bean, and an Enterprise Bean that implements a business task is a session Bean.

Typically, an entity Bean represents one row of persistent data stored in a database table. Entity Beans are transactional and long-lived. As long as the data remains, the entity Bean can access and update that data. This does not mean you need a Bean running for every table row. Instead, Enterprise Beans are loaded and saved as needed. A session Bean might execute database reads and writes, but it is not required. A session Bean might invoke the JDBC calls itself or it might use an entity Bean to make the call, in which case the session Bean is a client to the entity Bean. A session Bean's fields contain the state of the conversation and are transient. If the server or client crashes, the session Bean is gone. A session Bean is often used with one or more entity Beans and for complex operations.

SESSION BEANS	*ENTITY BEANS*
Fields contain conversation state.	Represents data in a database.
Handles database access for client.	Shares access for multiple users.
Life of client is life of Bean.	Persists as long as data exists.
Can be transaction aware.	Transactional.
Does not survive server crashes.	Survives server crashes.

Developing and Running Applications

Deployment tools and an Enterprise JavaBeans server are essential to running EJB applications. Deployment tools generate containers, which are classes that provide an interface to the low-level implementations in a given Enterprise JavaBeans server. The server provider can include containers and deployment tools for their server and will typically publish their low-level interfaces so other vendors can develop containers and deployment tools for their server.

Because everything is written to specification, all Enterprise Beans are interchangeable with containers, deployment tools, and servers created by other vendors. In fact, you might or might not write your own Enterprise Beans because it is possible, and sometimes desirable, to use Enterprise Beans written by one or more providers that you assemble into an Enterprise JavaBeans application.

JAVA DATABASE CONNECTIVITY (JDBC)

SQL is a language used to create, manipulate, examine, and manage relational databases. Because SQL is an application-specific language, a single statement can be very expressive and can initiate high-level actions, such as sorting and merging data. SQL was standardized in 1992 so that a program could communicate with most database systems without having to change the SQL commands. Unfortunately, you must connect to a database before sending SQL commands, and each database vendor has a different interface, as well as different extensions of SQL.

ODBC, a C-based interface to SQL-based database engines, provides a consistent interface for communicating with a database and for accessing database metadata (information about the database system vendor, how the data is stored, and so on). Individual vendors provide specific drivers or "bridges" to their particular database management system. Consequently, thanks to ODBC and SQL, you can connect to a database and manipulate it in a standard way. It's no surprise that, although ODBC began as a PC standard, it has become nearly an industry standard.

Though SQL is well suited for manipulating databases, it is unsuitable as a general application language and developers use it primarily as a means of communicating with all databases—another language is needed to feed SQL statements to a database and process results for visual display or report generation. Unfortunately, you cannot easily write a program that will run on multiple platforms even though the database connectivity standardization issue has been largely resolved.

For example, if you wrote a database client in C++, you would have to totally rewrite the client for each platform; that is to say, your PC version would not run on a Macintosh. There are two reasons for this. First, C++ as a language is not portable for the simple reason that C++ is not completely specified, for example, how many bits does an int hold? Second and more importantly, support libraries such as network access and GUI libraries are different on each platform.

You can run a Java program on any Java-enabled platform without even recompiling that program. The Java language is completely specified and, by definition, a Java-enabled platform must support a known core of libraries. One such library is JDBC, which you can think of as a Java version of ODBC, and is itself a growing standard. Database vendors are already busy creating bridges from the JDBC API to their particular systems. JavaSoft has also provided a bridge driver that translates JDBC to ODBC, allowing you to communicate with legacy databases that have no idea that Java exists. Using Java in conjunction with JDBC provides a truly portable solution to writing database applications.

WEB LOGIC SERVER 5.1

As the industry leading e-commerce transaction platform, WebLogic server provides a number of features critical to developing and deploying mission-critical e-commerce applications across distributed, heterogeneous computing environments. These include:

1. Standards leadership-Comprehensive Enterprise Java support, including EJB and JMS, to ease the implementation and deployment of application components.

2. Enterprise e-business scalability-WebLogic Server's software clustering of dynamic web pages (Servlets, JavaServer Pages)

3. EJB business components, coupled with client connection sharing and database resource pooling, maximize efficiency in the use of critical resources.

4. Robust administration-WebLogic Server offers a comprehensive pure-Java console operation, Zero Administration Client (ZAC) for managing the distribution of applications to remote users, and dynamic application partitioning and cluster membership.

5. E-commerce-ready security-WebLogic Server features Secure Sockets Layer (SSL) support for integration of encryption and authentication security into e-commerce solutions.

6. Maximum development and deployment flexibility-WebLogic Server features tight integration with and support for leading databases, development tools, and other environments.

WebLogic Server operates at the center of a multitier architecture. In this architecture, business logic is executed in WebLogic Server, rather than in client applications. The resulting "thin" client, three tier architecture allows the client to manage the presentation layer, the application server to manage the business logic and the back end data services manage the data. This makes WebLogic Server the ideal platform for web-enabled e-commerce applications.

In the middle tier, WebLogic Server provides a reliable, highly scalable platform for hosting business logic. It serves static and dynamic web pages, and manages database access, security, and transaction services for applications.WebLogic Server centralizes access to a variety of third-tier resources and back end services. Tier three services include databases, messaging systems, transaction monitors, real-time data feeds, and existing enterprise information systems integrated with WebLogic Server via 'connectors.' WebLogic Server shields client applications from propriety interfaces and provides efficient sharing of the resources.

Managing access to critical back-end resources from the middle tier helps to secure them. In a web-based application, all client interaction is accomplished with HTML

means that clients outside of a firewall can access a WebLogic Server application without compromising the security of the back-end resources the application uses. Java-based WebLogic Server applications provide similar security for back-end resources through the use of components located in the middle tier. Instead of focussing on all of the complexities of system infrastructure, WebLogic Server developers focus on modelling business processes and solving application needs.

With a WebLogic Server application deployed to the web, for example, all of the application- specific logic can be located in WebLogic Server; clients invoke applications by requesting web pages and providing input through HTML forms. Servlets or JavaServer Pages, running on WebLogic Server, accept input from the browser and provide the content for the browser.

Servlets and JSPs can access a variety of other services in WebLogic Server. They can execute database queries using standard JDBC (Java Database Connectivity) calls or Enterprise JavaBeans. They can invoke Enterprise JavaBeans that encapsulate specific business logic, and they can use JMS (Java Messaging Service) to exchange messages with other clients and applications.

With a Java client application, the application logic can be selectively deployed in the client or in WebLogic Server. Java clients are typically deployed to overcome the limitations of browsers access to resources on the client computer and less dynamic user interfaces. WebLogic Server can be a primary web server or it can process requests redirected to it by an existing web server. For example, it is common to have a Netscape Enterprise Server, Microsoft Internet Information Server, or Apache web server handle requests for static HTML web pages, but pass their servlet and JSP page requests to WebLogic Server.

RDBMS : Oracle 8.0

A database server is the key to solving the problems of information management. In general, a server must do

many users can concurrently access the same data. All this must be accomplished while delivering high level of performance. A database server must also prevent unauthorized access and provide efficient solutions for failure recovery.

The Oracle Server is an object-relational database management system that provides an open, comprehensive, and integrated approach to information management. An Oracle Server consists of an Oracle database and an Oracle Server instance. The Oracle Server provides efficient and effective solutions with the following main features:

1 *Client/server (distributed processing) environments.*

To take full advantage of a given computer system or network, Oracle allows processing to be split between the database server and the client application programs. The computer running the database management system handles all of the database server responsibilities while the workstations running the database application concentrate on the interpretation and display of data.

2. *Large databases and space management*

Oracle supports the largest of databases, potentially terabytes in size. To make efficient use of expensive hardware devices, it allows full control of space usage.

3. *Many concurrent database users*

Oracle supports large numbers of concurrent users executing a variety of database applications operating on the same data. It minimizes data contention and guarantees data concurrency.

4. *High transaction processing performance*

Oracle maintains the preceding features with a high degree of overall system performance. Database users do not suffer from slow processing performance.

5. Portability and Compatibility

Oracle software is ported to work under different operating systems. Applications developed for Oracle can be ported to any operating system with little or no modification.

Resource Bundle

Localization is the ability to customize a program to the user's language (e.g., for prompts and error messages) and display characteristics (e.g., for how to display times and dates). The different customizations are defined by individual areas or regions, each called a Locale.

Before you can localize any programs, you need to isolate the locale dependent pieces. This isolation process is called *internationalization*. In the simplest case, this involves the moving of all String constants into something called a ResourceBundle. However, resource bundles are not restricted to just text messages and may include images, sounds, and numeric information. And this isn't the only step involved

Why You Need It

Imagine taking this program and trying to run it in Italy and Finland. While it will run, there are a few internationalization issues:

- All text labels are in English
- The Date display format is specific to the United States
- The Money display format is specific to US $
- The map image and city listed are in the United States

You'll now see how

In order to localize labels, you need to pull them all out and do lookups at runtime. The means to do this is through the ResourceBundle class. Resource bundles provide a simple key-value lookup and may be provided via .class or .properties files. (Other means can also be provided, however you would have to do all the work yourself.)

To provide a lookup list via a class, you subclass ListResourceBundle and override the getContents method to return an Object[][]. For each pair of elements in the array, the first would be the lookup key and the second the value for the key for the given Locale.

You would then use this bundle to lookup resources in a two step process: load the resource list and search it. To load the list, use ResourceBundle.getBundle(). This will load the bundle for the appropriate Locale. Since there is only one so far, and it is the default, the training-labels-bundle one is what is loaded.

Once you have a bundle, you lookup entries via its getString() method:

Then, if you want the program to support Italy and Finland locales to display localized labels, you would have to create two new classes: training-labels-bundle-it and training-labels-bundle-fi. In these classes, only those entries that were different in the new locale would need to be specified in the bundle. Now that three bundles are defined, based on the current locale setting, the getBundle() method would fetch and load the appropriate class.

For Finland, the appropriate bundle would be:

public class training-labels-bundle-fi

For Italy, the appropriate bundle would be:

public class training-labels-bundle-it

Property Resource Bundles

In addition to providing resource bundles as ListResourceBundle classes, you can provide PropertyResourceBundles. These are property files where each resource is specified on its own line as text, as in Date=June 3, 2000. Property resource bundle files are named Bundlename_*Localization*.properties.

The previously described program requires three properties for the actual date, cost, and location map. You can create another bundle in filename training-settings-bundle.properties for this information. Since this information is likely to change more frequently, it makes sense to be editable by a human, without requiring recompilation.

Date=June 3, 2000

Cost=1200.00

Map=SMarea.gif

For Finland, the appropriate bundle file would be named training-settings-bundle-fi.properties and may include:

Cost=6240

Map=ilmakuva.gif

For Italy, the appropriate bundle file would be named training-settings-bundle-it.properties and may include:

Cost=2168400.00

Map=Milan.gif

This assumes the date for all three classes would be the same at the different locations.

Locale-Specific Formats

Once you've pulled all your string constants into resource bundles, its not time to figure out how to display that information. For instance, dates, numbers, and messages may be displayed

Date Formatting

Display formats of dates are not universal. Is "1/2/99" January 2nd or February 1st? It depends. Java supports the parsing and display of dates through the DateFormat class. There are four formats supported for dates. Each represents a constant in the class:

- SHORT - Completely numeric: 1/2/99
- MEDIUM - Abbreviated month: Feb 1, 1999
- LONG - Full month name: February 1, 1999
- FULL - Completely specified: Saturday, February 1, 1999 AD

The training-settings-bundle stored the contents of the "Date" string in the US/Long format. So, when converting from a text string to a Date object with the help of the DateFormat class, you must specify what format the input text is for the date:
Once you have a Date, you need another DateFormat to specify how to print out the date. This allows you to format the date for the local display properties (for things like month/day order and '.' vs. '/' as the separator).

For June 3, 2000, this results in the following values:

- US - 6/3/00
- Italy - 03/06/00
- Finland - 03.06.00

Number/Currency Formatting

The NumberFormat class works similarly as the DateFormat class. You can get an instance for a specific Locale via getNumberInstance() for normal number format, getCurrencyInstance() for monetary values, and getPercentInstance() for localized percentage format. For this example, the cost of the class needs to be displayed, so the getCurrencyInstance() method is needed to display the

However, in the resource file, this value was stored as a raw number; so for input the getNumberInstance() mehtod needs to be used.

For 1200.00, this results in the following values:

- US - $1,200.00
- Italy - L. 1.200,00
- Finland - 1 200,00 mk

Notice that it does include local currency symbols. However, it doesn't do currency conversions. It is your responsibility to place different costs for each locale in the resource bundle.

Localized Messages

While resource bundles allow you to get localized message information, if an error message -- or any message the user might see, such as dialog boxes -- needs to be ordered differently based upon the locale involved, just using resource bundles is not a sufficient solution. The MessageFormat class provides the means to properly create these types of messages. A resource bundle would contain the format of the message, while the run-time circumstances would fill in the specifics.

For instance, one part of the United States might like to see error messages of the form "I/O Exception while loading: Foobar.java," while another might like "Foobar.java loaded unsuccessfully: I/O Exception." Or, the message

The disk G contains 3 files. (English)

is translated into French as

Il y a 3 fichiers sur le disque G. (French)

Note that the order of the parameters "G" and "3" is reversed.

To do this, one would first need to create two resource bundles (as above), one for each part of the country. In bundle one, the format for this specific error would be defined as: "{0} while loading: {1}", while

the second would have: "{1} loaded unsuccessfully: {0}". The {0} represents a position holder to substitute arguments into the message. The position holders start at 0 and increase.

When it comes time to actually create the message to display, with the arguments filled in, you use the MessageFormat.format() method. This takes two arguments, the first being the message to format, the second being an Object [] of the arguments.

The output of running follows:

Foobar.java while loading: I/O Exception

I/O Exception loaded unsuccessfully: Foobar.java

Besides just plain numbers as arguments for the formatting, you can specify datatypes and formatting information, to automate the use of the other formatting classes, like DateFormat and NumberFormat. For instance, to specify the second parameter should be output as a long-formatted date string, you would use {1,date,long} as the argument. The MessageFormat API documentation provides complete information on all the special formatting options.

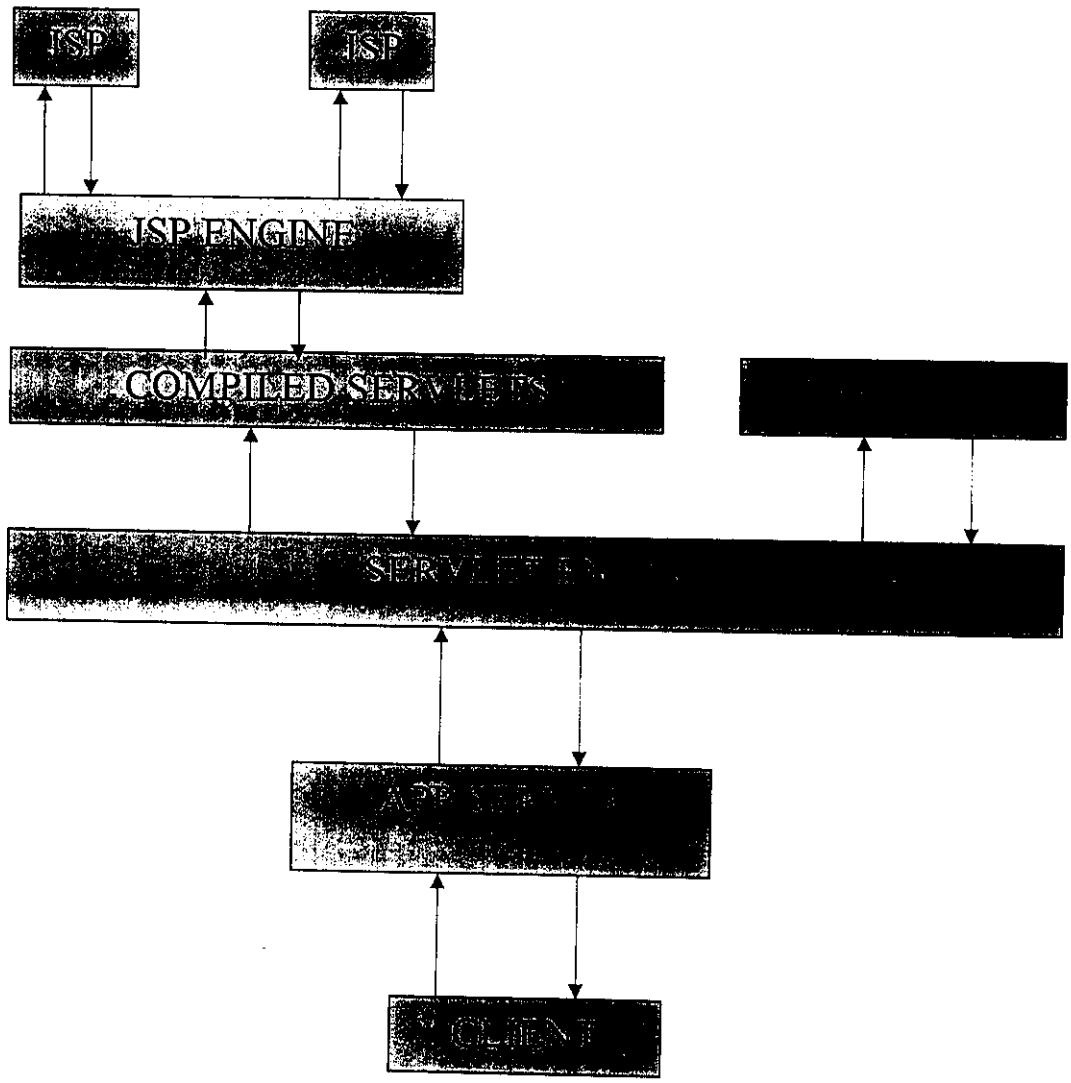
WAP :

The Wireless Application Protocol (WAP) is an open, global specification which gives mobile users with wireless devices the opportunity to easily access and interact with information and services. The protocol is developed by WAP Forum http://www.wapforum.org/, an organization of some of the most powerful Internet and telecom companies.

The advantages of WAP is that if you have a mobile phone you will probably have noticed that your phone is a more "private" device than your computer. Your phone is always within reach, you can get hold of all your friends using it and you have a lot of personal information in store.As opposed to your PC, the WAP phone will always be within range. It invites you to order something wherever you are, and you can search for information in the middle of a discussion around the café table.

SYSTEM DESIGN

4.1 Design Architecture

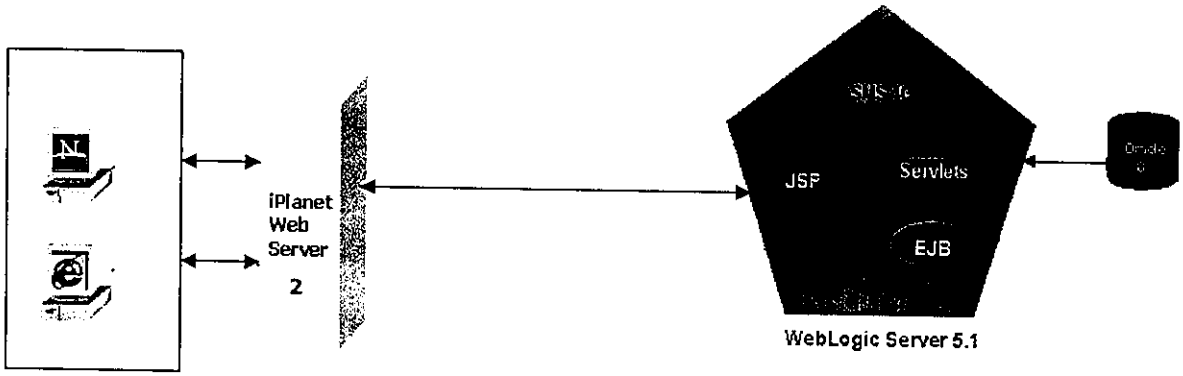


WEB APP SERVER ARCHITECTURE

The above diagram depicts the architecture of the web app server.The web application server will maintain the JSP's and SERVLETS in the above manner.The client request will be first sent to the app server which will then direct the request to the JSP or SERVLET as all JSP's will be compiled into servlets.The appserver will have a JSP engine and a SERVLET engine which will handle the requests.

ARCHITECTURE OF THE JAVA ENGINE

The JAVA engine will have the architecture given above.The JAVA Engine will include the DB Handler and Error Handler modules .All the JSP pages, Servlet and beans reside in the server the client will use his local web browser to open the JSP page, it is routed through an iPlanet server. All the JSP pages and the bean are linked to the central Servlet, which will control the action and flow of the system. All the transactions will be done by the Enterprise Java Bean (Stateless session bean). The bean will communicate with the database. The advantage of using this architecture is making use of the component based modeling of Java.



MANUAL DATA ENTRY MODULE ARCHITECTURE

The manual data entry module will cater Intranet users over the local Intranet network. All the JSP pages, Servlet and beans reside in the server the client will use his local web browser to open the JSP page it is routed through an iPlanet server. All the JSP pages and the bean are linked to the central Servlet, which will control the action and flow of the system. All the transactions will be done by the Enterprise Java Bean (Stateless session bean). The bean will communicate with the database. The advantage of using this architecture is making use of the component based modeling of Java.

4.2 Screen Design

The screen design is available only for the manual data entry module. The following are the screens.

USER IDENTIFICATION

Introduction.

The user will access the Data Entry Module through the Intranet. The users will be able to login using a browser running on their PC's .The browsers to be supported for this application are Netscape 4.6 or above. He is welcomed by a Home Page which is the login page of the Data Entry Module.

Without proper user authentication the Data Entry Module will deny all attempts of unauthorized access. The user identification and entry to the Data Entry Module is maintained by a security shell

SECURITY SHELL features

The security shell focuses on achieving maximum control and security to the computer operation while providing maximum flexibility to users. The major features are:-

- Providing a software layer to oversee and control the clusters so that the entire system looks like a virtual machine' to the user.
- Centralized administration of user profile, access rights, activities and control measures.
- Providing a single sign on facility to Users, so that by entering one user Id and password, a User can perform all operations for which he is authorized, on any machine / database
- Maintain audit trails of all sensitive activities
- Centralized maintenance of parameters, by which addition / deletion of machines, databases and users is seamless
- Monitor activities of all the users so that security can be maintained.

USER HOME PAGE

Introduction

The user is welcomed by a home page containing static text

The user can perform the following functions:

- Enter new Data.
- Edit data already present.
- Delete data.
- Change user password.

ENTER NEW DATA

Introduction

The user will be provided with an input screen where he will be able to enter all the information based on the nature of the system.

Field description

The following table gives the fields available and all details about the fields

Field	Type	Length	Default Value	Rules
Customer Id	Text	9	Nil	Only numbers(no decimal point)
Customer Financial Institution Id	Text	9	Automatically generated in a sequence	Only numbers(no decimal point)
Financial Institution Id	Text	6	Nil	Only numbers(no decimal point)
Account Type	Select box		*Credit Card *Investment *Banking	Nil
Account Balance	Text	20	Nil	Only numbers

Error Message

Fields	Error Messages
Customer Id	Please enter a valid customer Id
Financial Institution Id	Please enter a valid FI Id
Account Type	Please select an account type
Account Balance	Please Give a valid amount

SEARCH SCREEN

Introduction

The search screen is used to search for searching for a specific account detail based on a keyword the keyword used in this application is the Customer Id.

Field description

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Customer Id	Text	9	Nil	Only numbers(no decimal point)

In the above fields the dates fields are set using the set option which we will have a calendar from which the user will select the date.

Error Message

[illegible]	[illegible]
Customer Id	Id number can be a number only

SEARCH RESULT SCREEN

Introduction

This screen will be displayed on searching for a specific criterion. It will display all the data matching the specific criteria. In this case it will display all the accounts present for the Customer Id presented for the search. Each of these will be clickable. Which will take us to the account detail screen

ACCOUNT DETAIL SCREEN

Introduction

This page display's the account details for the selected account all the details are extracted from the database and created in the form of a report and displayed to the user. The screen will have two links one is the delete confirmation page and edit page.

CONFIRM DELETE SCREEN

Introduction

This screen will be displayed on selecting the delete option. It will display the account nickname and FI name and will ask whether we want to confirm account deletion. We will have a yes and a no button. The yes button will delete the account and take us to the home page. The no button will take us back to the search result page without deleting the account.

EDIT SCREEN

Introduction

This screen will be displayed on selecting the edit option. It will be similar to Account entry screen. The customer Id and Customer Financial Institution Id will not be edited. On clicking submit the data posted in the page it will be modified in the backend.

Field description

The following table gives the fields available and all details about the fields

Field	Type	Length		
Customer Id	Text	9	Extracted from database (not editable)	Only numbers(no decimal point)

Customer Financial Institution Id	Text	9	Extracted from database (not editable)	Only numbers(no decimal point)
Financial Institution Id	Text	6	Extracted from database	Only numbers(no decimal point)
Account Type	Select box		*Credit Card *Investment *Banking	Nil
Account Balance	Text	20	Extracted from database	Only numbers

Error Message

Field	Error Message
Financial Institution Id	Please enter a valid FI Id
Account Type	Please select an account type
Account Balance	Please Give a valid amount

CHANGE PASSWORD SCREEN

Introduction

This screen will be displayed on selecting the change password option. Three fields are present one is for old password, new password and confirm new password .on submitting valid details the password for the user will be reset.

Field	Type	Length	Valid Values	Default Value
nter Old assword	Password	15	Nil	None

Enter new Password	Password	15	Nil	None
Reenter new password	Password	15	Nil	None

Error Message

[illegible]	[illegible]
Enter password	The two password's don't match
Re-enter password	The two password's don't match

CONFIRM CHANGE PASSWORD SCREEN

Introduction

This screen will be reached from change password after changing the password successfully.

LOGOUT SCREEN

Introduction

This screen will be displayed on selecting the logout option.

USER REQUEST DESIGN

Introduction

We will have to design the user request formats and also the response formats. We will also have to design the error responses in case of an error. This is the major part that will be customized for each client based on his requirement.

4.3 Detailed Design

Detailed design of a system includes developing prototypes, User interfaces and Backend databases. For this phase, Data Flow diagrams (DFD), Entity Relationship diagrams (ERD) and System Flow Charts (SFC) are used.

Data Flow Diagrams depict how data interact with a system. DFDs are extremely useful in modeling many aspects of a business function because they systematically subdivide a task in to its basic parts, helping the Analyst understand the system, which they are trying to model.

A DFD models a system by using external entities from which data flows to a process which transforms the data and creates output data which goes to other processes or external of files. Data in files may also flow to processes as inputs.

The main merit of data flow diagram is that it can provide an overview of what data a system would Process, what information of data are done, what files are used and where the results flow. The graphical representation of the system makes it a good communication tool between the user and an analyst. It's difficult to represent the business process through verbal description alone. Here data flow diagrams help in illustrating the essential component of a process and the way they interact.

DFD Components

DFDs are constructed using four major components : (a) External entities (b) Data stores (c) Processes and (d) Data flows.

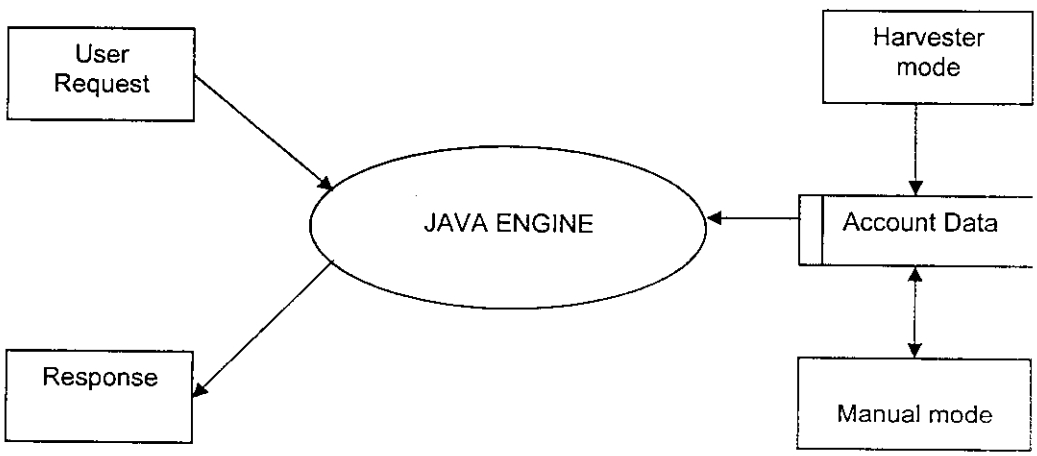
External entities represent the sources of data that enter the system or the

may be a Databases or individual files. Processes represents activities in which data is manipulated by being stored or retrieved or transformed in some way. Data flows represent the movement of data between other components, for example a report produced by a process and sent to an external entity.

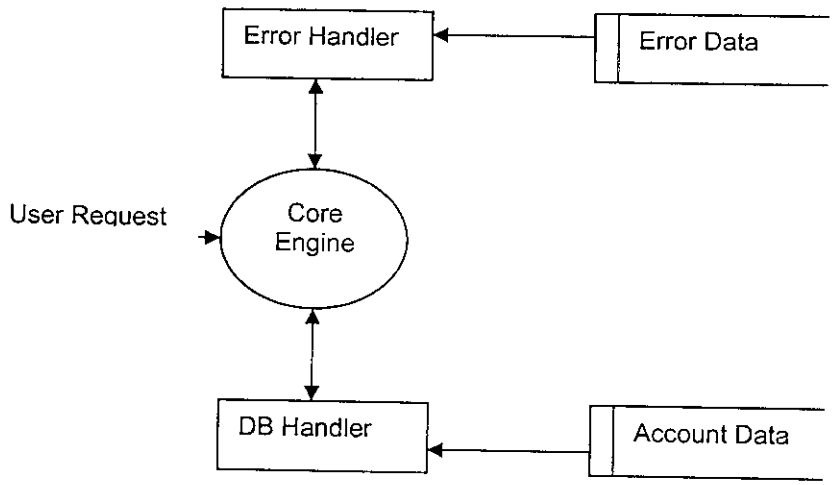
A circle is used to depict a process. Both input and output are dataflows. An arrow represents the data flows. External entities are represented by rectangles. Entities supplying data are known as *Sources* and those that consume data are called as *Sinks*

4.4 Data Flow Diagrams

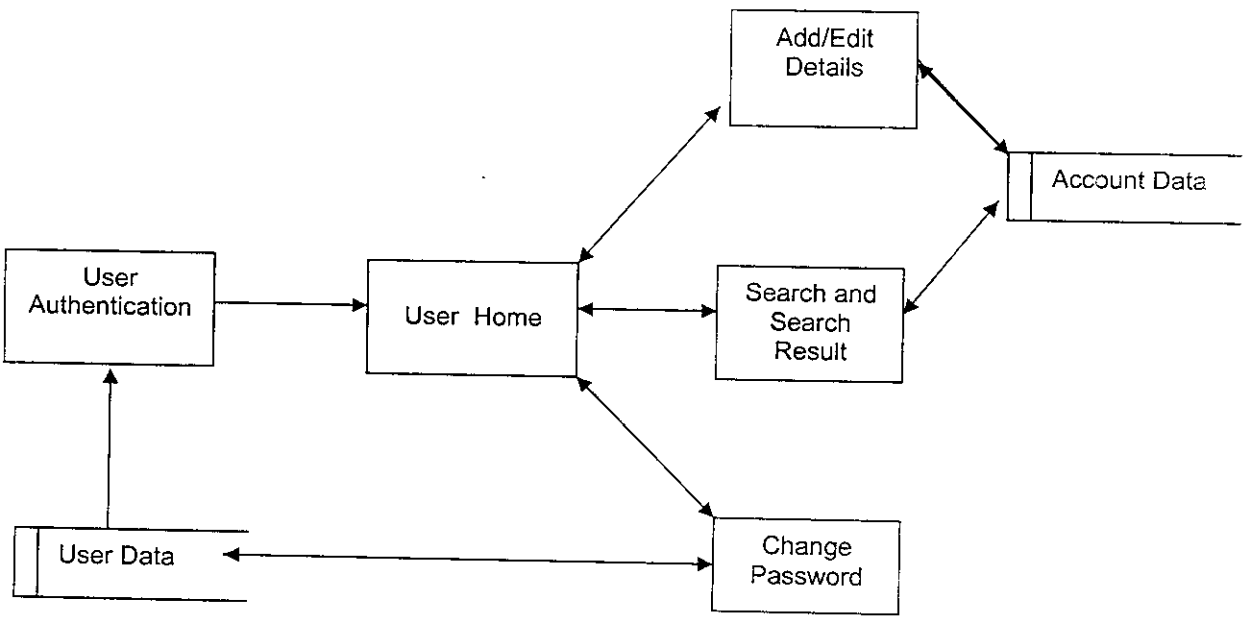
Context Level Flow



JAVA ENGINE


Error Handler
Error Data
User Request
Core
Engine
DB Handler
Account Data

Manual Data Entry



4.5 Table Details

Table Name : HostingHome

Description : Stores Details about Server home details

Primary Key : HOME_ID

Field Name	*Null*	*Field Type*
HOME_ID	NOT NULL	NUMBER
HOME_NAME	NOT NULL	VARCHAR2(255)
CREATED_ON	NOT NULL	DATE
CREATED_BY	NOT NULL	VARCHAR2(30)
MODIFIED_ON		DATE
MODIFIED_BY		VARCHAR2(30)
ENC_CUSTODIAN_KEY		VARCHAR2(1000)
HOME_CATEGORY	NOT NULL	VARCHAR2(30)

Table Name : User

Description : Stores details about SS User

Field Name	*Null*	*Field Type*
PERSON_ID	NOT NULL	NUMBER
FIRST_NAME	NOT NULL	VARCHAR2(255)
LAST_NAME	NOT NULL	VARCHAR2(255)
HOME_ID		NUMBER
MIDDLE_NAME		VARCHAR2(255)
EMAIL		VARCHAR2(255)
GENDER		VARCHAR2(255)
DAYTIME_PHONE		VARCHAR2(255)
EVENING_PHONE		VARCHAR2(255)
LOGIN_NAME	NOT NULL	VARCHAR2(255)
FAX		VARCHAR2(255)
UENCRYPT_PASSWORD		VARCHAR2(255)
CREATED_BY	NOT NULL	VARCHAR2(30)
CREATED_ON	NOT NULL	DATE
LOGIN_ATTEMPT		NUMBER
LAST_VISITED_DATE		DATE
IS_LOCKED	NOT NULL	NUMBER(1)
IS_DELETED	NOT NULL	NUMBER(1)
MODIFIED_BY		VARCHAR2(30)
MODIFIED_ON		DATE
CHANGE_PASSWORD_STATUS	NOT NULL	NUMBER(1)
FORCED_CHANGE_PASSWD		NUMBER(1)
IS_DELETED_TIMESTAMP		DATE

Table Name : CSC_SUMMARY
Description : Stores details about Customer Credit Transactions

Name	*Null*	*Field Type*
CUSTOMER_INSTITUTION_ID	NOT NULL	NUMBER
DATE_RETRIEVED		DATE
ACCOUNT_BALANCE		NUMBER
SUMMARY_DATE		DATE
AVAILABLE_BALANCE		NUMBER
FEES_THIS_PERIOD		NUMBER
FEES_YTD		NUMBER
INTEREST_THIS_PERIOD		NUMBER
INTEREST_YTD		NUMBER
DESCRIPTION		VARCHAR2(255)
CREATED_ON	NOT NULL	DATE
CREATED_BY	NOT NULL	VARCHAR2(30)
MODIFIED_ON		DATE
MODIFIED_BY		VARCHAR2(30)
LAST_DOWNLOAD_TRANSACTION_DATE		DATE
DUE_DATE		DATE

Table Name : MMF_SUMMARY
Description : Stores details about Money Market Funds

Name	*Null*	*Field Type*
DATE_RETRIEVED		DATE
CUSTOMER_INSTITUTION_ID	NOT NULL	NUMBER
SUMMARY_DATE		DATE
DESCRIPTION		VARCHAR2(255)
CLOSING_SHARES_AMOUNT		NUMBER
OPENING_SHARES_AMOUNT		NUMBER
SHARES_PURCHASED		NUMBER
SHARES_REDEEMED		NUMBER
NET_DIVIDEND_REINVESTED		NUMBER
YIELD		NUMBER
FEES_THIS_PERIOD		NUMBER
FEES_YTD		NUMBER
CREATED_BY	NOT NULL	VARCHAR2(30)
CREATED_ON	NOT NULL	DATE
MODIFIED_BY		VARCHAR2(30)
MODIFIED_ON		DATE
CURRENCY_CODE	NOT NULL	VARCHAR2(3)
DUE_DATE		DATE

Table Name : Customer_Financial_Institution
Description : Stores details about Money Market Funds

Name	*Null*	*Field Type*
CUSTOMER_INSTITUTION_ID	NOT NULL	NUMBER
FINANCIAL_INSTITUTION_ID	NOT NULL	NUMBER
PERSON_ID	NOT NULL	NUMBER
ACCOUNT_TYPE_ID	NOT NULL	NUMBER
ACCOUNT_NUMBER	NOT NULL	VARCHAR2(255)
ACCOUNT_LOGIN_INFO	NOT NULL	VARCHAR2(2000)
UENCRYPT_ACCOUNT_PASSWORD		VARCHAR2(255)
ACCOUNT_NAME_ALIAS		VARCHAR2(2000)
USER_SPECIFIC_MINIMUM_BALANCE		NUMBER
CRAWLED_ACCOUNT_NUMBER		VARCHAR2(255)
USER_SPECIFIC_INTEREST_RATE		NUMBER
AVAILABLE_OVERDRAFT		NUMBER
ALLOWED_OVERDRAFT_AMOUNT		NUMBER
USED_OVERDRAFT		NUMBER
CASH_NEEDS		NUMBER
SAFETY_CUSHION		NUMBER
LAST_DOWNLOAD_TRANSACTION_DATE		DATE
LAST_DOWNLOAD_ATTEMPT_DATE		DATE
CREATED_BY	NOT NULL	VARCHAR2(30)
LAST_SUCCESSFUL_DOWNLOAD_DATE		DATE
CREATED_ON NOT NULL		DATE
LAST_DOWNLOAD_STATUS_CODE		VARCHAR2(1)
MODIFIED_BY		VARCHAR2(30)
IS_FT_APPROVED		NUMBER(1)
USER_CREDIT_PREFIX		VARCHAR2(255)
USER_DEBIT_PREFIX		VARCHAR2(255)
USER_CREDIT_ROUTING_NUMBER		VARCHAR2(255)
USER_DEBIT_ROUTING_NUMBER		VARCHAR2(255)
MICR_ON_CHECK		VARCHAR2(255)
MODIFIED_ON		DATE
DEBIT_ACCOUNT_NUMBER		VARCHAR2(255)
LAST_DOWNLOAD_STATUS_MESSAGE		VARCHAR2(2000)
IS_ACCOUNT_WEBCRAWL_VERIFIED		NUMBER(1)
IS_SIGNED_FOR_RECOM_IN		NUMBER(1)
IS_DASHBOARD NOT NULL		NUMBER(1)
IS_SIGNED_FOR_RECOM_OUT		NUMBER(1)
IS_SIGNED_FOR_FT_IN NOT NULL		NUMBER(1)
IS_SIGNED_FOR_FT_OUT NOT NULL		NUMBER(1)
USER_SET_LIMIT_PER_TRANSFER		NUMBER
MAX_FT_OUT_LIMIT		NUMBER
MAX_FT_IN_LIMIT		NUMBER
NO_OF_USERS_USING_ACCT		NUMBER(1)
ACCOUNT_NAME_ALIAS_SOURCE		VARCHAR2(2000)
MISC		VARCHAR2(2000)
CURRENCY_CODE		VARCHAR2(3)
PREVENTIVE_LOCK		NUMBER
IS_ACCOUNT_WEBCRAWL_SUSPENDED	NOT NULL	NUMBER(1)
TRUST_MODE		VARCHAR2(1)
PARENT_CFI_ID		NUMBER
IS_RECLASSIFICATION_REQUIRED		NUMBER(1)

OFX_ACCOUNT_NUMBER	VARCHAR2(255)
IS_AUTOLOGIN NOT NULL	NUMBER(1)

Table Name : Activity_Log
Description : Stores details activites and events.

Name	*Null*	*Field Type*
ACTIVITYLOG_ID	NOT NULL	NUMBER
CATEGORYTYPE_ID	NOT NULL	NUMBER
ACTOR_ID	NOT NULL	NUMBER
EVENT_ID	NOT NULL	NUMBER
USER_NAME	NOT NULL	VARCHAR2(30)
IP_ADDRESS		VARCHAR2(15)
TIME_STAMP		DATE
HOST_NAME		VARCHAR2(255)
EVENT_RESULT	NOT NULL	VARCHAR2(30)
REMARKS		VARCHAR2(255)

SYSTEM DEVELOPMENT

So far, in previous chapters the design of the system was discussed. This chapter gives a brief description of the Functional specifications, system features, external interfaces and Coding standards for the whole system development.

5.1 Functional Specifications

The requirement should be written down, and this specification should refer a working Prototype, rather than embedded screen shots in the document. As a result, the document and the Prototype together constitute the signed-off requirements. These two combinations provide the much more realistic representation of the requirements and leads to a solution that is much more closer to what the client really want.

5.2 System Features

The SS system offers the following features:

Help files, and Context Specific Help

Context Specific help will be available on each module, to assist the Client's user and the Client's customer in the process of using the system. Help files oriented towards performing a given step or a process will also be provided, to help any new user get acquainted with the SS Application.

Audit Trail

Audit Trail for the entire user login related details will be available. All audit logs for the invalid id's, failed authorization, etc will also be available to the administrator a time stamp is included in all of these logs to maintain accuracy of time.

Error Messages and Error logs

Java Script validation is done at all stages to ensure that the errors get trapped at the client edge by itself. All the fields in the Web Modules are validated for data types and file location before the request is send to the client. Wherever necessary alerts and error messages are given to the user to ensure consistency in the application. Apart from the client side validation server side validation is done by the JSP portion and error log information for all kinds of breakdowns and other errors that occurred during service times are stored in the oracle database.

Data Validations

Though Data Validation is not necessary in most of the cases due to the pre-structured nature of data in application , data validation is done at the Web Module level and the Manual Data Entry module level to ensure data consistency.

Intuitive Navigation & Screen flow

The SS system will support intuitive navigation, so that the system is user friendly, as well as efficient in functioning.

Security

Style Studio is maintained over the Internet. It follows the current security policies and it is possible to easily update for new security standards in future. Encryption has to meet the CISC standards. A CISO approved Security shell is used.

Customer logs

Style Studio provides the user with various types of Logs for different status. These logs will aid in providing quick service to the customer

5.3 External Interfaces

Security Shell

The User will log-on to security shell to access the Style Studio manual data entry module. The Security shell will validate the user login. The Security shell will allow wrong logins only for 3 times, after that the system will lock the user ID. Then the user has to apply to the administrative users to unlock his login ID.

5.4 Coding Standards

There are two main principle characteristics, which are standards to coding. (a) They force to maintain a methodical and disciplined approach to coding and (b) They constantly remained the internal quality of the code.

The very decision to use standards will affect the coding. By making it clear that the standards are mandatory rules, not mere guidelines, meeting standards is an integral part because standards are not guidelines, they should not be flexible.

Traditionally the coding standards are focussed on the following topics : (a) *Naming* (b) *Layout* (c) *Commenting* and (d) *Coding* : Do's and Don'ts , such as error handling. The emphasis on writing code that's shareable , that's other programmers can also use it easily.

The proposed system is developed using the above mentioned standards. Variable names are declared meaningful with respect to the information stored as well as the data type of the variable. Comments help the programmer what the module does or the set of statements do. Error handling is well taken care and Error messages are meaningful and suggestive.

SYSTEM TESTING

6.1 Testing concepts

Software is only one element of a larger computer based system. Ultimately Software is incorporated with other system elements (Ex. New hardware) and a series of system integration and validation tests are conducted. System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system.

Testing presents an interesting anomaly for the software development. The testing phase creates a series of test cases that are intended to 'Demolish' the software that has been built. A good test case is one that has a high probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error.

Testing process breakes applications down in to two main parts : Unit Testing and System testing. In Unit testing, the modules of the system are tested as individual units. Each unit has definite input and output parameters and often a definite single function. In System testing, the system is tested as a whole; that's inter communication among the individual units and functions of the complete system is tested.

Many of the problems experienced with testing Java based applications occur because testers are trying to apply these conventional method to system for which they aren't appropriate. These are fundamental differences in the way that Java based applications. In Java based applications , one can do things that have no equivalent in a Non–GUI application.

To test a GUI based application, the old testing methods has to be broken in to four test streams. (a) Destruction, (b) Window Design (c) Navigational and (d) Functional Testing.

6.2 Destruction Testing

In this type of testing, the application is tested until it does something it's Not supposed to often in a totally unstructured fashion. For example, in a text field where only numeric values should be entered , all types of data are tried until only numeric values are accepted. Even the structure of the field is also tested by trying to resize the field, drag it, move it and position it out of the screen.

6.3 Window Design Testing

This kind of test proves that each individual window (Such as Primary, Secondary, Pop-Up, Dialog box and Message box) that a system consists of has been designed and built according to the project standards. The best method of ensuring this in the form of a check-list that's, to be checked and signed off by the user.

6.4 Navigational Testing

This test determines whether each window can be navigated to by initiation (Via all the multiple way of initiating) of all the different functions from any other appropriate window in the system, without necessarily performing any of the detailed processing that might be required when it gets there. Again the best format for this test is a check-list. Each list is unique to the particular window that's being tested. The list of navigational actions and results can be easily re-tested and verified.

6.5 Functional Testing

Having now tested that an individual window (or group of windows) is designed Correctly, contains all the functions is should, contain the required methods for initiating those

functions and can navigate via those functions to all the places it has to go, one can now proceed to the 'Real' testing of the system.

Functional testing ensures that the nuts and bolts functions of the system are actually tested. For example, when you initiate the save function for a transaction, does the information saved correctly to the data source? The list of such tests will be a smaller list once the window design and navigational aspects of initiating a function are separated out. This part of the testing is probably the one that equates most closely with old concept of Unit testing.

6.6 Test Scenarios

Test case scenario is a single or a sequence of steps of checks that are made on the system to check how the system reacts for each scenario and recheck the points where the system is considered sensitive. Here are a few sample scenarios.

User Request Sample Scenarios

SCENARIO 1

- A valid user sends a valid request.

SCENARIO 2

- An invalid user sends a valid request.

SCENARIO 3

- A valid user sends an invalid request.

SCENARIO 4

- A invalid user sends a invalid request.

Manual Data Entry Module Sample Scenarios

SCENARIO 1

- A valid user enters the system.
- Searches for a valid entry.
- Edits a valid entry.
- Adds a valid entry.
- Changes his password by providing valid details.
- The user log's out

SCENARIO 2

- An invalid user tries to log into the system

SCENARIO 3

- A valid user enters the system.
- He tries to search fro an invalid entry

SCENARIO 4

- A valid user enters the system.
- He searches for a valid entry.
- He makes invalid edit.

SCENARIO 5

- A valid user enters the system.
- He adds an invalid entry.

SCENARIO 6

- A valid user enters the system.
- He changes his password using invalid values.

6.7 Bug List

After they test the system once completely using the individual test cases and test scenarios the testing team will release a bug list. Bug list is a list of all the bugs identified in the system. Here is given two sample bug lists that was received for the SS system after the first round of official testing.

Project Name: SS	**Module / Screen Name: Manual Data Entry Module (Home Page)**
Tested By:	**Test Conducted On:**

S. No.	Test Condition	Expected Result	Actual Result
1.	When we click on the help from	Help screen should be displayed	The same screen is displayed
2.	On Select Logout	The system should logout completely and close the window	A Blank Screen is appearing. **(SHOW STOPPER)**
3.	On screen load.	The menu bar contents must be displayed clearly	The menu is not displayed clearly.
4.	On screen load	A welcome message banner should be displayed	The screen is blank
5.	On screen load	The color of the bar displayed on top and bottom must be Grey.	The color is blue.
6.	On placing the cursor over the menu	The system must show the appropriate message in the status bar of the navigator.	The status is not displayed correctly
7.	On screen Load	The company logo should be spelled correctly.	The Company logo is not spelled correctly.

Project Name: SS	Module / Screen Name: Manual Data Entry Module (User validation screen)
Tested By:	Test Conducted On:

S. No.	Test Condition	Expected Result	Actual Result
1.	Check Menu Bar and Menu bar items	They must be displayed clearly	Some of the items are overlapped
2.	On load of user validation screen and clicking ok without entering user name	Appropriate error message should be displayed	The error message is irrelevant.
3.	On Click of cancel from customer validation screen	The system must reset the form	The system displays an error message saying that that username and password are blank
4.	On clicking of the cancel buton	All the details should be discarded and the system must return to the main menu	The system proceeds to a blank screen **(SHOW STOPPER)**

Project Name: SS	Module / Screen Name: Manual Data Entry Module (Edit account detail screen)
Tested By:	Test Conducted On:

S. No.	Test Condition	Expected Result	Actual Result
1	On screen load	The CFI column must be able and must not be editable.	It is a text box.
2	On Click of cancel from customer validation screen	The system must take us to home page.	The system proceeds to a blank screen **(SHOW STOPPER)**

SYSTEM IMPLEMENTATION

7 System Implementation

This chapter gives a brief description of how the system is deployed in the actual Environment.

7.1 System Implementation

Before implementing this system in any environment it is necessary for us to customize the system for that client. We need to create databases in relation with the client. We will have to make change to UI, message formats and Resource bundle and text in relation to the client. We need to create appropriate error message and o/p formats pertaining to the client.

Before implementing the System, it's forced in to many severe testing phases. After the system clears all the tests, it's released for implementation. After the data has been initially set, the system is ready for use. The implementation type or the change over technique from the existing system is a step by process .

First a module in the part of the system is implemented and checked for suitability and the efficiency. If the end user related to the particular module is satisfied, the next step of implementation is processed with. That's modules related to the previous modules are implemented.

7.2 User Training

Training is given to all the particular users from the Client side. The training varies from user to user depending upon the information needed pertaining to the user. For example, the application users need help only on ad-hoc queries and how to take suggestions based upon the reports, whereas data entry operators need only information on how to key in suitable data.The WAP user's need to have training on the request formats and what each part of request is meant for.

CONCLUSION

The goal of the system is to revolutionize the wireless networking model. With the present technical world, industries providing various services to compete in the market and without this a progress setback to the organizations in this competing world. The basic idea of this project is to change the existing trend where networking is done on peer to peer basis, thus increasing the utilization of resources improving data integrity and security, thereby keeping in track the IT world and keeping pace with the competitors. The main power of the system is its use of latest technologies and a flexible solution that can be modified to fit in new requirements and types in the future. Since each of the modules exists as separate entities, any given module can be used as a plug-in module into another system with a similar need.

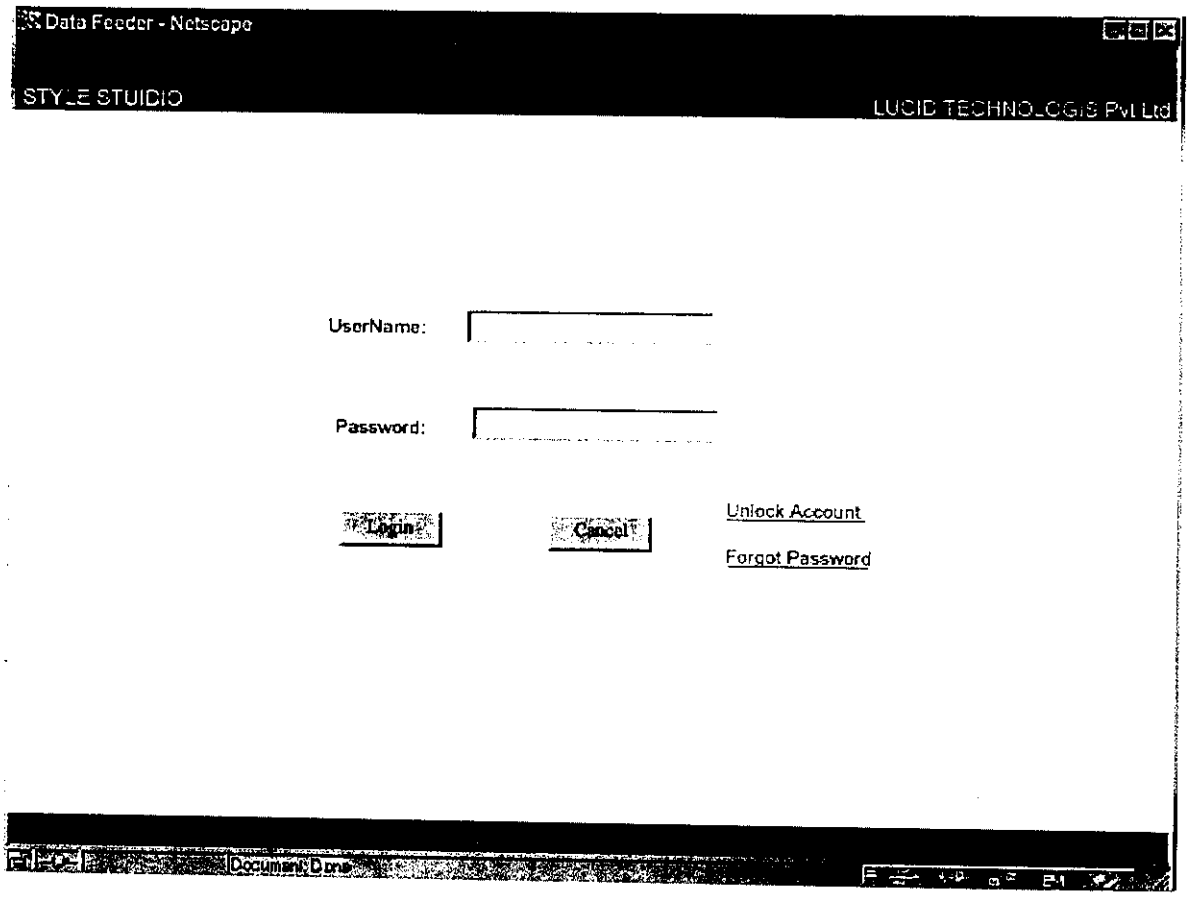
Future Enhancements

The core competence of this product lies in the factor that it is totally expandable and flexible to the latest and future technologies. The product has been developed with the present working condition and environment in mind. The Current environment is a fast growing area and new features, new technologies and different work styles are expected. Hence this software has been developed with near future needs in mind and it has appropriate slots for any future modifications. The product can be modified to support more media types like streaming and SVG with few more additions to the functionality. The flexible nature of the system and the technologies used make the system expandable and scalable.

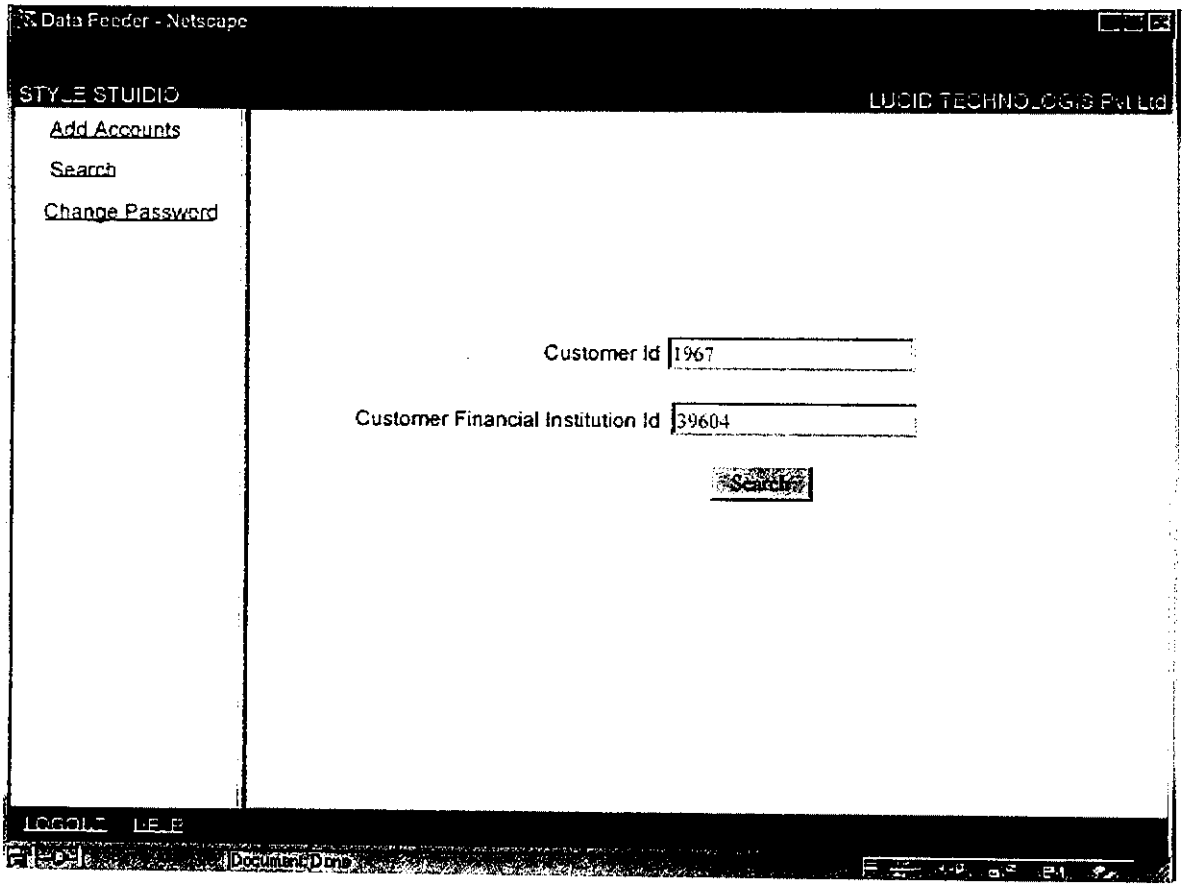
Future modifications though are limitless the immediate step towards future enhancement would be to make it PDA Compatible. Future enhancement in terms of customization, scalability , compatibility and technology has been made widely possible due to the style of design adopted and the strict adherence to W3C standard makes the system even more flexible for future enhancements.

APPENDIX

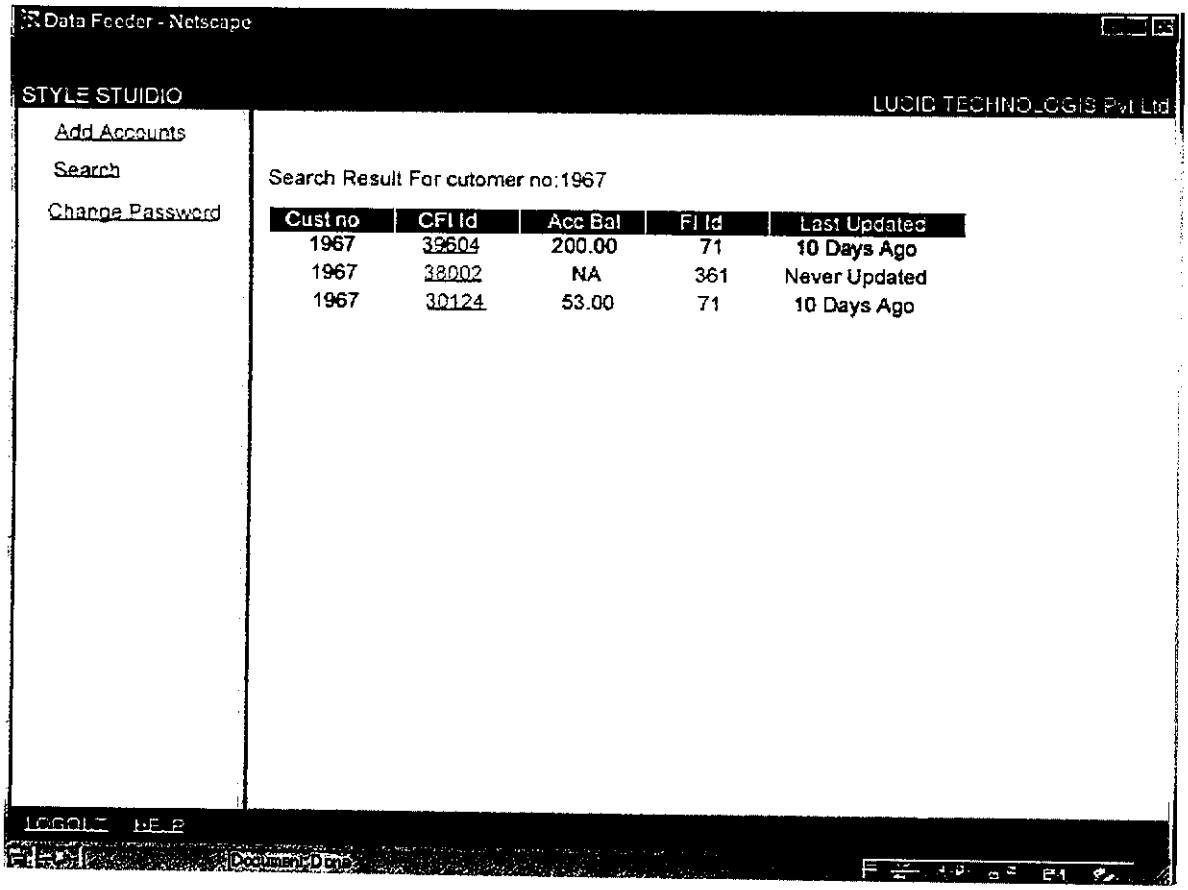
10.1 Screen Shots



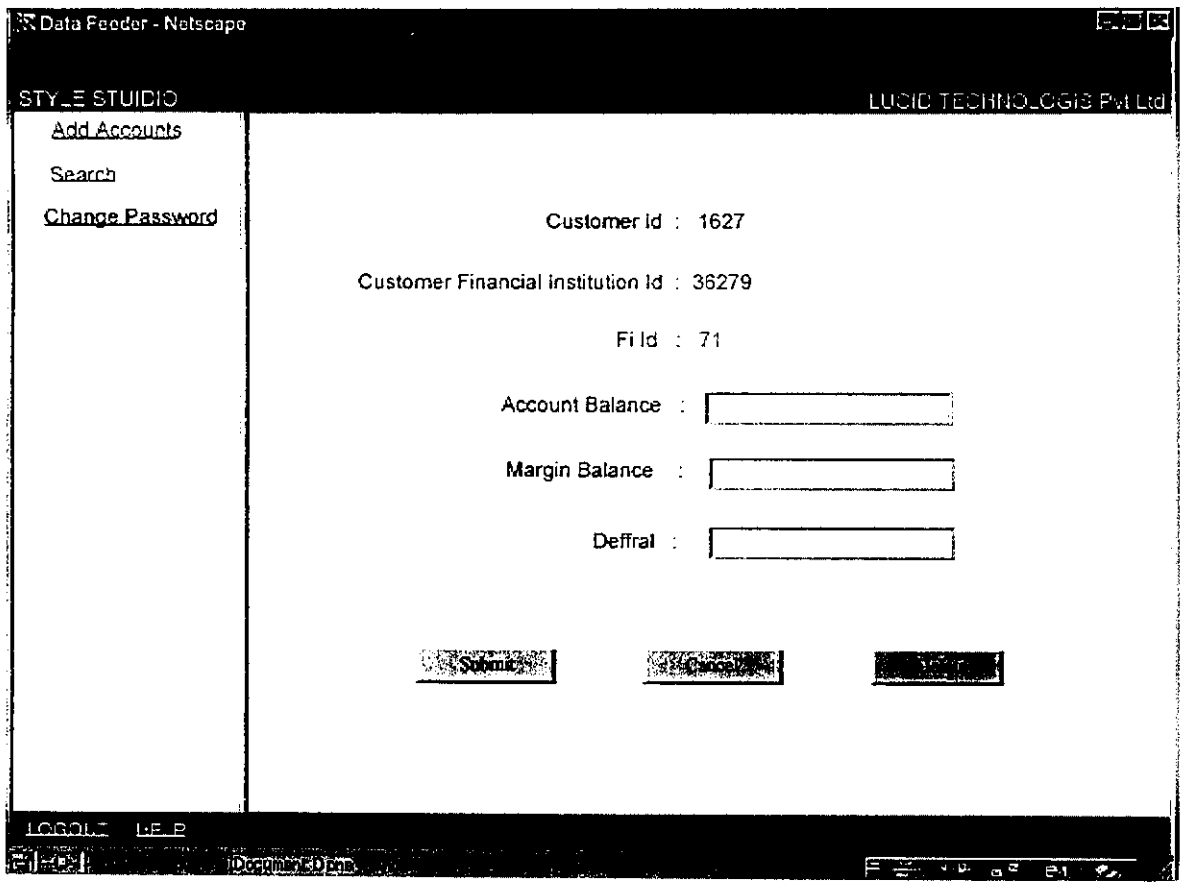
Login Screen



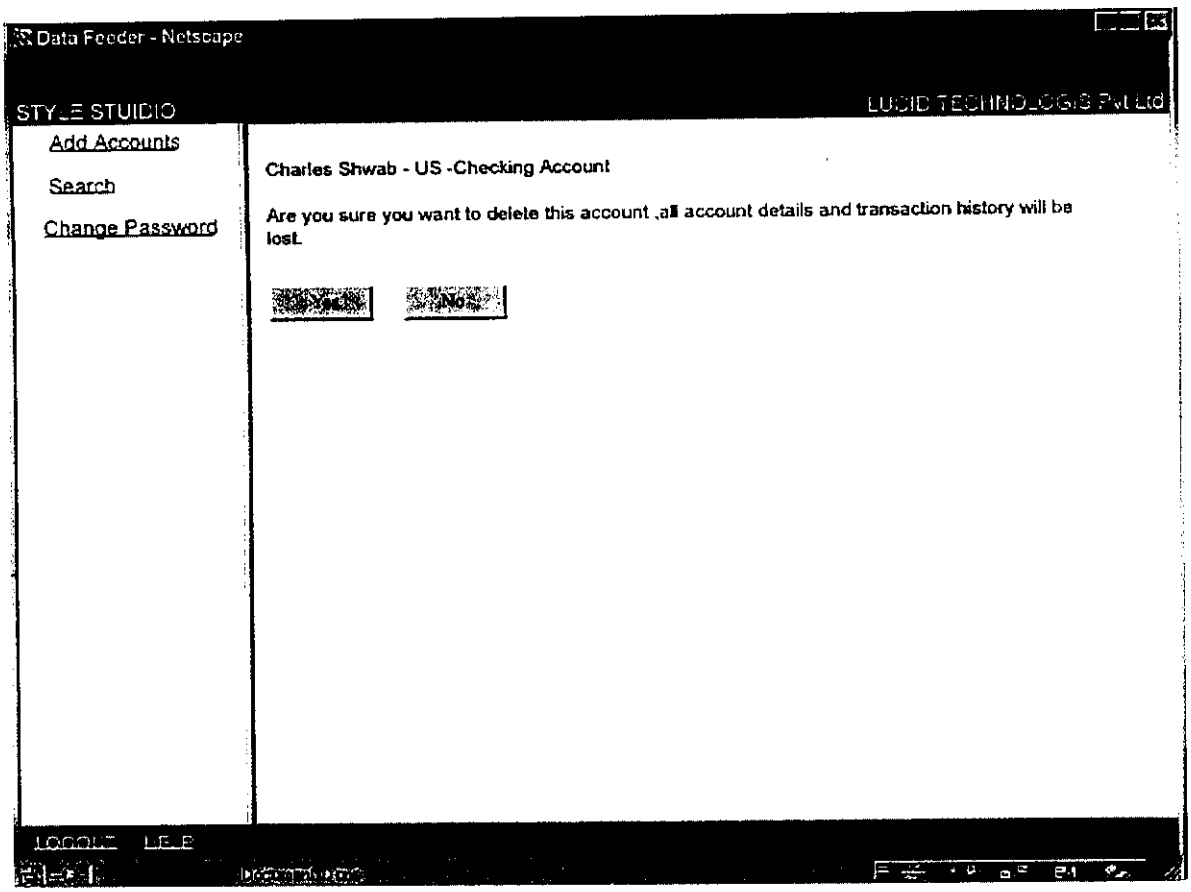
Search Screen



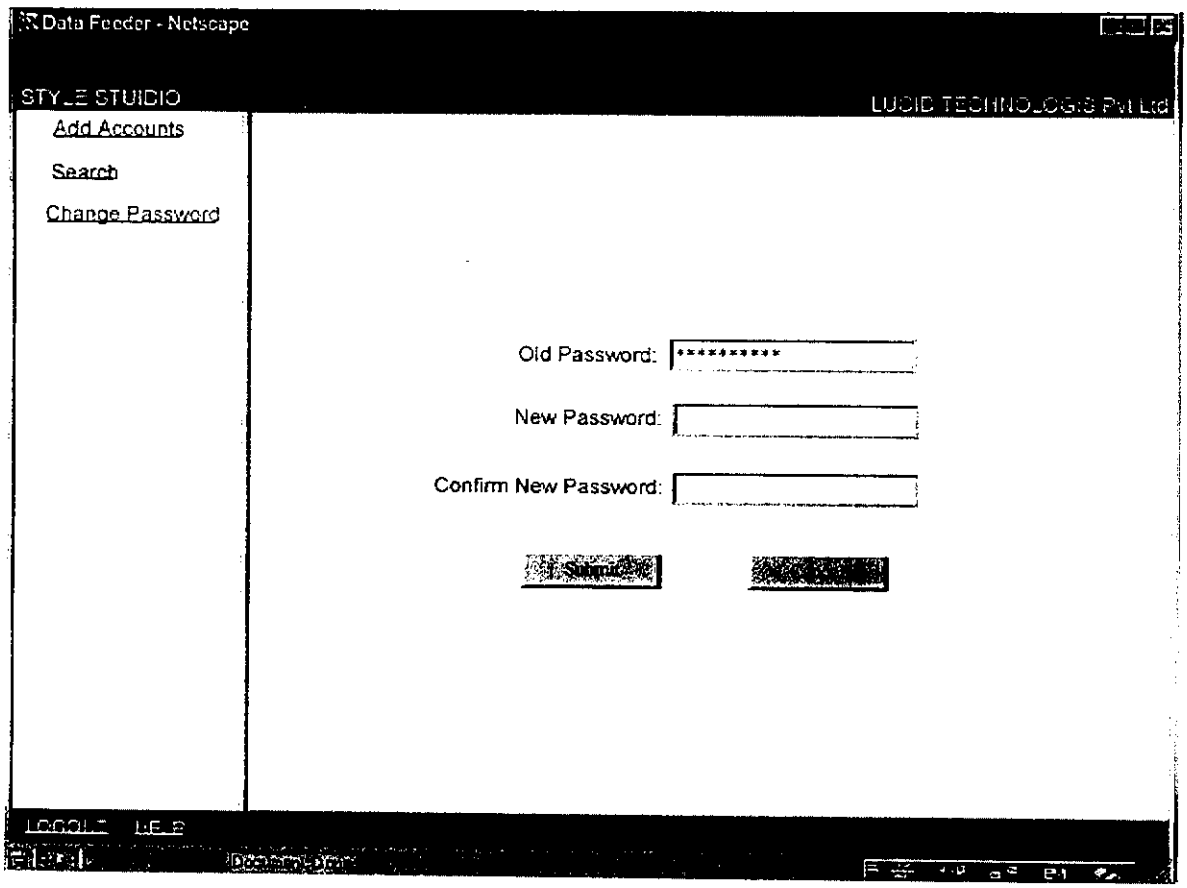
Search Result



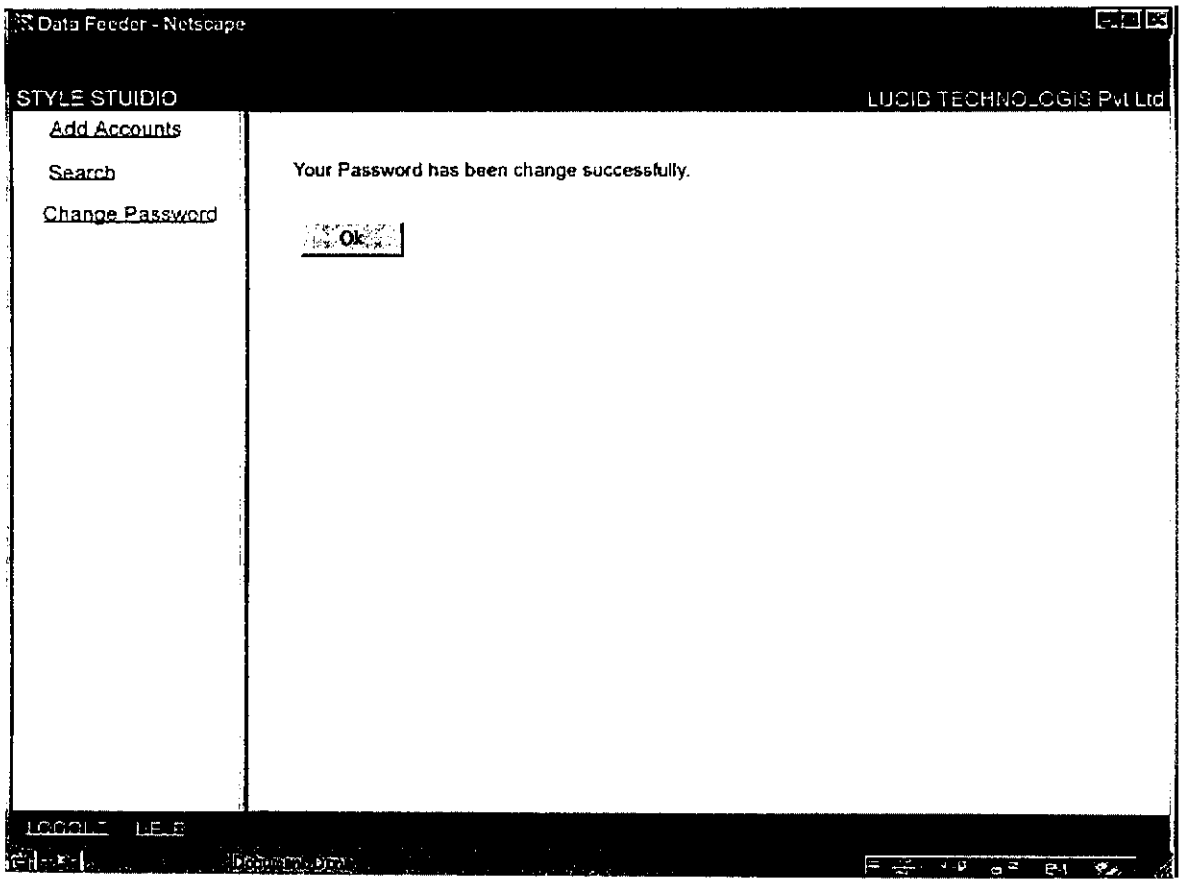
Add/Edit Data Screen



Confirm Data Deletion Screen



Change User Password



Viewing a Customer's complaint details

10.2 References

1. Java - Complete Reference,
   - Patrick Naughton and Herbert Schildt,Tata McGraw-hill, Third edition, 2000.
2. Oracle 8.0 - User Manual
   - Oracle Corporation Press , 2000
3. Java Servlets
   - Using Java Servlets, Mark C. Reynolds.
   - http://www.sun.com/markrey/servlets/index.htm
4. Software Engineering
   - Roger Pressman,Tata McGraw Hill edition,1998.
5. EJB Spec1.1
   - Vlaeda Materia and Mark Harper, Sun Microsystems Final Release, 1999.
6. Java Script
   - JavaScript Workshop, Laura Lemay and Michael G. Moncur,Prentice hall.
   - http://www.starlingtech.com/books/javascript/
   - http://www.developer.netscape.com/
7. JSP
   - Professional JSP ,Wrox Press.