CONSUMER ISSUE BOARD

By

K.SATHIESH KUMAR
(Registration Number: 71205621041)

Of

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641 006

A PROJECT REPORT
Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements
For the award of the degree

Of

MASTER OF COMPUTER APPLICATIONS

ANNA UNIVERSITY
Chennai 600 025

JULY 2008

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore-641006

DEPARTMENT OF COMPUTER APPLICATION

Bonafide Certificate

Certified that this project report titled **CONSUMER ISSUE BOARD** is the bonafide work of **Mr.K.SATHIESH KUMAR (Registration Number: 71205621041)** who carried out the research under my supervision. Certified further, that to the best of my Knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Supervisor 25/6/'08

Head of the Department

Submitted to Project and Viva Examination held on 01-07-2008.

Internal Examiner 1/7/08

External Examiner 1/7/08

IXIN TECHNOLOGY SOLUTIONS

aixin
MAKES THE GROWTH FASTER

30th May' 08

PROJECT COMPLETION CERTIFICATE

This is to certify that **Mr. K. SATHIESH KUMAR (Reg. no 71205621041)** of Kumaraguru College of Technology, has done his project "**CONSUMER ISSUE BOARD**" Chennai, at our organization from 20th December to 30th May08. During the project, he has successfully covered all the areas required for his project.

We wish him all the very best.

For Aixin Technology Solutions,

AIXIN TECHNOLOGY SOLUTIONS Chennai-15.

Manager, Human Resources

ABSTRACT

Consumer Issue Board (CIB) project provides the solution for whole sale dealers or any business enterprise, needs to keep the client information and their business entities related to the client.

The large scale business enterprise had a necessity to keep track of their client's records. It is also necessary to sustain the orders of clients and have to delivery them aptly. Enterprise also maintains the stock information, department details, branch details, workers profile etc.

CIB is a web application. The customers can contact the enterprise through internet by means of CIB web application. For example placing order, changing the order, checking the delivery status, requesting for service, checking service status, confirming service provided.

According to this web application, there are five kinds of users as follows

- Guest
- Admin
- Sales Manger
- Workers
- Dealers

Each kind of users has their own access permissions. To access this web application, users have to register themselves. Users will be provided with User ID. Using ID user can enter into the web application and perform operations. Three kinds of registration are available for Guest, Dealers and workers

.

This Project is developed using ASP.NET as Front-end and SQL Server as Back-end aims at proper management of the company by providing necessary Reports to the management.

ACKNOWLEDGEMENT

I would like to take this opportunity to say thank you to the people who have helped to make this project.

First I would like to express my heartfelt thanks to our principal **Dr.Joseph V.Thanikal**, B.E., M.E., Ph.D., PDF., CEPIT., for having given me the opportunity to do this project.

I express gratitude and thanks to **Dr.S.Thangasamy**, Ph.D., Dean, Computer Science and Engineering, for providing moral support towards this project work.

I express my deep sense of gratitude and profound thanks to **Dr.M.Gururajan**, Ph.D., HOD, Computer Application, for providing moral support towards this project work.

I tender my special thanks to **Mr. S.Hameed Ibrahim,** MCA, M.Phil Senior Lecturer and Project Coordinator, Department of Computer Applications for his constant encouragement.

I sincerely express my humble gratitude to **Ms.R.VeenaShree**, M.C.A., M.Phil.,Lecturer, Department of Computer Application, who has been my guide with valuable and holistic suggestions and extended kind of operation and encouragement to make this project.

I also express my deep sense of gratitude and profound thanks to **Mr.R.Arul kumar**, Project Manager, Aixin Technology Solutions, who has been my guide and gave valuable suggestions and encouragement.

I acknowledge my hearty thanks to all my beloved friends on their valuable co-operation in the proceeding of my work.

I wish to credit my special accordance to my parents for their encouragement and prayers to have a successful project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 ORGANIZATION PROFILE

ABOUT US:

Aixin Technology Solutions is the Software and IT services providing company which gathers the industry needs by representing major Technologies in the development and services both in the National and International level as well as in software, hardware, voice and non-voice support.

Aixin Technology Solutions deliver software products, ranging from enterprise project to Internet applications with enabled security Aixin envisions architects and delivers strategic and mission critical programs to support our customers in their business transformation initiatives. Our strength in architecting large and complex applications is valuable to customers trying to create agile IT systems. Our quality of work and the impact it creates is our self-governing benchmark.

WE FOCUS ON:

Aixin provides IT & verification and validation solutions to meet the business needs of small and medium enterprises. We provide comprehensive technology and web branding solutions spanning varied business domains and entrepreneurial ventures. Our solutions involve research, business process analysis and use of relevant technology.

Aixin is focused on leveraging information technology to help small and medium enterprises achieve their business goals. We do this by combining our technology strengths with our business analysis skills so that the application serves the needs of the target audience and is technologically robust.

OUR FACE:

There is a trend among software engineers to move an organization to another, never staying for the lifetime of a product, never caring about what happens after they leave. Then software projects fail and the money invested disappear in a pit.

This is not the case with Aixin Technology Solutions. The company is here to stay, and the employee turnover is zero. We believe that excellent products are delivered by people who care about their work. We aim to deliver the best possible software and be proud of it. We favor technical decisions as opposed to political ones. Our relationship with the customer grows naturally with the successful delivery of each new product.

OUR PROCESS:

The IT field is all about thought leadership. We invest in thought for you as Customers. We remain open, it helps us think better and differently. We believe in providing quality work every time we do it. Our psyche has been geared to produce quality irrespective of the client and the cost. A good job goes a long way in giving us more clients and more importantly we have a reputation to protect.

For us process is about a structure and working within the structure to provide decisive, effective and quality works. We respect our processes and are well aware of their true worth. We are a well coordinated team and work like a well-oiled machine. We work on a human scale and build long lasting partnerships with our clients. Our compactness provides us the needed room and flexibility to manage our future and independence. It also provides us a strong platform to safeguard our ethics and core values.

1.2 PROBLEM DEFINITION

OVERVIEW OF THE PROJECT:

Consumer Issue Board (CIB) project provides the solution for whole sale dealers or any business enterprise, needs to keep the client information and their business entities related to the client.

The large scale business enterprise had a necessity to keep track of their client's records. It is also necessary to sustain the orders of clients and have to delivery them aptly. Enterprise also maintains the stock information, department details, branch details, workers profile etc.

CIB is a web application. The customers can contact the enterprise through internet by means of CIB web application. For example placing order, changing the order, checking the delivery status, requesting for service, checking service status, confirming service provided. It has five types of users like Admin, Dealer, Sales manager, Worker and Guest user.

Guest user can view available stock items and make order. Admin has the responsibility of validating the user based on their registration, providing the security code for employee and maintains the overall work.

Dealers can view stock items, make order and cancel order. Canceling of order is available only for Dealers. Dealer can fix appointments through this web application. Using the calendar, dealer can fix up appointment with a concern person. This information is stored and potted as records to avoid clash.

Sales Manager maintains stock level detail in branch wise. Worker process service request post by client, deliver the item to client and generate the bill for order.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM ARCHITECTURE

The existing way of the article sharing is tedious. The study of existing system helps to determine which activities currently being performed should be continued in this new system. This step can be simple if the current system is well documented.

LIMITATION OF EXISTING SYSTEM:

- Large amount of clerical time is required.
- Data reliability and maintainability is difficult.
- Accessibility of accurate information is difficult.
- There is always delay in information about the suitable CIB.
- A qualitative work cannot be achieved.
- Inconsistent.

2.2 PROPOSED SYSTEM ARCHITECTURE

In order to overcome the difficulties in the existing system the "Consumer Issue Board System" is proposed. The proposed system is in such a way that it is easily operated. The site care will be taken to makes the pages attractive to the viewer and they are user-friendly screens. The screens in every page provide necessary links to every other page to make navigation through the system easy. Sufficient details are provided to the user.

This system entitled "Consumer Issue Board System" is aimed to provide a service to the Relationship Managers. This is developed for the benefit of the organization.

ADVANTAGES OF PROPOSED SYSTEM:

- Reduction of manual work
- Simple and fault free
- Flexible and easy to use
- User Friendly.
- Efficient
- Reliability

2.3 USER INTERFACE REQUIREMENTS

MODULES:

The System Development team grasps the system and breaking the it into several smaller systems. These smaller systems will be a part of the original system yet they will be independent in the sense that they will incorporate within them the major functionalities of the proposed system.

A software system is always divided into several subsystems which make it easier to develop and perform tests on the whole system. The subsystems are known as the modules and the process of dividing an entire system into subsystems is known as Decomposition.

ADMIN MODULE:

In this module admin provides access to all type of users. Admin can view all request post by the customer and validates customer on registration. Admin provides user code to user type of employee. Admin can also able to maintain the record of sales and order by viewing the details through online. The admin can provide the access to users to make change of their password and the forget password can also able to retrieve.

CUSTOMER MANAGEMENT MODULE:

The customer module will provide a brief description about the customer and dealer operations. This Module which provide a two user such that they are Dealer and Guest User.

This module helps dealer to place the order, cancel the order, and make an appointment with the concern person inside the Organization. In this module the Guest user can view the available stock and place the order and also make service for their purchased product.

COMPANY MANAGEMENT MODULE:

This module helps the users such that the sales manager and the worker to manage the organization and make more improvement. In this module the sales manager and the worker check availability of stock, reorder stock when reorder level is reached and view status of delivery, service request and then make a bill generation for particular order, delivering the stock on proposed date, delivering the stock after servicing.

REPORTING MODULE:

In the reporting module all the necessary reports that are needed to the management. The reports can directly printed using the IE menu option. This helps to get a hardcopy of the report. The Corresponding reports provided in this module are as

- Dealer Details Report.
- Guest Details Report.
- Stock Details Report.
- Order Details Report.
- Bill Generation Report.
- Delivery Details Report.

CHAPTER 3

DEVELOPMENT ENVIRONMENT

3.1 HARDWARE REQUIREMENTS

The hardware used for the development of the project is:

PROCESSOR	:	PENTIUM IV
RAM	:	1 GB
MONITOR	:	17" COLOR
HARD DISK	:	120 GB
FLOPPY DRIVE	:	1.44 MB
CDDRIVE	:	LG 52X
KEYBOARD	:	MULTIMEDIA
MOUSE	:	LOGITECH

3.2 SOFTWARE REQUIREMENTS

The software used for the development of the project is:

OPERATING SYSTEM	:	WINDOW XP
ENVIRONMENT	:	VISUAL STUDIO .NET
PLATFORM	:	ASP.NET
DATABASE	:	SQL SERVER
SCRIPTING LANGUAGE	:	C#

3.3 SOFTWARE DESCRIPTION

VISUAL STUDIO.NET

Visual Studio .NET is the rapid application development tool for C#. Visual Studio.NET offers complete integration with ASP.NET and enables to drag and drop server controls and design Web Forms as they should appear when user views them.

Some of the other advantages of creating C# applications in Visual Studio.NET are

- Visual Studio .NET is a Rapid Application (RAD) tool. Instead of adding each control to the Web Form programmatically, it helps to add these controls by using Toolbox, saving programming efforts.

- Visual Studio .NET supports custom and composite controls. Can create custom controls that encapsulate a common functionality that might need to use in a number of applications.

- Visual studio .NET does a wonderful job of simplifying the creation and consumption of Web Services. Much of the programmer-friendly stuff (creating all the XML-based documents) happens automatically, with out much effort on the programmer's side. Attribute-based programming is a powerful concept that enables Visual Studio .NET to automate a lot of programmer-unfriendly tasks.

.NET FRAMEWORK:

- The .NET Framework is the infrastructure for the new Microsoft .NET Platform.

- The .NET Framework is a common environment for building, deploying, and running Web applications and Web services.

- The .NET Framework contains a common language runtime and common class libraries –like ADO .NET, ASP.NET and Windows Forms to provide advanced standard services that can be integrated into a variety of computer systems.

- The .NET Framework provides a feature-rich application environment, simplified development and easy integration between a numbers of different developments languages.

- The .NET Framework is language neutral. Currently it supports C++, C#, Visual Basic, and Jscript (The modern version of JAVASCRIPT).

- Microsoft's Visual Studio.NET is a common development environment for the new .NET Framework.

COMMON LANGUAGE RUNTIME:

One of the design goals of .NET Framework was to unify the runtime engines so that all developers could work with a set of runtime services. The .NET Framework's solution is called the Common Language Runtime (CLR). The CLR provides capabilities such as memory management, security, and robust error handling to any language that work with the .NET Framework.

The CLR enables languages to inter operate with one another. Memory can be allocated by code written in one language and can be freed by code written in another language. Similarly, errors can be raised in one language and processed in another language.

.NET CLASS LIBRARIES:

The .Net Framework provides many classes that help developers re-use code. The .Net class libraries contain code for programming topics such as threading, file I/O, database support, XML parsing, and data structures, such as stacks and queues, this entire class library is available to any programming languages that support the .NET Framework.

Because all languages now support the same runtime, they can reuse any class that works with the .NET Framework. This means that any functionality available to one language will also be available to any other .NET language.

CREATING ASP.NET APPLICATIONS USING WEB FORMS

In the successor to ASP, ASP.NET, programming for the Web has been structured to be as similar as possible to development for Microsoft Windows, through a new type of interface called Web Forms. Web Forms, which is just another name for a Microsoft ASP.NET page, support a similar event-driven model to Windows Forms and to previous versions of Visual Basic.

ASP.NET

ASP.NET Web pages are used as programmable user interface for the Web application. An ASP.NET Web page presents information to the user in any browser or client device and implements application logic using server-side code.

ASP.NET Web page features are:

- Based on Microsoft ASP.NET technology, in which code that runs on the server dynamically generates Web page output to the browser or client device

- An ASP.NET Web page automatically renders the correct browser-compliant HTML for features such as styles, layout, and so on. Alternatively, you can design your ASP.NET Web pages to run on a specific browser such as Microsoft Internet Explorer 6 and take advantage of browser-specific features.

- Compatible with any language supported by the .NET common language runtime.

- As it built on the Microsoft .NET Framework, it provides all the benefits of the framework like managed environment, type safety, and inheritance.

COMPONENTS OF ASP.NET WEB PAGES

In ASP.NET Web pages, user interface programming is divided into two pieces: the visual component and the logic.

The visual element consists of a file containing static markup such as HTML or ASP.NET server controls or both. The ASP.NET Web page works as a container for the static text and controls that's wanted to display.

The logic for the ASP.NET Web page consists of code that you create to interact with the page. The code can reside either in a script block in the page or in a separate class. If the code is in a separate class file, this file is referred to as the code-behind file. The code in the code-behind file can be written in Visual Basic, Visual C#, Visual J#, or JScript.

ASP.NET Web pages are compiled into a dynamic-link library (.dll) file. The first time a user browses to the .aspx page, ASP.NET automatically generates a .NET class file that represents the page and then compiles it. The .dll file runs on the server and dynamically produces the HTML output for that page.

TRADITIONAL WEB PAGE CHALLENGES:

Web application programming has certain challenges. Some among them are:

- **Implementing a rich Web user interface:** It can be difficult and tedious to design and implement a user interface using basic HTML facilities, especially if the page has a complex layout, a large amount of dynamic content, and full-featured user-interactive objects.

- **Separation of client and server:** In a Web application, the client (browser) and server are different programs often running on different computers (and even on different operating systems). Consequently, the two halves of the application share very little information; they can communicate, but typically exchange only small chunks of simple information.

- **Stateless execution:** When a Web server receives a request for a page, it finds the page, processes it, sends it to the browser, and then discards all page information. If the user requests the same page again, the server repeats the entire sequence, reprocessing the page from scratch. If an application needs to maintain information about a page, its stateless nature can become a problem.

- **Unknown client capabilities**: Web applications are accessible to many users using different browsers. Browsers different capabilities, makes it difficult to create an application that will run equally well on all of them.

- **Complications with data access**: Reading from and writing to a data source in traditional Web applications can be complicated and resource-intensive.

- **Complications with scalability**: In many cases, Web applications designed with existing methods fail to meet scalability goals due to the lack of compatibility between the various components of the application. This is often a common failure point for applications under a heavy growth cycle.

CHALLENGES OVERCOME BY ASP.NET:

ASP.NET Web pages and the ASP.NET page framework address these challenges in the following ways:

- **Consistent object model:** The ASP.NET page framework presents an object model that enables programmer to think of forms as a unit, not as client and server unit. It includes the ability to set properties for page elements and respond to events. In addition, programmer can use server controls like controls in a client application and not have to think about how to create the HTML to present and process the controls.

- **Event-driven programming model:** ASP.NET Web pages bring to Web applications the familiar model of writing event handlers for events that occur on either the client or server. The ASP.NET page framework abstracts this model in such a way that the underlying mechanism of capturing an event on the client, transmitting it to the server, and calling the appropriate method is all automatic and invisible. The result is, easily written code structure that supports event-driven development.

- **State management:** The ASP.NET page framework automatically handles the task of maintaining the state of web page and its controls. This is accomplished without heavy use of server resources and can be implemented with or without sending cookies to the browser.

- **Browser-independent application:** The ASP.NET page framework enables to create all application logic on the server and eliminating the need of explicitly coding it for different browsers.

ADDING FUNCTIONALITY WITH INHERITANCE:

One of the more powerful features of object-oriented programming lies in inheritance. The concept of inheritance stems from the fact that complex objects are extensions of simple ones. For example, the Hummer is an extension of a standard SUV; a standard SUV is an extension of a standard pickup truck; a pickup truck is an extension of a standard automobile. Inheritance makes design and implementation much easier because it gives designers and engineers a starting point. That is, when building a Hummer, the engineers already have the base components done for them: they don't need to worry about the dynamics of the internal combustion engine, or the computer system that works the fuel injector.

Since .NET is a fully object-orientated platform, it should come as no surprise that the ASP.NET object model is ripe with possibilities for extension via inheritance. Just look at the provided ASP.NET server controls. All server controls are extended, either directly or indirectly, from the System.Web.UI.Control class. This class provides the base functionality all ASP.NET server controls need to perform in the life cycle of an ASP.NET Webpage. This class is refined further with the System.Web.UI .WebControls.WebControl class, which adds style information, such as background color, fore color, font information, and such, to the base properties defined by the Control class.

In a similar vein, we can take an existing ASP.NET server control class, such as the Calendar or Data Grid or Textbox, and create a new class derived from this base class. Automatically, without having to write a single line of code, this inheritance gives our new server control all of the functionality provided in the base class.

Therefore, to add new functionality we only need to write the code necessary for said new functionality - and with that little addition, we'll have a fully robust, useable Web control similar to the base control in all ways except now that its functionality has been extended.

STEP 1: INITIAL REQUEST

The web client requests the main application URL, shown in **Figure**

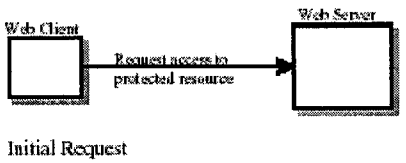


Figure 3.3.1 Initial Request

Since the client has not yet authenticated itself to the application environment, the server responsible for delivering the web portion of the application detects this and invokes the appropriate authentication mechanism for this resource.

STEP 2: INITIAL AUTHENTICATION

The web server returns a form that the web client uses to collect authentication data (for example, username and password) from the user. The web client forwards the authentication data to the web server, where it is validated by the web server, as shown.

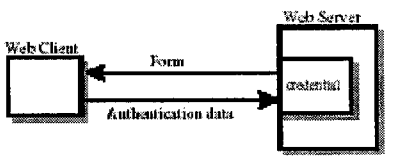


Figure 3.3.2 Initial Authentication

The validation mechanism may be local to the server, or it may leverage the underlying security services. On the basis of the validation, the web server sets a credential for the user.

STEP 3: URL AUTHORIZATION

The credential is used for future determinations of whether the user is authorized to access restricted resources it may request. The web server consults the security policy (derived from the deployment descriptor) associated with the web resource to determine the security roles that are permitted access to the resource. The web container then tests the user's credential against each role to determine if it can map the user to the role.

The web server's evaluation stops with an "is authorized" outcome when the web server is able to map the user to a role. A "not authorized" outcome is reached if the web server is unable to map the user to any of the permitted roles.

STEP 4: FULFILLING THE ORIGINAL REQUEST

If the user is authorized, the web server returns the result of the original URL request, as shown.

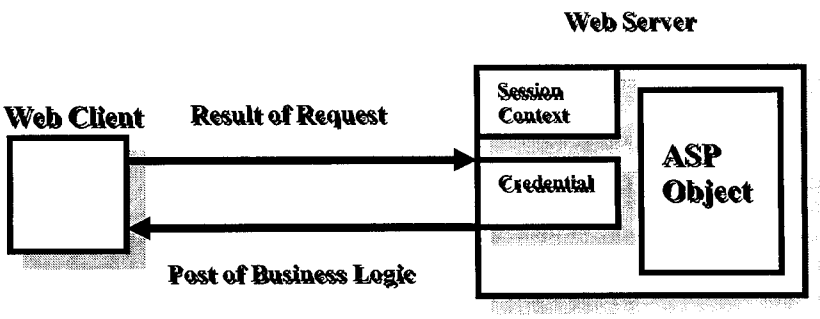


Figure 3.2.3 Fulfill the original Request

ADO.NET

The .NET Framework includes a new data access technology named as ADO.NET. It provides consistent access to data sources such as Microsoft SQL Server, as well as data sources exposed via OLE DB and XML. Data-sharing consumer applications can use ADO.NET to connect to these data sources and retrieve, manipulate, and update data.

ADO.NET includes .NET data providers for connecting to a database, executing commands, and retrieving results. Those results are either processed directly, or placed in an ADO.NET Dataset object in order to be exposed to the user in an ad-hoc manner, combined with data from multiple sources, or remote between tiers. The ADO.NET Dataset object can also be used independently of a .NET data provider to manage data local to the application or sourced from XML.

SUMMARY

While the .NET platform provides a powerful general-purpose foundation for many types of applications, it does meet all of the needs of web development teams. Achieving productivity in web development requires a specialized development platform. Such a platform must be tailored to the skills and needs of the developers building the applications. It must support the unique requirements of web development by providing specialized services for Web applications. And lastly, it must meet the needs of IT management, enabling the IT organization to support the business through the timely and cost-effective delivery of new solutions.

SQL SERVER:

STRUCTURED QUERY LANGUAGE (SQL):

SQL (pronounced SEQUEL) is the programming language that defines and manipulates the database. SQL databases are relational databases, which mean data is stored in a set of simple relations. A database can have one or more tables. Each table has columns and rows. Oracle stores each row of a database table containing data for less than 256 columns as one or more row pieces. A table that has an employee database, for example, can have a column called employee number and each row in that column is an employee's employee number.

Data can be defined and manipulated in a table with SQL statements. SQL's data definition language (DDL) statements are used to define data. DDL statements include statements for creating and altering databases and tables.

Update, delete, or retrieve a data in a table are done by SQL's data manipulation language (DML). DML statements include statements to alter and fetch data. The most common SQL statement is the `SELECT` statement, which retrieves data from the database.

In addition to SQL statements, the Oracle server has a procedural language called PL/SQL. PL/SQL enables programmers to program SQL statements. It lets user control the flow of a SQL program, use variables, and write error-handling procedures.

SQL STATEMENTS:

All operations on the information in an Oracle database are performed using SQL **statements**. A SQL statement is a string of SQL text that is given to Oracle to execute. A statement must be the equivalent of a complete SQL **sentence**, as in:

```
SELECT ename, deptno FROM emp;
```

Only a complete SQL statement can be executed, whereas a **sentence fragment**, such as the following, generates an error indicating that more text is required before a SQL statement can run:

SELECT ename

A SQL statement can be thought of as a very simple, but powerful, computer program or instruction. SQL statements are divided into the following categories:

- Data definition language (DDL) statements
- Data manipulation language (DML) statements
- Transaction control statements
- Session control statements
- System control statements
- Embedded SQL statements

DATA DEFINITION LANGUAGE (DDL) STATEMENTS:

Data definition language statements define, maintain, and drop schema objects when they are no longer needed. DDL statements also include statements that permit a user to grant other users the **privileges**, or rights, to access the database and specific objects within the database.

DATA MANIPULATION LANGUAGE (DML) STATEMENTS:

Data manipulation language statements manipulate the database's data. For example, querying, inserting, updating, and deleting rows of a table are all DML operations. Locking a table or view and examining the execution plan of an SQL statement are also DML operations.

TRANSACTION CONTROL STATEMENTS:

Transaction control statements manage the changes made by DML statements. They enable the user or application developer to group changes into logical transactions. Examples include `COMMIT`, `ROLLBACK`, and `SAVEPOINT`.

SESSION CONTROL STATEMENTS:

Session control statements let a user control the properties of his current session, including enabling and disabling roles and changing language settings. The two session control statements are `ALTER SESSION` and `SET ROLE`.

SYSTEM CONTROL STATEMENTS:

System control statements change the properties of the Oracle server instance. The only system control statement is `ALTER SYSTEM`. It lets user change such settings as the minimum number of shared servers, to kill a session, and to perform other tasks.

EMBEDDED SQL STATEMENTS:

Embedded SQL statements incorporate DDL, DML, and transaction control statements in a procedural language program (such as those used with the Oracle precompilers). Examples include `OPEN`, `CLOSE`, `FETCH`, and `EXECUTE`.

CHAPTER 4

SYSTEM DESIGN

4.1 ELEMENTS OF DESIGN

Design is multi-step process that focuses on data structure software architecture, procedural details, algorithms and interface between modules. The design process also translates the requirements into the presentation of software that can be accessed for quality before coding begins.

Computer software design changes continuously as new methods; better analysis and broader understanding evolved. Software Design is at relatively early stage in its revolution.

Therefore, Software Design methodology lacks the depth, flexibility and quantitative nature that are normally associated with more classical engineering disciplines. However techniques for software designs do exist, criteria for design qualities are available and design notation can be applied.

4.1.1 INPUT DESIGN

The user interface design is very important for any application. The interface design describes how the software communicates within itself, to system that interpreted with it and human who use it. The input design is the process of converting the user oriented inputs into the computer based format. The data is fed into the system using simple interactive forms. The forms have been supplied with messages so that user can enter data without facing any difficulty. The data is validated wherever it requires in the project. This ensures that only the correct data have been incorporated into the system. The input design requirements such as user friendly, consistent format and interactive dialogue for giving the right message and help for the user at right time are also considered for the development of this project.

In the project, the Billing Page is made with several easy to use options. For example, in the billing form the appropriate order no can be viewed and the corresponding item amount can be calculated by giving necessary inputs.

4.1.2 OUTPUT DESIGN

The objective of the output design is to convey the information of all the past activities, current status and to emphasize important events. The output generally refers to the results and information that is generated from the system. Outputs from computers are required primarily to communicate the results of processing to the users.

In this project various output windows are:

- Stock Details Window.
- Confirm Order Details Window.
- Bill Details Window.
- Delivery Details Window.

4.1.3 DATABASE DESIGN

The database design is a must for any application developed especially more for the data store projects. The database is normalized so as to avoid redundant data. Normalization reduces the wastage of valuable memory space. The database system must give assurance for security of information, despite the system crashes due to unauthorized access.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general theme behind a database is to integrate all the information. The general objective of database design is to make the data access easy, inexpensive and flexible to the user.

The main objectives of designing a database are:

- Data integration
- Data integrity
- Data independence

TABLE DESIGN

TABLE NAME: LOGIN

FIELD NAME	FIELD TYPE	CONSTRAINTS	DESCRIPTION
Uname	Varchar(50)	Not Null	User Name
PWord	Varcahr(50)	Not Null	Password
UType	Varchar(50)	Not Null	User Type

TABLE 4.1.3.1 LOGIN

TABLE NAME: EMPREGISTRATION

FIELD NAME	FIELD TYPE	CONSTRAINTS	DESCRIPTION
EmpCode	Numeric	Not Null	Employee Code
EmpName	Varcahr(50)	Not Null	Employee Name
EmpAddrs	Varchar(50)	Not Null	Employee Address
EmpCity	Varchar(50)	Not Null	Employee City
EmpState	Varchar(50)	Not Null	Employee State
EmpCountry	Varchar(50)	Not Null	Employee Country
Pincode	Numeric	Not Null	Pincode
EmpPhoneNo	Varchar(50)	Not Null	Employee Phone Number
EmpMobile	Varchar(50)	Not Null	Employee Mobile Number
EmpEmailId	Varchar(50)	Not Null	Employee Email ID
PWord	Varchar(50)	Not Null	Employee Password
UType	Varchar(50)	Not Null	User Type

TABLE 4.1.3.2 EMP REGISTRATION

TABLE NAME: GUEST REGISTRATION

FIELD NAME	FIELD TYPE	CONSTRAINTS	DESCRIPTION
GuestCode	Numeric	Not Null	Guest Code
GuestName	Varcahr(50)	Not Null	Guest Name
GuestAddrs	Varchar(50)	Not Null	Guest Address
GuestCity	Varchar(50)	Not Null	Guest City
GuestState	Varchar(50)	Not Null	Guest State
GuestCountry	Varchar(50)	Not Null	Guest Country
Pincode	Numeric	Not Null	Pincode
GuestPhoneNo	Varchar(50)	Not Null	Guest Phone Number
GuestMobile	Varchar(50)	Not Null	Guest Mobile Number
GuestEmailId	Varchar(50)	Not Null	Guest Email ID
PWord	Varchar(50)	Not Null	Guest Password
Gid	Varchar(50)	Not Null	User ID
BName	Varchar(50)	Not Null	Bank Name
BAccNo	Varchar(50)	Not Null	Bank Account Number
BBranchName	Varchar(50)	Not Null	Bank Branch Name

TABLE 4.1.3.3 GUEST REGISTRATION

TABLE NAME: BILL MASTER

FIELD NAME	FIELD TYPE	CONSTRAINTS	DESCRIPTION
BillNo	Numeric	Not Null	Bill Number
CusNo	Numeric	Not Null	Customer Number
UType	Varchar(50)	Not Null	User Type
BillDate	Varchar(50)	Not Null	Bill Date
TotAmt	Numeric	Not Null	Total Amount
Status	Numeric	Not Null	Status of Bill

TABLE 4.1.3.4 BILLMASTER

TABLE NAME: DEALER REGISTRATION

FIELD NAME	FIELD TYPE	CONSTRAINTS	DESCRIPTION
DelCode	Numeric	Not Null	Dealer Code
DelName	Varcahr(50)	Not Null	Dealer Name
DelAddrs	Varchar(50)	Not Null	Dealer Address
DelCity	Varchar(50)	Not Null	Dealer City
DelState	Varchar(50)	Not Null	Dealer State
DelCountry	Varchar(50)	Not Null	Dealer Country
Pincode	Numeric	Not Null	Pincode
DelPhoneNo	Varchar(50)	Not Null	Dealer Phone Number
DelMobile	Varchar(50)	Not Null	Dealer Mobile Number
DelEmailId	Varchar(50)	Not Null	Dealer Email ID
PWord	Varchar(50)	Not Null	Dealer Password
Did	Varchar(50)	Not Null	User ID
BName	Varchar(50)	Not Null	Bank Name
BAccNo	Varchar(50)	Not Null	Bank Account Number
BBranchName	Varchar(50)	Not Null	Bank Branch Name

TABLE 4.1.3.5 DEALER REGISTRATION

TABLE NAME: BRANCH MASTER

FIELD NAME	FIELD TYPE	CONSTRAINTS	DESCRIPTION
BranchId	Numeric	Not Null	Branch ID
BranchName	Varcahr(50)	Not Null	Branch Name
BranchAddrs	Varchar(50)	Not Null	Branch Address
BranchlCity	Varchar(50)	Not Null	BranchCity
Pincode	Numeric	Not Null	Pincode
BranchPhoneNo	Varchar(50)	Not Null	BranchPhone Number
BranchFaxNo	Varchar(50)	Not Null	Branch Mobile Number
RegTime	Varchar(50)	Not Null	Registration Time

TABLE 4.1.3.6 BRANCH REGISTRATION

TABLE NAME: DELIVERY MASTER

FIELD NAME	FIELD TYPE	CONSTRAINTS	DESCRIPTION
Delivery No	Numeric	Not Null	Delivery Number
Bill No	Varcahr(50)	Not Null	Bill Number
Delivery Addrs	Varchar(50)	Not Null	Delivery Address
Status	Varchar(50)	Not Null	Status
DeliverDate	Numeric	Not Null	DeliverDate
Delivered Date	Varchar(50)	Not Null	DeliveredDate

TABLE 4.1.3.7 DEALER MASTER

TABLE NAME: ORDER MASTER

FIELD NAME	FIELD TYPE	CONSTRAINTS	DESCRIPTION
Order No	Numeric	Not Null	Order Number
UID	Varcahr(50)	Not Null	User ID
DepID	Varchar(50)	Not Null	Department ID
ItemID	Varchar(50)	Not Null	Item ID
Quantity	Numeric	Not Null	Quantity
TotalAmt	Varchar(50)	Not Null	Total Amount
DTime	Varchar(50)	Not Null	Date
BillNo	Numeric	Not Null	Bill Number
ItemName	Varchar(50)	Not Null	Item Name

TABLE 4.1.3.8 ORDER MASTER

TABLE NAME: SERVICE MASTER

FIELD NAME	FIELD TYPE	CONSTRAINTS	DESCRIPTION
Service No	Numeric	Not Null	Service Number
UID	Numeric	Not Null	User ID
Service	Varchar(50)	Not Null	Type of Service
ServiceDate	Varchar(50)	Not Null	Service Date
BranchID	Varchar	Not Null	Branch ID
Status	Numeric	Not Null	Status

TABLE 4.1.3.9 SERVICE MASTER

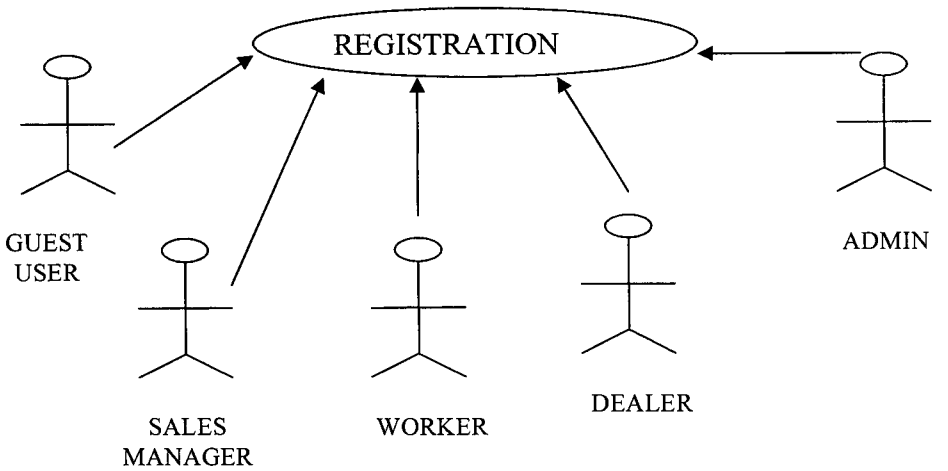
TABLE NAME: STOCK MASTER

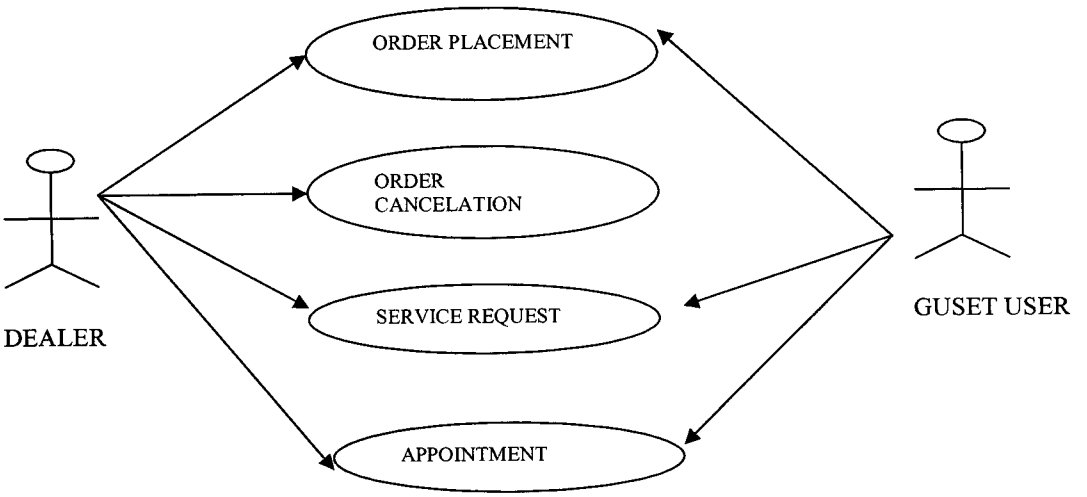
FIELD NAME	FIELD TYPE	CONSTRAINTS	DESCRIPTION
StockID	Numeric	Not Null	Stock ID
StockName	Varchar	Not Null	Stock Name
BrandName	Varchar(50)	Not Null	Brand Name
Quantity	Numeric	Not Null	Quantity
BranchID	Varchar	Not Null	Branch ID
UnitPrice	Numeric	Not Null	Unit Price

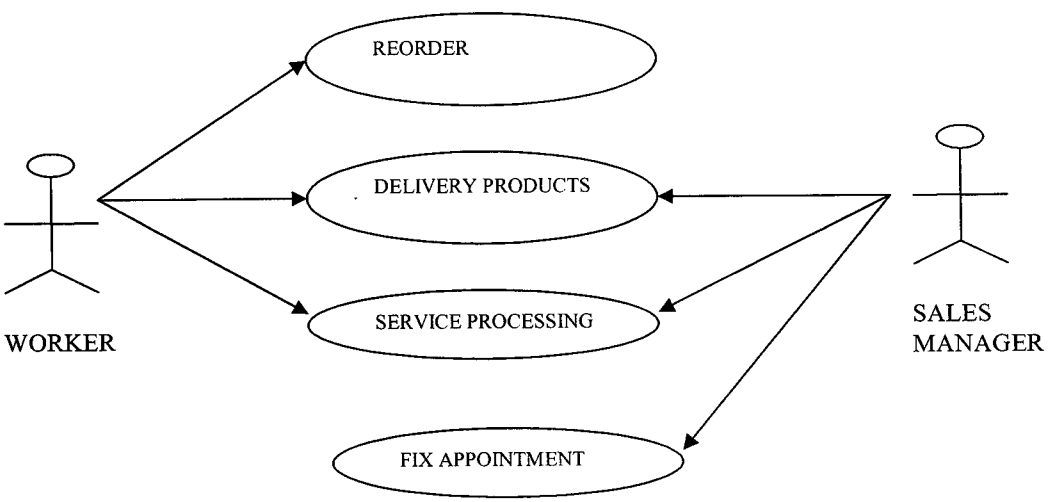
TABLE 4.1.3.10 STOCK MASTER

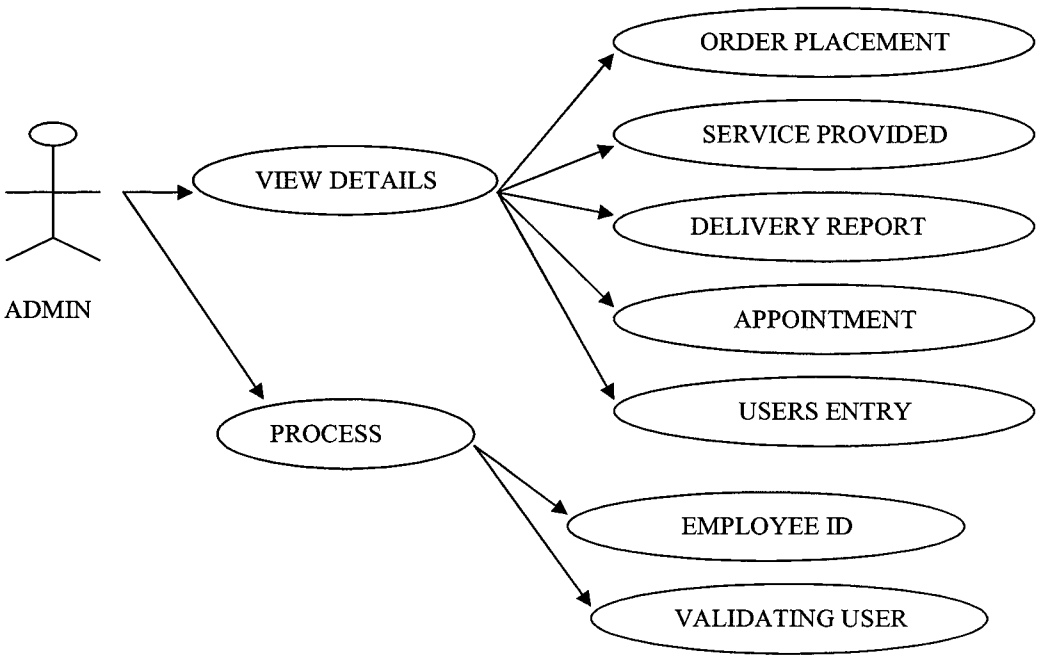
4.2 USE CASE DIAGRAM

A use case diagram is a type of behavioral diagram defined by the Unified Modeling Language (UML) created from a use case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals represented as use cases and any dependencies between those use cases.




ORDER PLACEMENT
ORDER CANCELATION
SERVICE REQUEST
APPOINTMENT
DEALER
GUSET USER


REORDER
DELIVERY PRODUCTS
SERVICE PROCESSING
FIX APPOINTMENT
WORKER
SALES MANAGER



4.3 DATA FLOW DIAGRAM

A Data Flow Diagram is used to define the flow of the system and the resources such as information. It is the way of expressing system requirements in a graphical manner. It is also known as bubble chart. It consists of a series of bubbles joined by lines. The bubbles represent data transformation and the lines represent data flow in the system.

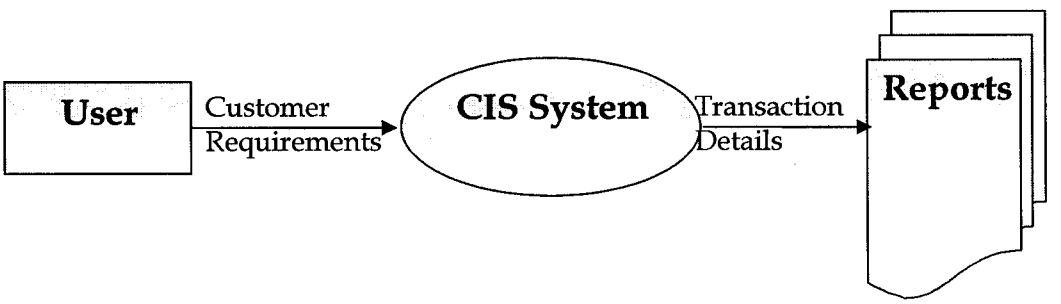
Data Flow Diagram is directed graphs in which the nodes specify processing activities and the arcs that specify data items transmitted between processing nodes. Like flow charts, data flow diagram can be used at any desired level of abstraction. A data flow diagram can be used to represent data flow between individual statements or block statements in a routine, data flow sequential routine between concurrent processes or data in a distributed computing system, where each node represents a geographically remote

processing unit. Unlike flowcharts, data flow diagrams do not indicate decision logic or condition under which various processing nodes in the diagram being activated.

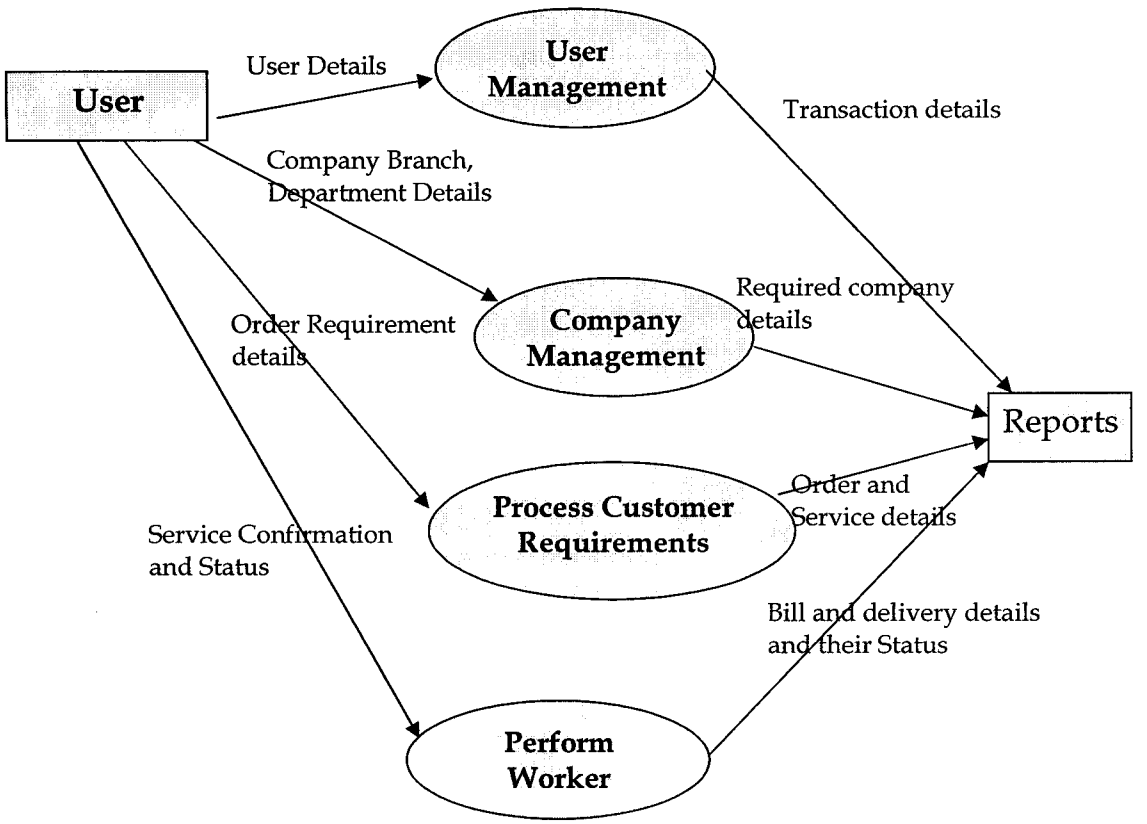
Data flow diagrams are excellent mechanisms for communicating with customers during requirement analysis; also they are widely used for representation of external and top-level internal design specifications.

The Data flow diagrams may be used to represent a system or software at any level of abstraction. DFD's may be partitioned into levels that represent increasing information flow and functional details.

4.3.1 LEVEL-0 DFD



4.3.2 LEVEL-1 DFD



CHAPTER 5

ARCHITECTURAL DESIGN

5.1 MODULE DESIGN

The System Development team grasps the system and breaking the it into several smaller systems. These smaller systems will be a part of the original system yet they will be independent in the sense that they will incorporate within them the major functionalities of the proposed system.

A software system is always divided into several subsystems which make it easier to develop and perform tests on the whole system. The subsystems are known as the modules and the process of dividing an entire system into subsystems is known as Decomposition.

1. Admin Module:

In this module admin provides access to all type of users. Admin can view all request post by the customer and validates customer on registration. Admin provides user code to user type of employee. Admin can also able to maintain the record of sales and order by viewing the details through online. The admin can provide the access to users to make change of their password and the forget password can also able to retrieve.

2. Customer Management Module:

The customer module will provide a brief description about the customer and dealer operations. This Module which provide a two user such that they are Dealer and Guest User.

This module helps dealer to place the order, cancel the order, and make an appointment with the concern person inside the Organization. In this module the Guest

user can view the available stock and place the order and also make service for their purchased product.

3. Company Management Module:

This module helps the users such that the sales manager and the worker to manage the organization and make more improvement. In this module the sales manager and the worker check availability of stock, reorder stock when reorder level is reached and view status of delivery, service request and then make a bill generation for particular order, delivering the stock on proposed date, delivering the stock after servicing.

4. Reporting Module:

In the reporting module all the necessary reports that are needed to the management. The reports can directly printed using the IE menu option. This helps to get a hardcopy of the report. The Corresponding reports provided in this module are as

- Dealer Details Report.
- Guest Details Report.
- Stock Details Report.
- Order Details Report.
- Bill Generation Report.
- Delivery Details Report.

CHAPTER 6

SYSTEM TESTING AND IMPLEMENTATION

6.1. OBJECTIVE OF TESTING

The objective of testing is to prove that there are no errors in the software. This is extremely difficult since developer cannot prove to be hundred percent accurate. Therefore the most useful and practical approach is with the understanding that testing is the process of executing a program with explicit intention of finding errors and check for the basic flow of the process.

Testing has its own cycle. The testing process begins with the product requirements phase and from there parallels the entire development process. In other words for each phase of the development process there is an important testing activity. Successful testing requires a systematic approach. It requires focusing on the basic critical factors: planning, project control, risk management, inspections, measurement, tools, organization and professionalism.

6.2 TYPES OF TESTING

6.2.1 SYSTEM TESTING

Testing is a set of activities that can be planned in advance and conducted systematically. A number of testing strategies have been proposed; in literature all provide the software developer with the template for testing and having the following generic characteristics.

- Testing begins at the component level and works outward towards the integration of the entire computer based system.

- The developer of the software conducts testing and for large products an independent test group may be used.
- Testing and debugging are different activities but debugging may be accommodated in any testing strategy.

6.2.2 SOFTWARE TESTING TECHNIQUES:

The test case design methods applied are

- White Box Testing
- Black Box Testing

WHITE BOX TESTING:

Using this testing method it was assured that all the independent paths were exercised at least once. All the logical decisions on their true and false side were executed. All loops were executed at their boundaries.

BLACK BOX TESTING:

Using this testing technique, incorrect and missing functions were identified and corrected, incorrect information, interfacing errors; performance errors, initialization errors and termination errors were also found using this technique.

6.2.2.1 SOFTWARE TESTING STRATEGIES:

A strategy for the software testing integrates software test case design techniques into well planned series of steps that result in the successful construction of software. Any testing strategy must incorporate Test Planning, Test Case Design, Resultant Data Collection and Evaluation. The different levels of testing are:

INTEGRATION TESTING:

This kind of testing is a systematic testing for constructing tests to uncover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. The system underwent a series of Integration tests that recorded smooth transmission of data from one module to the other. The bottom up approach was applied.

In this project the developed system is tested after integrating various modules together, and the detected errors were corrected.

VALIDATION TESTING:

Validation testing is carried out to verify whether the software functions works in a manner that is expected by the customer. So alpha Testing was done to ensure validity.

UNIT TESTING:

This kind of testing is to verify the smallest unit of the software module. This is also known as "Module Testing". This test is carried out during the programming stage. This test ensures the expected output from each of the module. The modules including admin, dealer, worker, sales manager and guest user. Exceptions have been handled and appropriate Error messages have been given in each module so as to avoid abnormal termination of the program. The unit testing considers the actions that were taken into account is as follows:

- Interfacing errors
- Integrity of local data structures.
- Boundary conditions.
- Independent paths.
- Error handling paths

TEST CASES

S.No	Test Case Name	Test Case Description	Expected Result	Actual Result	Test Script Status
1	User Name	Enter null string in "User Name" Textbox	"User Name" cannot be empty. Should display message to user	"User Name" cannot be empty. Should display message to user	Pass
2	Password	Enter null string in "Password" Textbox	"Password" cannot be empty. Should display message to user	"Password" cannot be empty. Should display message to user	Pass
3	User Name	Check maximum number of characters allowed in "User Name" Textbox	Should allow 10 characters in "User Name"	Should allow 10 characters in "User Name"	Pass
4	User Name	Check Shift+Tab functionality in "User Name" Textbox	On hitting Shift+Tab key the focus should be taken to the screen	On hitting Shift+Tab key the focus should be taken to the screen	Pass
5	Password	Check Tab functionality in "Password" Textbox	On hitting Tab key the focus should be taken to the "Login" Button	On hitting tab key the focus should be taken to the "Login" Button	Pass

6	User Name	Enter invalid User name in "User Name" Textbox	Should display message to user	Should display message to user	Pass
7	Password	Enter invalid User name in "Password" Textbox	Should display message to user	Should display message to user	Pass
8	User Name and Password	Enter valid User Name and Password	Should navigate to Home page	Should navigate to Home page	Pass
9	Password	Check Shift+Tab functionality in "Password"	On hitting Shift+Tab key the focus should be taken to the "User Name" TextBox	On hitting Shift+Tab key the focus should be taken to the "User Name" TextBox	Pass

Table: 6.2.2.1 Test cases for Login Screen

S. No	Test Case Module	Typical test strategy	Excepted Output	Actual Output	Test Status
1	Employee Registration	Have to enter the pin number.	Have to store in the database	Records stored in the database	Pass
2	Dealer Registration	Bank A/c No and Bank Name Should be entered	Store in Database	Records stored in the database	Pass

3	Billing	Appropriate Bill No Should be Entered	Corresponding items and their amount should be viewed	Items and their amount are viewed	Pass
4	Delivery Status	Click the delivered link such that message should be displayed	Status field should change	Status field are changed to 1	Pass

6.3 MAINTENANCE:

The objectives of this maintenance work are to make sure that the system gets into work all time without any bug. Provision must be for environmental changes which may affect the computer or software system. This is called the maintenance of the system. Nowadays there is the rapid change in the software world. Due to this rapid change, the system should be capable of adapting these changes. In our project the process can be added without affecting other parts of the system.

Maintenance plays a vital role. The system liable to accept any modification after its implementation. This system has been designed to favor all new changes. Doing this will not affect the system's performance or its accuracy.

6.4 SYSTEM IMPLEMENTATION

Implementation is the most crucial stage in achieving a successful system and giving the user's confidence that the new system is workable and effective. Implementation of a modified application to replace an existing one. This type of conversation is relatively easy to handle, provide there are no major changes in the system.

Each program is tested individually at the time of development using the data and has verified that this program linked together in the way specified in the programs specification, the computer system and its environment is tested to the satisfaction of the user. The system that has been developed is accepted and proved to be satisfactory for the user. And so the system is going to be implemented very soon. A simple operating procedure is included so that the user can understand the different functions clearly and quickly.

Initially as a first step the executable form of the application is to be created and loaded in the common server machine which is accessible to all the user and the server is to be connected to a network. The final stage is to document the entire system which provides components and the operating procedures of the system.

RESULT:

Thus various kinds of tests like unit testing, integration testing and validation testing are done. Thus it is tested that the system will work properly and effectively.

CHAPTER 7

PERFORMANCE AND LIMITATIONS

7.1 MERITS OF THE SYSTEM

This web application makes business transaction simpler. User can order the stock by being in the place. All business are done through online by this web application.

7.2 LIMITATION OF THE SYSTEM

This web application has been developed only for hardware components of computer. Only admin can have the rights to monitor and make changes in the proposed system.

7.3 FUTURE ENHANCEMENT

This newly developed system is able to meet the requirements of the company. Their requirements may change in near future. This web application is developed in such a way that is flexible for changes. Here are some of the ideas for future enhancements.

- The system can be linked with the internal mail of the organization so that alert messages like re-order level can be mailed to the respective users or employees.
- The clients can use the voice mail for their registration process and feedback process.
- Securities of the system for using hand print through password entries.

CHAPTER 8

CONCLUSION

It is concluded that the Consumer Issue Board which helps the organization to improve their customer and make their process quickly. The reports are produced easily through this automation process. It also eliminates the possibility of making any error in the manual calculation of the total transactions that take place.

The Consumer Issue Board will fully reduce the work of the organization. It makes the entire procedure faster, error free, simple and efficient and more performance enhance one. It enhances the scope of the entire project by integrating with the user login and registration, online payment, service register and make appointment to appropriate employee. It provides the basic details and information like users information, online transaction details. This information will be useful for future reference. The security feature of the system allows only administrator to make updates to important and sensitive data.

Thus the Consumer Issue Board increases efficiency, performance and reduces the work and helps to improve the organization.

CHAPTER 9

APPENDICES

9.1 SAMPLE SCREENS

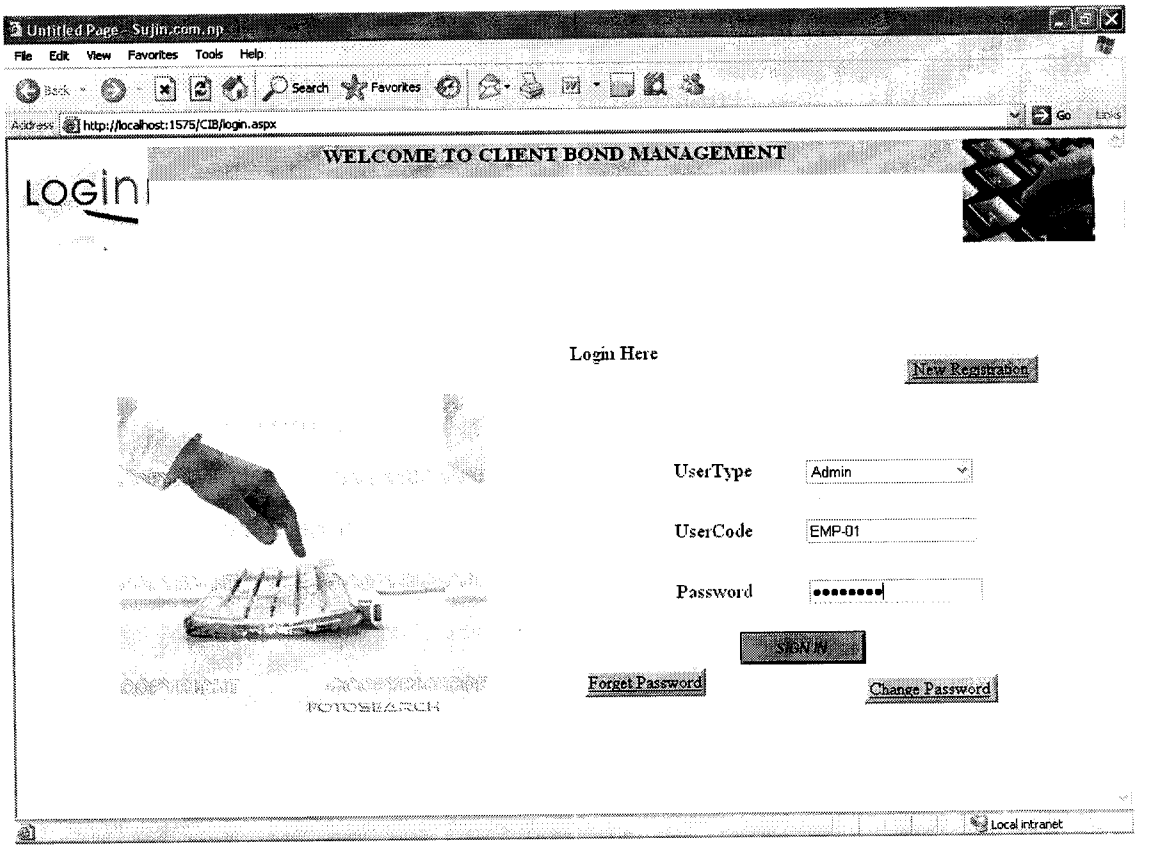


Figure A.1 User Login Form

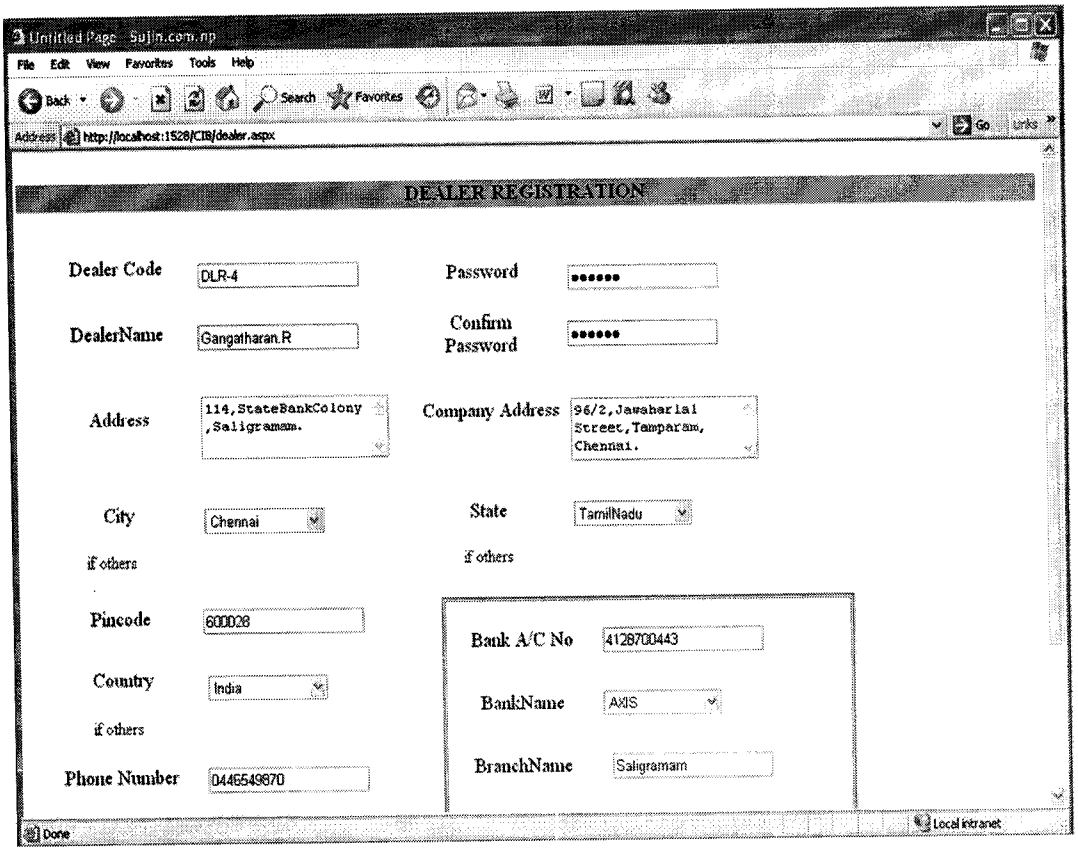


Figure A.2.a Dealer Registration Form

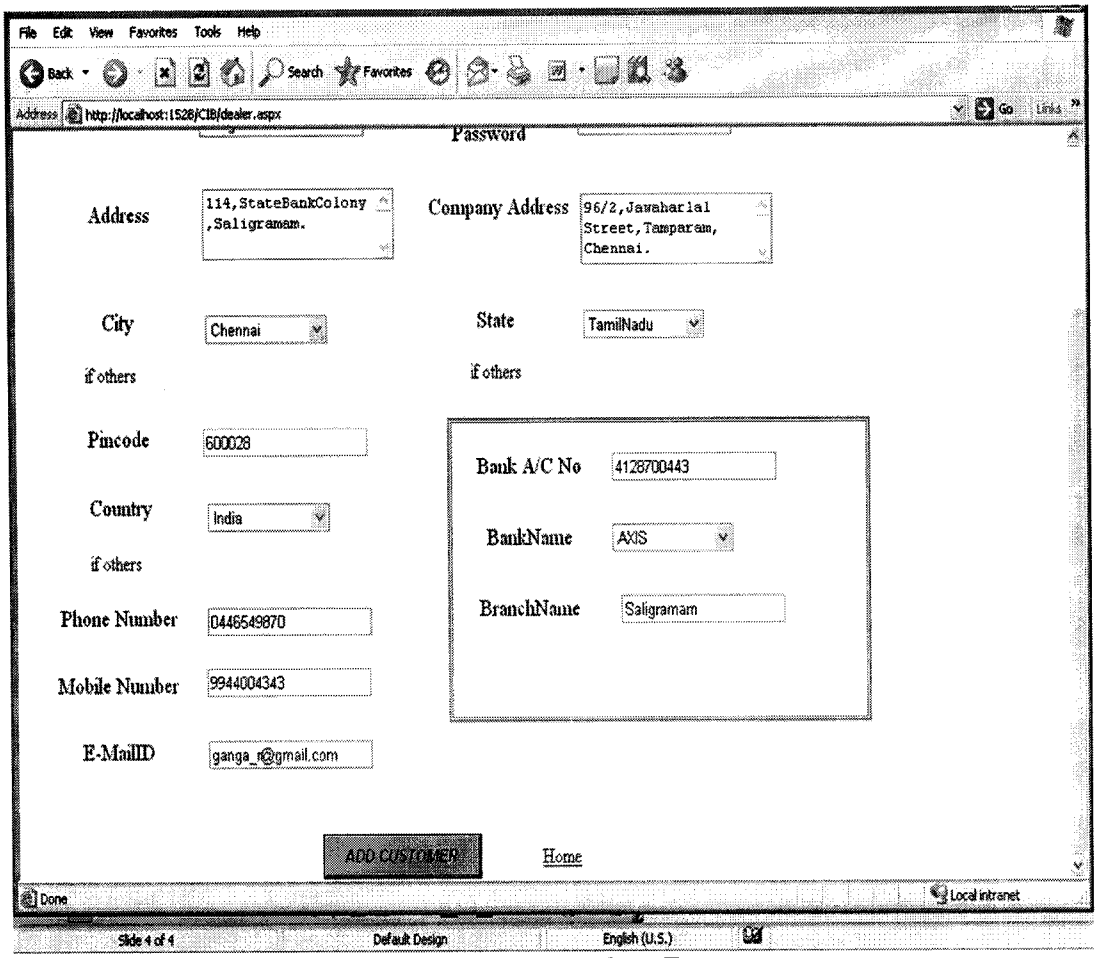


Figure A.2.b Dealer Registration Form

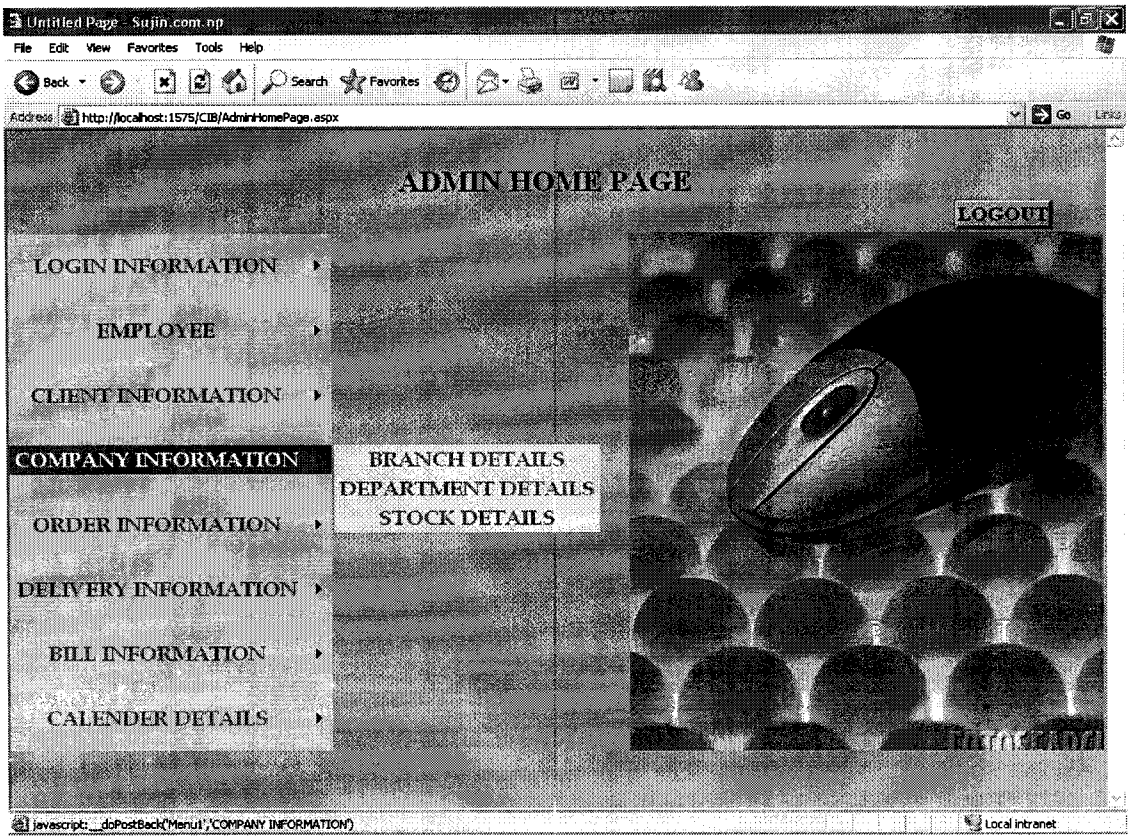


Figure A.3 Admin Home Page

Admin Operation:

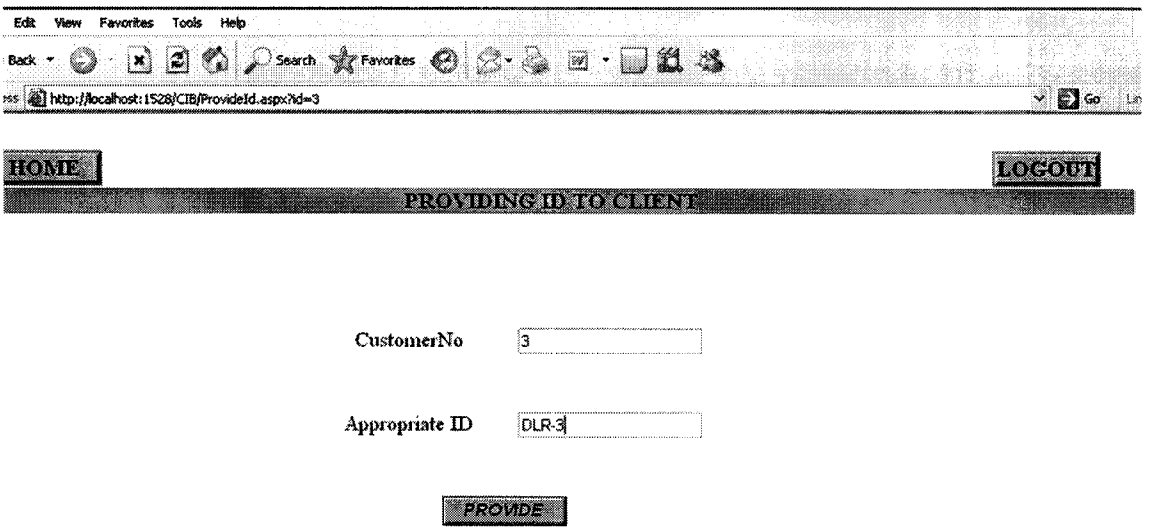
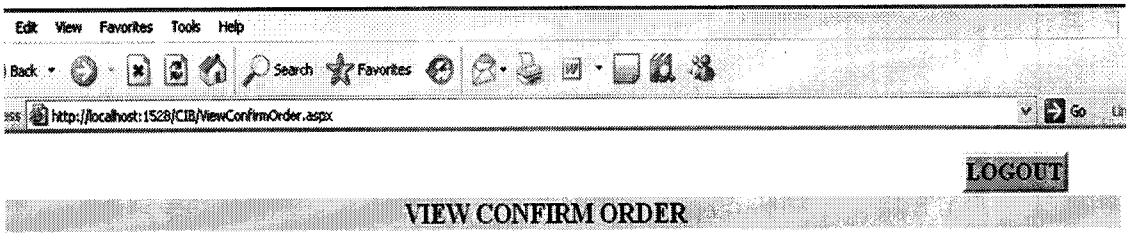
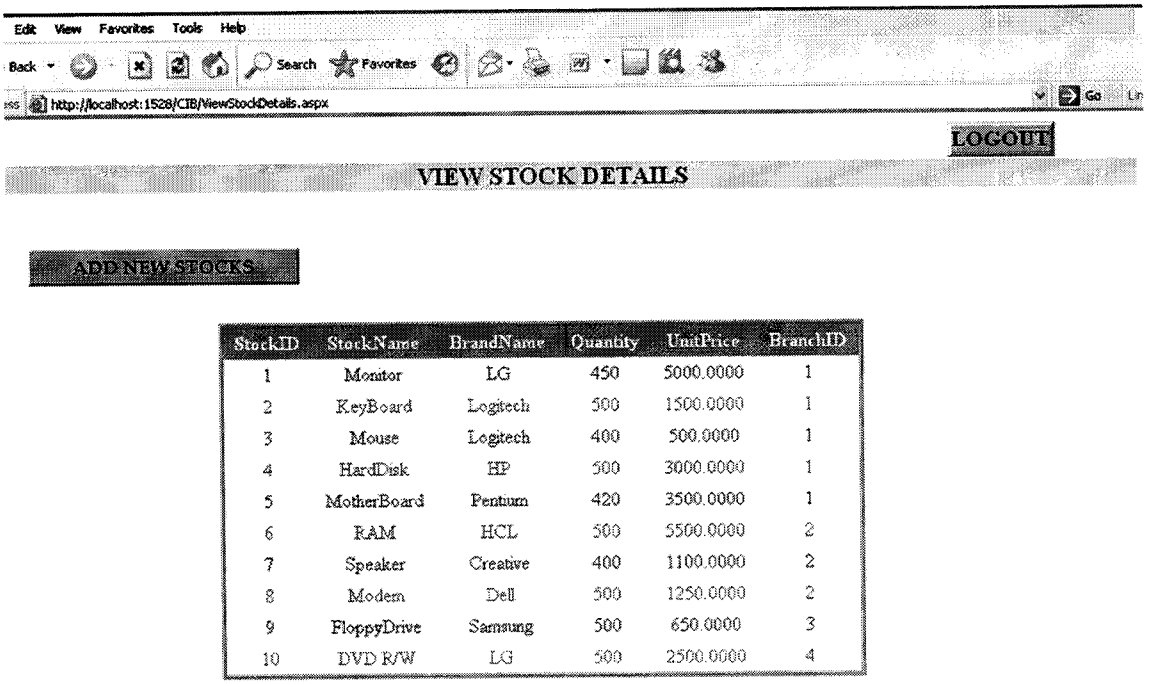


Figure A.4 Providing Id Form



OrderNo	CusId	ItemId	ItemName	BrandName	Quantity	BillNo	TotAmt	DTime	Utype
1	1	9	Floppy Drive	Samsung	10	Bill-1	2500	4/24/2008	Dealer
1	1	4	Harddisk	HP	20	Bill-1	100000	4/24/2008	Dealer
1	1	2	Keyboard	Logitech	10	Bill-1	7500	4/24/2008	Dealer
2	1	5	MotherBoard	Pentium	40	Bill-2	180000	4/25/2008	Dealer
2	1	7	Speaker	Creative	50	Bill-2	90000	4/25/2008	Dealer
3	3	1	Monitor	LG	50	Bill-3	250000	4/29/2008	Dealer
3	3	3	Mouse	Logitech	50	Bill-3	25000	4/29/2008	Dealer

Figure A.5 View Confirm Order Detail Form



StockID	StockName	BrandName	Quantity	UnitPrice	BranchID
1	Monitor	LG	450	5000.0000	1
2	KeyBoard	Logitech	500	1500.0000	1
3	Mouse	Logitech	400	500.0000	1
4	HardDisk	HP	500	3000.0000	1
5	MotherBoard	Pentium	420	3500.0000	1
6	RAM	HCL	500	5500.0000	2
7	Speaker	Creative	400	1100.0000	2
8	Modem	Dell	500	1250.0000	2
9	FloppyDrive	Samsung	500	650.0000	3
10	DVD R/W	LG	500	2500.0000	4

Figure A.6 View Stock Detail Form

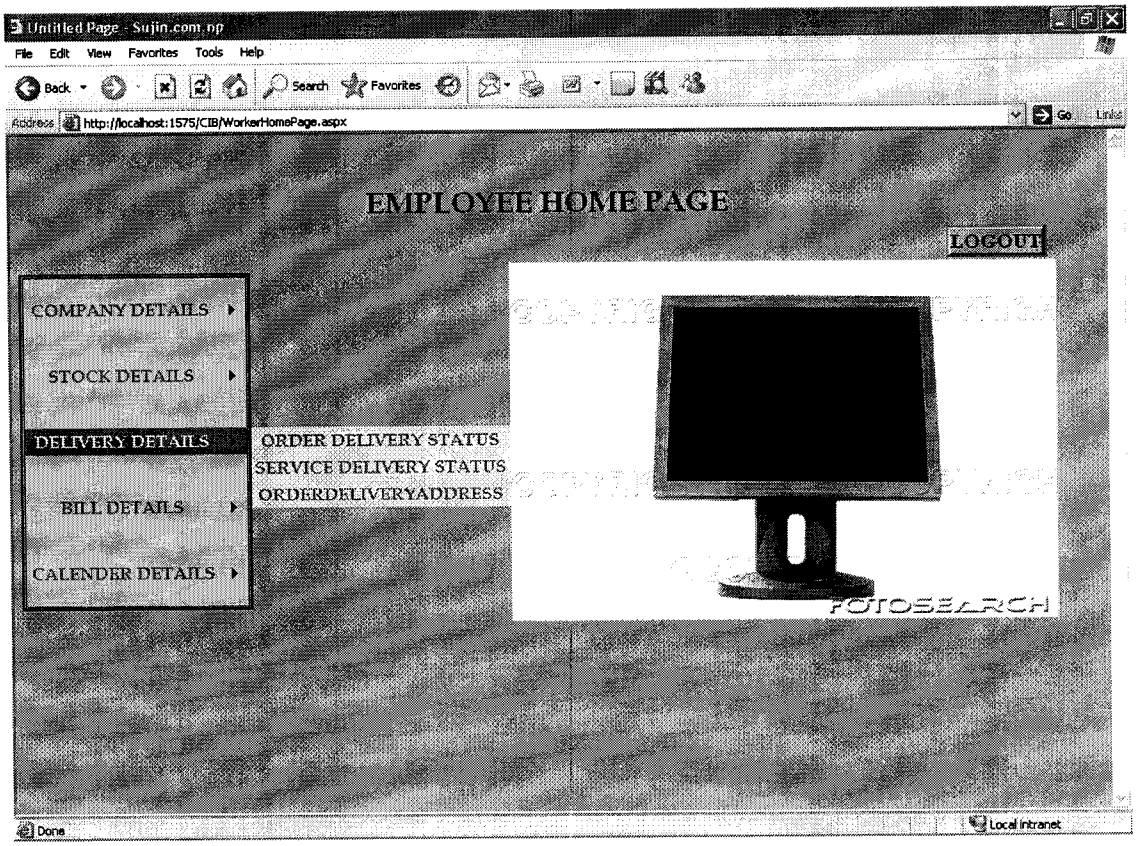
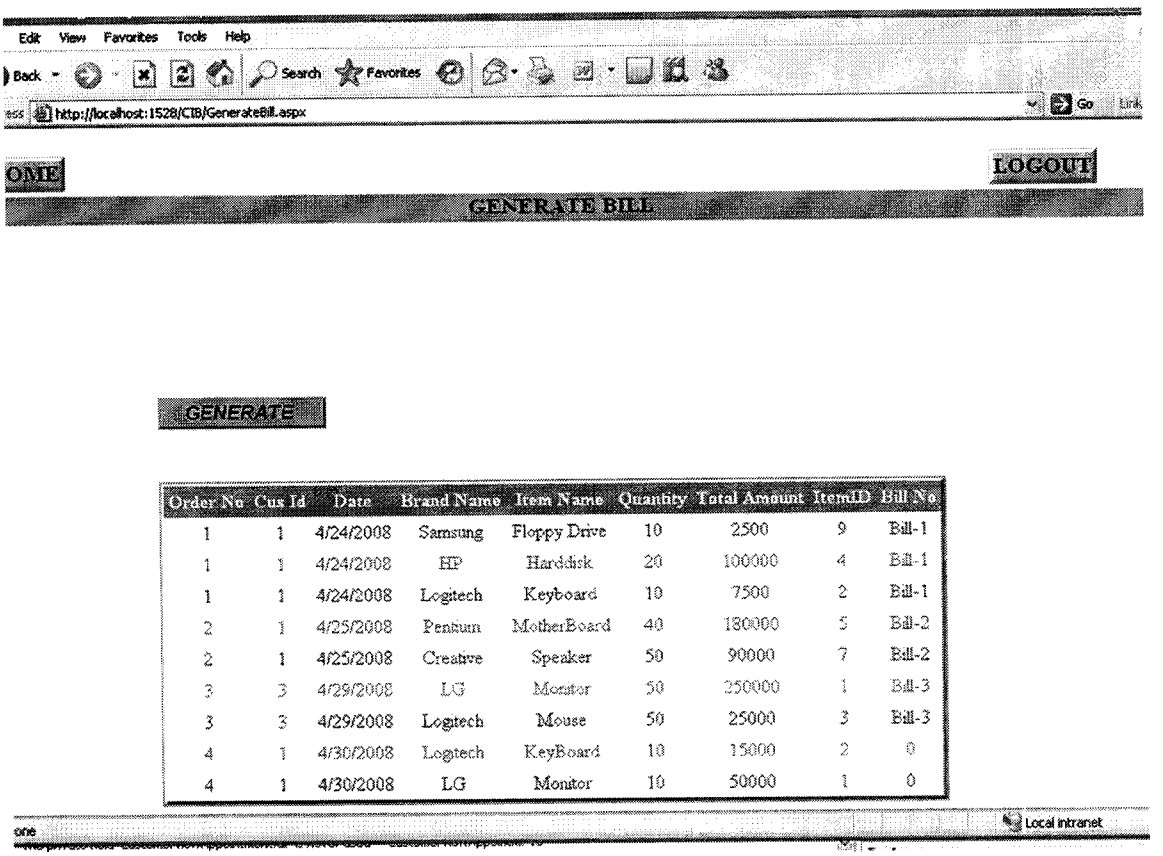


Figure A.7 Employee Home Page Form

Employee Operation:



Order No	Cus Id	Date	Brand Name	Item Name	Quantity	Total Amount	ItemID	Bill No
1	1	4/24/2008	Samsung	Floppy Drive	10	2500	9	Bill-1
1	1	4/24/2008	HP	Harddisk	20	100000	4	Bill-1
1	1	4/24/2008	Logitech	Keyboard	10	7500	2	Bill-1
2	1	4/25/2008	Pentium	MotherBoard	40	180000	5	Bill-2
2	1	4/25/2008	Creative	Speaker	50	90000	7	Bill-2
3	3	4/29/2008	LG	Monitor	50	250000	1	Bill-3
3	3	4/29/2008	Logitech	Mouse	50	25000	3	Bill-3
4	1	4/30/2008	Logitech	KeyBoard	10	15000	2	0
4	1	4/30/2008	LG	Monitor	10	50000	1	0

Figure A.8 Bill Status Form

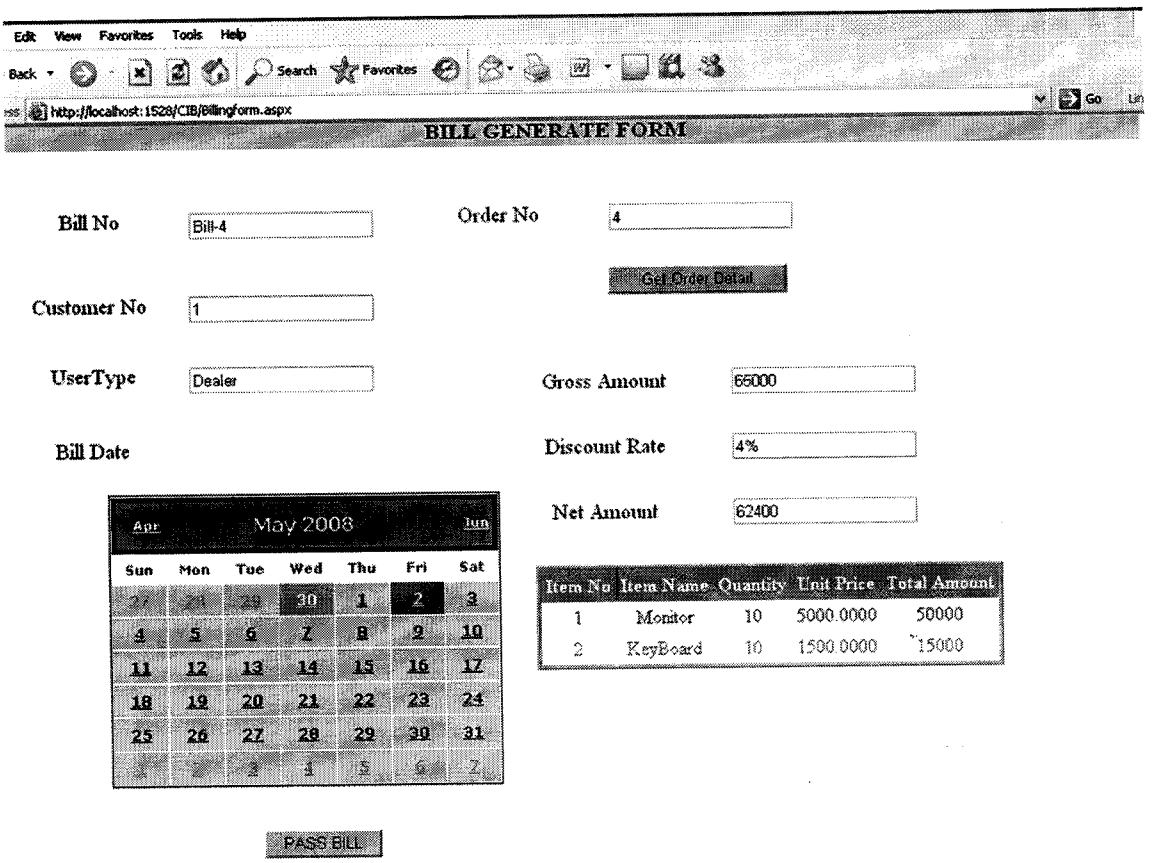
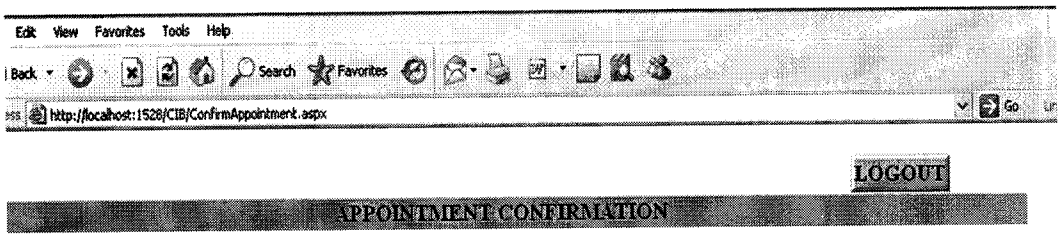


Figure A.9 Bill Generate Form



AppNo	CusId	BranchCity	UType	AppWith	UserId	AppDate	StartTime	EndTime	Status	
APP-1	DLR-1	Chennai	Admin	Sathya.R	2	29 04 2008	2:00pm	3:00pm	1	Confirmed
APP-2	DLR-2	Chennai	SalesManager	S.Selvaraj	1	30 04 2008	4:00pm	5:00pm	1	Confirmed
APP-3	DLR-3	Chennai	SalesManager	Navaneth.S	4	02 05 2008	10:00am	11:00am	1	Confirmed
APP-4	DLR-3	Chennai	Admin	Sathya.R	1	05 05 2008	11:00am	11:30am	1	Confirmed

Figure A.10 Appointment Confirmation Form

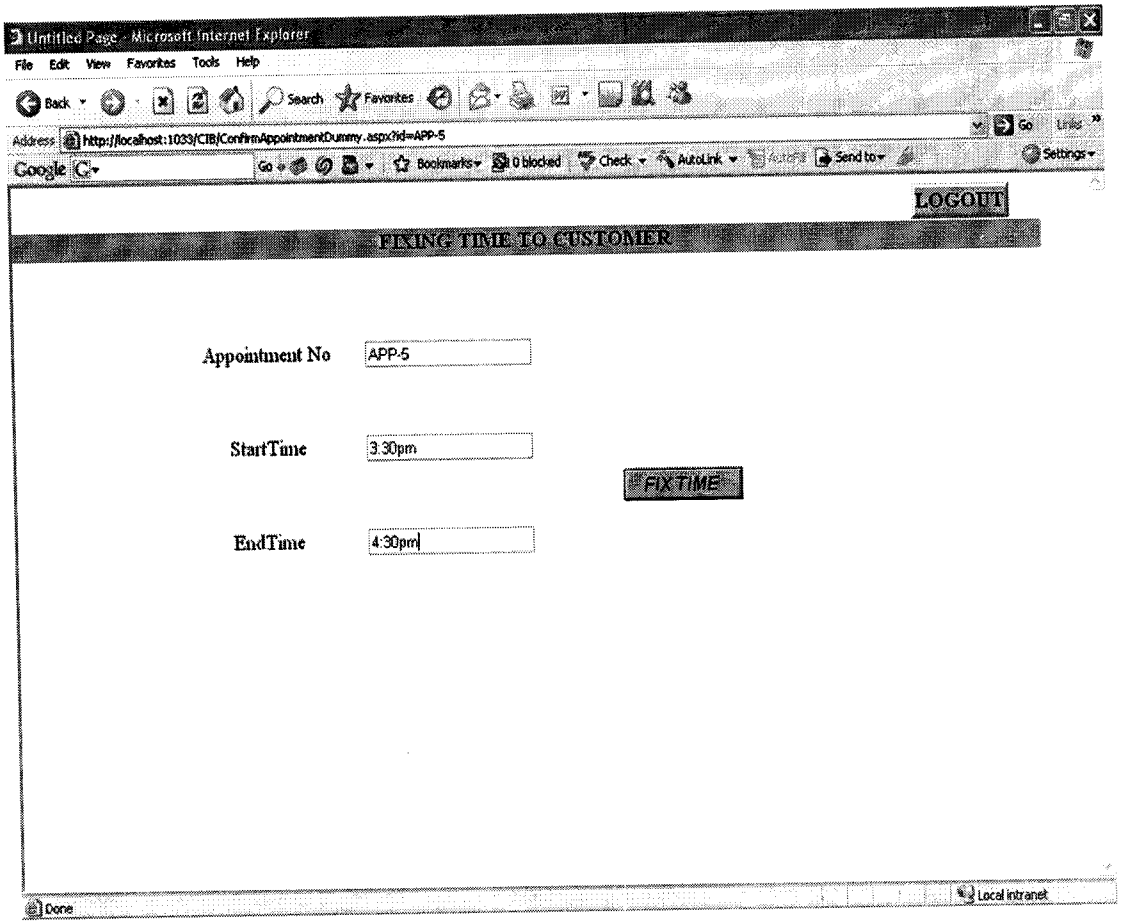


Figure A.11 Appointment Time Fixing Form

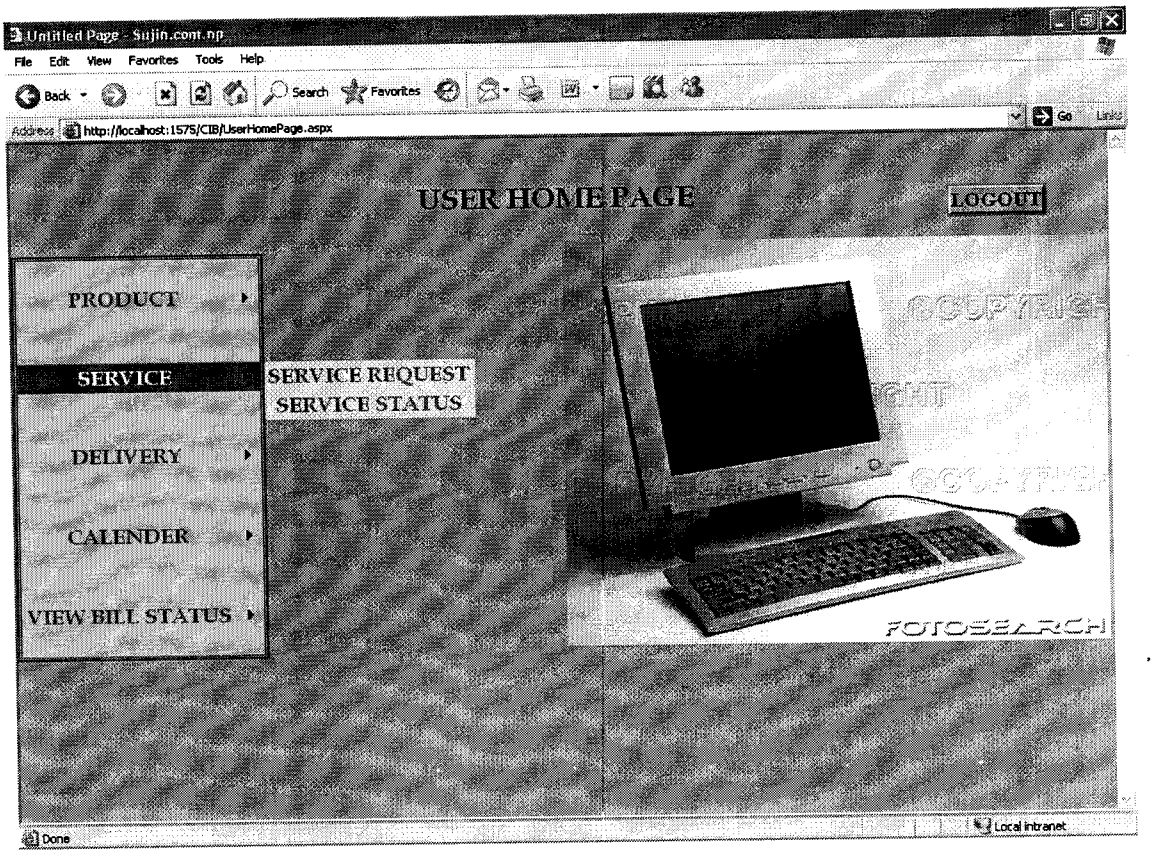


Figure A.12 User Home Page Form

User Operation:

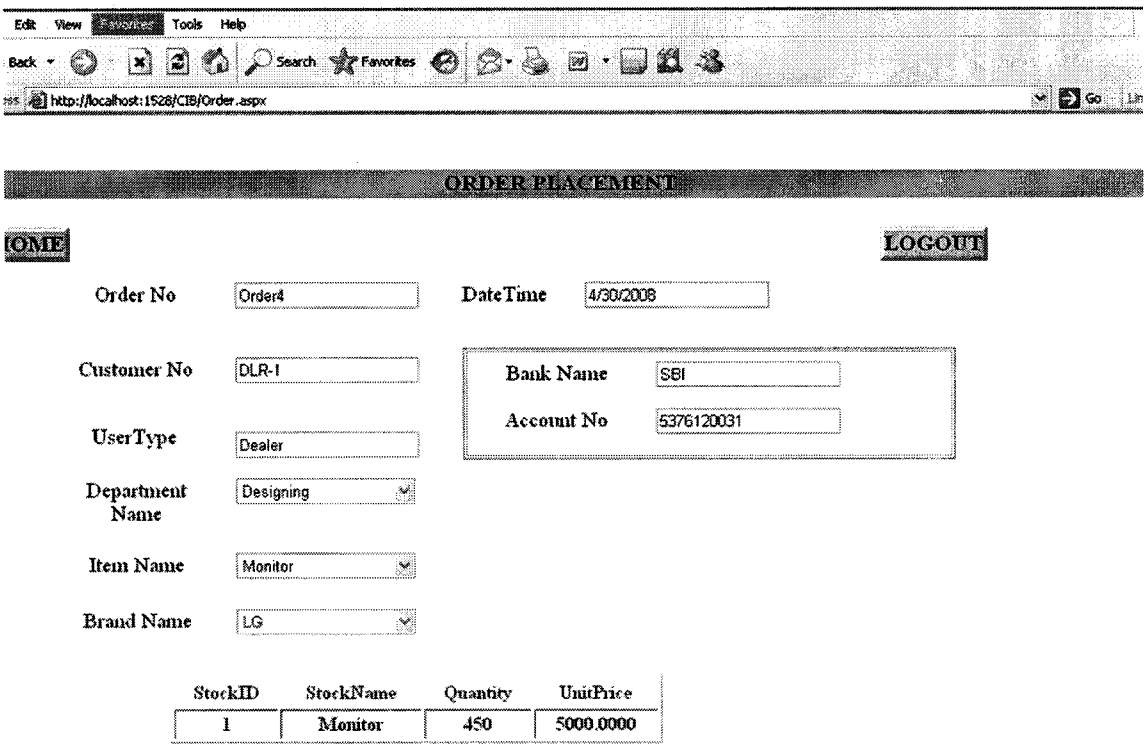


Figure A.13 Order Placement Form

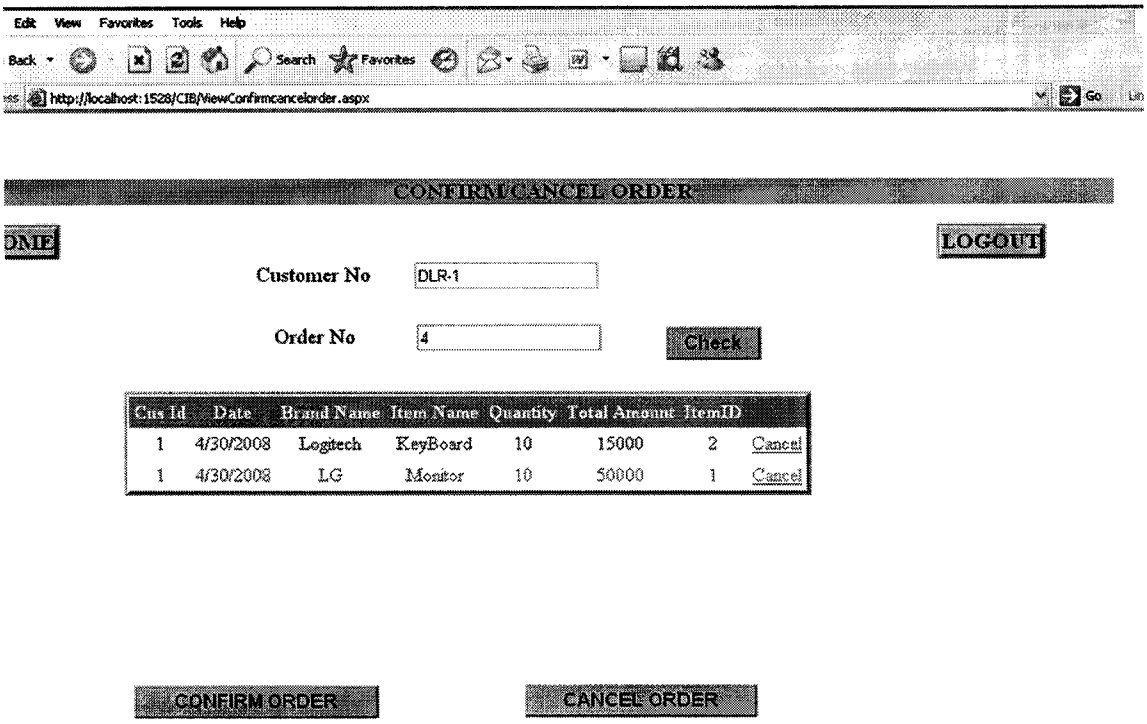


Figure A.14 Confirm/Cancel Order Form

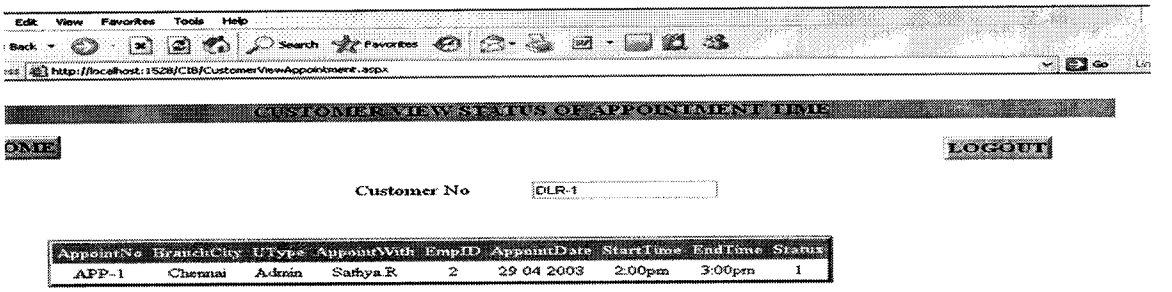


Figure A.15 Customer View Appointment Status Form

Reports:

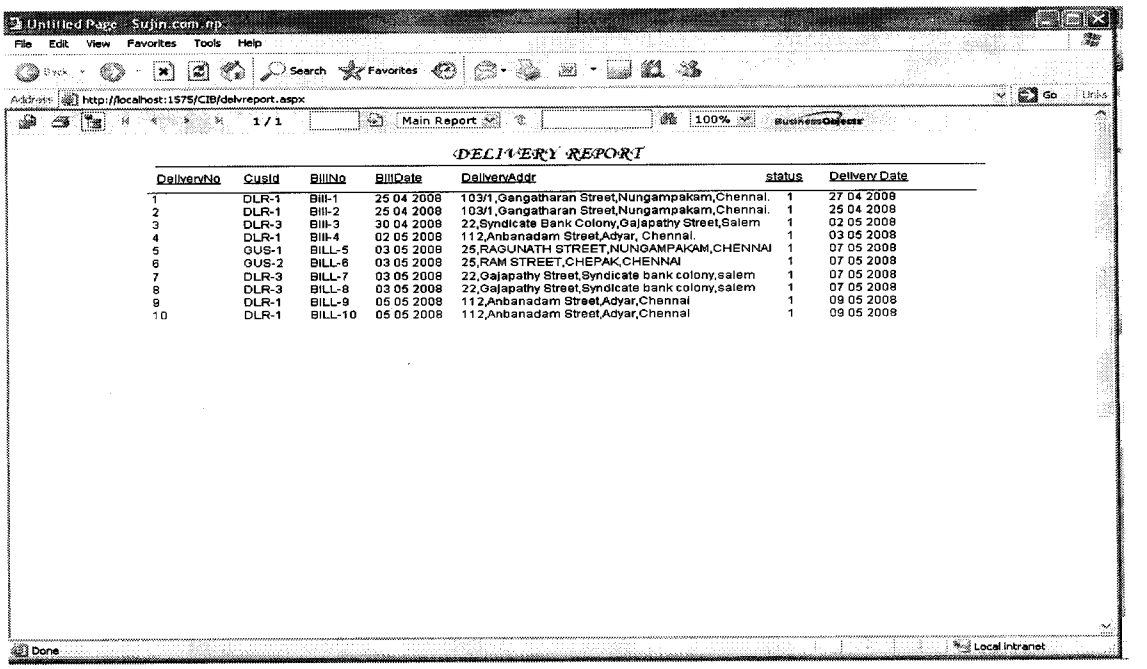


Figure A.16 Delivery Report Form

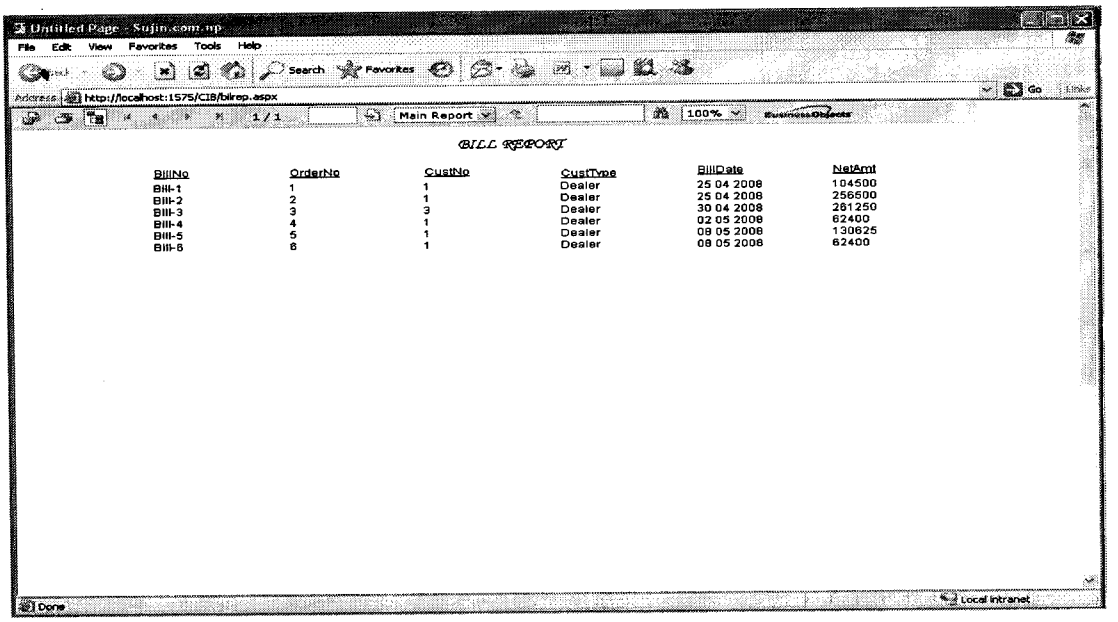


Figure A.17 Bill Report Form

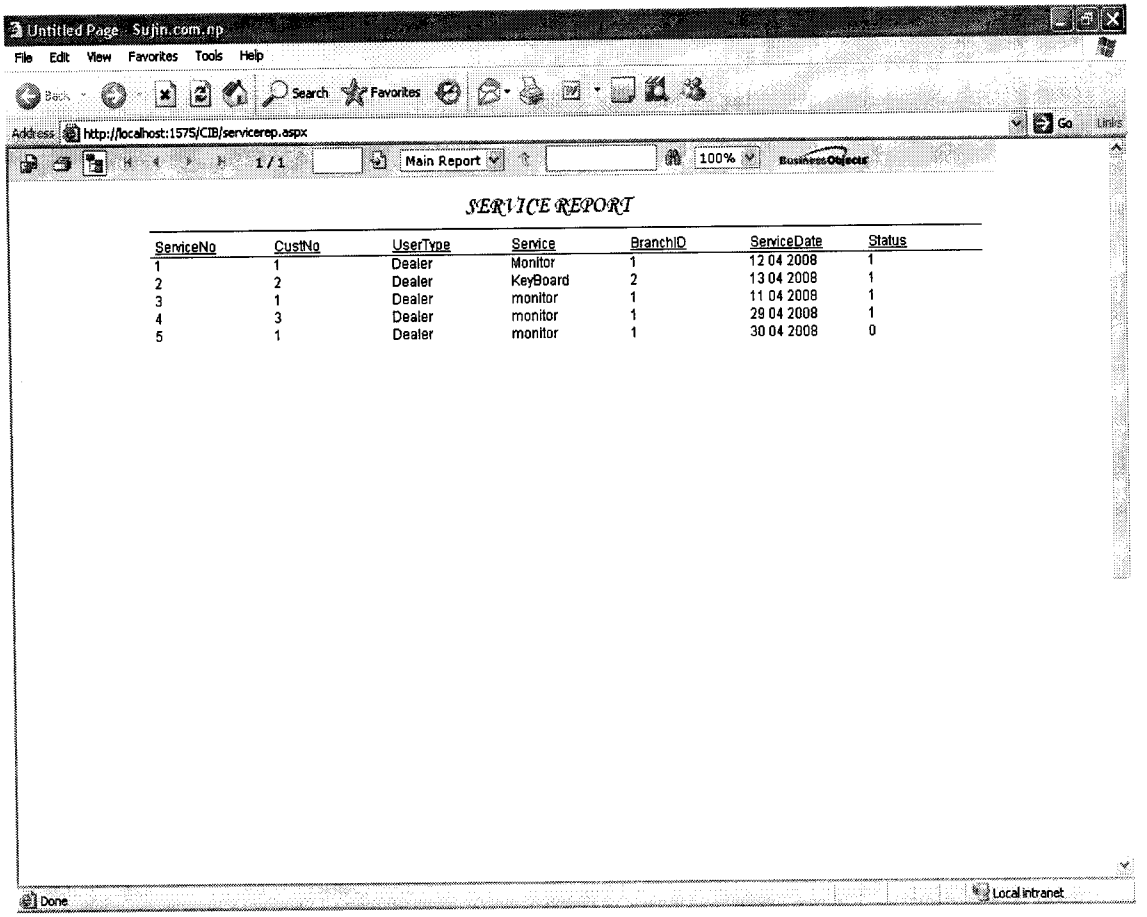


Figure A.18 Bill Report Form

REFERENCES

1. Elias M.Awad (2000) **"System Analysis and Design"**, Galgotia Publications.

2. Jeffery Richter (2002) **"Applied Microsoft .NET Framework Programming"**,WP Publishers.

3. Matthew MacDonald (2002) **"The Complete Reference ASP.NET"**, Tata McGraw-Hill Publishing Company Limited.

4. Roger S.Pressman (2000) **"Software Engineering"**, McGraw-Hill Publishing Company Limited.

REFERENCE SITES:

1. www.dotnet-guide.com.

2. http://msdn.microsoft.com