NOTATION ANALYSIS AND SCRUTINY SYSTEM

By

M.S.MURUGANANDAN

(Register Number: 71205621026)

Of

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641 006

A PROJECT REPORT
Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING
ANNA UNIVERSITY
Chennai 600 025

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

Of

MASTER OF COMPUTER APPLICATIONS

JUNE 2008

BONAFIDE CERTIFICATE

Certified that this project report titled "**NOTATION ANALYSIS AND SCRUTINY SYSTEM**" is the bonafide work of **Mr. M.S.MURUGANANDAN** 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.

Countersigned by

Signature of the Supervisor

Head of the Department

Submitted for project viva voce examination held on 01.07.08 .

INTERNAL EXAMINER

EXTERNAL EXAMINER

DATE: 19-06-2008

CERTIFICATE

This is to certify that the project titled **"NOTATION ANALYSIS AND SCRUTINY"** submitted by **Mr.M.S.MURUGANANDAN (Reg.No: 71205621026)** from **KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE** in partial fulfillment of M.C.A is a bonafied work of original project work done at **ESPADE IT** between DEC 2007 to MAY 2008.

It is seen that a project was successfully completed and complies with all the requirements suggested initially.

M. Sabarinathan

Mr.M.Sabarinathan.M.C.A
(Tech Lead)

ESPADE IT
BANGALORE

ABSTRACT

This project is entitled "**NOTATION ANALYSIS AND SCRUTINY SYSTEM**". The main aim of the business is the Customer satisfaction. An organization/company will have a number of products and services they may offer to their customers. Depending on the type of business or service offered the customers may have several doubts, complaints, and problem. The "Notation Analysis and Scrutiny System" helps to accomplish this task of solving and clarifying the customers' queries. The "Notation Analysis and Scrutiny System" can be used by a business organization to provide online support to its customers.

This may include questions about their services or even complaints the customers may have. Although a manual system can be done, time plays an important role in customer satisfaction. A customer always expects services to be offered as soon as possible and the organization is responsible for making sure its customers stay satisfied. The "Notation Analysis and Scrutiny System" will prove an added advantage to them in this internet world.

In this project **ASP.Net with C#** is used as the **Front-End** and **SQL Server** is used as the **Back-End**. This project is designed in such a way it is user friendly with look and feel appearance, portable and flexible enhancement.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowns all efforts with success.

I wish to express my profound gratitude to **Dr. Joseph V. Thanikal, Ph.D, Principal – Kumaraguru College of Technology, Coimbatore** for providing an opportunity and necessary facilities in carrying out this project work.

I am very glad to extend my deepest thanks to **Dr. M. Gururajan M.Sc., Ph.D, Head of the Department, Kumaraguru College of Technology, Coimbatore,** who has always been a source of inspiration.

I tender my special thanks to **Ms.V.Geetha M.C.A,, M.Phil,** Assistant Professor and Project Coordinator, Department of Computer Applications for his constant encouragement.

My wholesome thanks to my Project mentor **Ms.R.K.Kavitha M.C.A.,M.Phil** for his continued encouragement and timely help during the project.

I also dedicate equal and grateful acknowledgements to all the respectable members of the faculty and lab in-charges of the **Department of Computer Applications, Kumaraguru College of Technology, Coimbatore** and student friends for their motivation, encouragement and continuous support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SL.NO.	ABBREVIATIONS	DESCRIPTION
1	ADO	ActiveX Data Object
2	GUI	Graphical User Interface
3	IIS	Internet Information Service
4	SQL	Structured Query Language
5	DFD	Data Flow Diagram
6	CSR	Customer Support Representative
7	QA	Quality Analyst
8	XP	e***XP***erience
9	API	Application Programming Interface

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT

The manual submission of complaints and customer servicing has been the most tedious part of the business. The time delay for a response and accuracy of complaint solving plays as vital role for customer servicing. This scenarios and objectives play the mandatory development of application to serve the customer with customer service satisfaction.

This System will allow the business process to run efficiently and will be used to assist or satisfy the customer by giving a brief solution for their complaint within a minimum time. The Customer receives the solution within a day online.

The customers may arise with many problems which cannot be solved at the same time with full support by the business. The main aim of the business is the Customer satisfaction. This system helps the customers to give their problems online whenever the customer needs, without any constraints implied on them. This system helps the customers to briefly explain their complaint without any hesitation.

Through this system the customers are assisted with some predefined complaints and immediate solutions which save the precious time of the customers. The customers receive the solutions for their complaint within a day and the customer can also view the status of the complaint at any time using the complaint number. Through this system the customers receive a brief solution for their complaint.

1.2 ABOUT THE ORGANIZATION

E-Spade IT is one among the emerging global consulting and IT services company, offering wide array of solution.

E-Spade IT is a customer focused company working to provide solutions, which are delivered with reliability and timeliness, flexibility and low cost. Customer satisfaction is our aim. We design solutions keeping your Business requirements as well as Technical requirements in mind. We strictly follow Software development life cycle to ensure that project is delivered in time without compromising Quality.

As a diverse end to end solutions provider, E-Spade IT offers a range of expertise aimed at helping customers re-engineer and re-invent their business to compete successfully. We help you in managing IT operations, increasing efficiency and cost control, improving time-to-time market and software quality in application development and delivery. We have the necessary expertise in offering Software solutions in multiple technologies like Web Design, .Net, J2EE, Software Testing, PHP, Perl. We have also a specialized team working on Multimedia and CAD/CAM Projects.

We are a group of highly motivated IT professionals with ability to innovate and a strong desire to excel. All our team members are constantly and consistently trained with the latest tools. We have a range of Services to offer for our clients and we are aiming to make it one stop solutions for all your Software Development needs.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System study is the process of gathering and interpreting facts, diagnosing the problems and using the information to recommend improvements on the system. System analysis or study is a problem solving activity that requires intensive communication between the system users and system developers. The system is studied to the minutest details and analyzed. The system analyst plays the role of an interrogator and dwells deep into working of the present system. The system is viewed as a whole and the inputs to the system are identified. The outputs from the organization are traced through various phases of the processing of inputs.

2.2 PROBLEM DEFINITION

An organization/company will have a number of products and services they may offer to their customers. Depending on the type of business or service offered the customers may have several doubts/complaints/problems. The **"NOTATION ANALYSIS AND SCRUTINY SYSTEM"** helps to accomplish this task of solving and clarifying the customers' queries. This product facilitates the user to submit online complaints independent of the system

The **"NOTATION ANALYSIS AND SCRUTINY SYSTEM"** can be used by a business organization to provide online support to its customers. This may include questions about their services or even complaints the customers may have. A customer always expects services to be offered as soon as possible and the organization is responsible for making sure its customers stay satisfied.

2.3 EXISTING SYSTEM

Current System doesn't extend its functionality for Overseas Business Process and Outsourcing. The cost and human resource expenses towards building Customer Services for Business prosperity with in the Locality are undesirable. The system cannot be enhanced to provide distribution of data for different services. The complaints are dropped in the complaint box and they are collected by the respective members for scanning process and served on day to day basis. This process model and business approach is inefficient and not dynamic for the future growth of the organization. Analyzing the Employees performance and generating reports are done manually which is time consuming.

The drawbacks of using the existing system are

Time-consuming

A tedious task

Cause damage to the brand image of the organization

Slow updating and retrieval of information

2.4 PROPOSED SYSTEM

The aim is to develop a system to assist the customers by providing solutions for their complaints. This system focuses on receiving the complaints from the customer, distributing complaints to the customer support representatives, solving the complaints based on some constraints, solved complaints are verified by the quality analyst, providing accuracy percentage to the customer support representatives based on failure constraints, validation engine which is used to provide the solution to a repeated complaint.

The complaints are distributed by the distribution engine category wise. The Customer Support Representatives are classified under different categories. The distribution engine does not distribute the same complaint for more than one customer support representative. The distribution engine maintains the complaint details of how many complaints have been distributed to the Customer Support Representative and how many complaints have been solved and how many complaints have been verified.

The distribution engine maintains all the complaint details category wise. The Complaints are tracked as when the complaints are distributed to the customer support representative and when the customer support representatives solve the complaint. The Time taken for each complaint to be solved is also maintained. The solution given by the customer support representatives are forwarded to the team leaders who verify the solution. The Team Leaders verify whether the given solution has followed the deployed constraints.

The customers can view the status of the complaint using the complaint number generated at the time of submission. The complaint number is submitted by the customer only after login.

2.5SYSTEM OBJECTIVE

The manual submission of complaints and customer servicing has been the most tedious part of the business

The time delay for a response and accuracy of complaint solving plays as vital role for customer servicing.

This System will allow the business process to run efficiently and will be used to assist or satisfy the customer by giving a brief solution for their complaint within a minimum time. The Customer receives the solution within a day online.

The main aim of the business is the Customer satisfaction.

This system helps the customers to give their problems online whenever the customer needs, without any constraints implied on them

2.6 SYSTEM SPECIFICATION

2.6.1 Software Specification

- Operating System : Windows XP
- Application : ASP.NET
- Back-End Tool : Microsoft SQL Server 2000
- Application server : IIS 5.1

WINDOWS XP

Windows XP is a line of proprietary operating systems developed by Microsoft for use on general-purpose computer systems, including home and business desktops, notebook computer, and media centers. The letters "XP" stands for *eXPerience*. Codenamed "Whistler" after Whistler, British Columbia, as many Microsoft employees skied at the Whistler-Blackcomb ski resort during its development, Windows XP is the successor to both Windows 2000 and Windows Me, and is the first consumer-oriented operating system produced by Microsoft to be built on the Windows NT kernel and architecture. Windows XP was first released on October 25, 2001, and over 400 million copies are in use, according to a January 2006 estimate by an IDC analyst. It is succeeded by Windows Vista, which was released to volume license customers on November 8, 2006 and worldwide to the general public on January 30, 2007.

Windows XP is known for its improved stability and efficiency over previous version of Microsoft Windows. Its presents a redesigned graphical user interface, a change Microsoft promoted as more user-friendly than previous version of Windows. New software management capabilities were introduced to avoid the "DLL hell" that plagued older consumer versions of Windows. It is also the first version of Windows to use product activation to combat software privacy, a restriction that did not sit well with some users and privacy advocates. Windows XP

has also been criticized by some users for security vulnerabilities, tight integration of applications such as Internet Explorer and Windows Media player, and for aspects of its user interface.

.NET FRAMEWORK

The .NET Framework is many things, but it is worthwhile listing its most important aspects. In short .NET Framework is:

A platform designed from the start for writing Internet-aware and Internet-enabled applications that embrace and adopt open standards such as XML, HTTP, and SOAP.

A platform that provides a number of very rich and powerful application development technologies, such as Windows Forms, used to build classic GUI applications, and of course ASP.NET, used to build web applications.

A platform with an extensive classic library that provides extensive support for date access (relational and XML), a director services, message queuing, and much more.

A platform that has a base class library that contains hundreds of classes for performing common tasks such as file manipulations, registry access, security, threading, and searching of text using regular expressions.

A platform that doesn't forgot its origins, and has great interoperability support for existing components that you or third parties have written, using COM or standard DLL's.

A platform with an independent code execution and management environment called the Common Language Runtime(CLR), which ensures code is safe to run, and provides an abstract

layer on top of the operating system, meaning that elements of the .NET Framework can run on many systems and devices.

ASP.NET

ASP.NET is part of the whole .NET Framework, built on top of the Common Language Runtime (also known as the CLR) – a rich and flexible architecture, designed not just to cater for the needs of developers today, but to allow for the long future we have ahead of us. ASP.NNET is very much more than just an upgrade of existing technology – it is the gateway to a whole new era of web development.

ASP.NET is as feature at the following web server releases

Microsoft IIS 5.0 on WINDOWS 2000 Server

Microsoft IIS 5.1 on WINDOWS XP

ASP.NET has been designed to try and minimum syntax and run-time compatibility with existing ASP pages wherever possible. The motivation behind this is to allow existing ASP pages to be initially migrated ASP.NET by simply renaming the file to have an extension of .aspx. For the most part this goal has been achieved, although there are typically some basic code changes that have to be made, since VBScript is no longer supported, and the VB language itself has changed.

Some of the key goals of ASP.NET were to

Remove the dependency on the script engines; enable pages to be type safe and compiled.

Reduce the amount of code required to develop web applications.

Benefits of ASP.NET

The .NET Framework includes a new data access technology named ADO.NET, an evolutionary improvement to ADO. Though the new data access technology is evolutionary, the classes that makes up ADO.NET bear little resemblance to the ADO objects with which might be familiar. Some fairly significant changes must be made to existing ADO applications to convert them to ADO.NET. The don't have to be made immediately to existing ADO applications to run under ASP.NET, however.

ASP.NET has several advantages over ASP:

The following are some of the benefits of ASP.NET:

Make code cleaner.

Improve deployment, scalability, and reliability.

Provide better support for different browsers and devices.

Enable a new breed of web applications.

SQL

SQL is a standard computer language for accessing and manipulating database.

What is SQL?

SQL stands for Structured Query Language

SQL allows to access a database

SQL is an ANSI standard computer language

SQL can execute queries against a database

SQL can retrieve data from a database

SQL can insert new records in a database

SQL can delete records from a database

SQL can update records in a database

SQL is easy to learn

SQL as a Standard

SQL is an ANSI (American National Standard Institute) standard computer language for accessing and manipulating database systems. SQL statements are used to retrieve and update data in a database. SQL works with database programs like MS Access, DB2, Informix, MS SQL Server, Oracle, Sybase, etc. Unfortunately, there are many different versions of the SQL language, but to be in compliance with the ANSI standard, they must support the same major keywords in a similar manner (such as SELECT, UPDATE, DELETE, INSERT, WHERE, and others).

SQL SERVER 2000

Microsoft SQL Server 2000, the data management and analysis backbone for Microsoft's .NET enterprise applications and servers, improves on the performance, reliability, scalability, quality, and ease of use of its predecessors.

SQL Server 7.0 and SQL Server 6.5, SQL Server 2000 offers rich Extensible Markup Language (XML) support, comprehensive analysis services, and simplified database administration - features that combine to produce a solution able to rapidly deliver reliable, scalable e-commerce, data warehousing and line of business upgrade from SQL Server 6.5 and

SQL Server 7.0.

Microsoft SQL Server 2000 introduce new feature that support XML functionality. The combination of these features makes SQL Server 2000 an XML enabled database server. These features include:

The ability to access SQL Server using HTTP.

Support for XDR (XML – Data Reduced) schemes and the ability to specify XPath queries against these schemas.

IIS

Microsoft Internet Information Services (IIS; formerly called **Server)** is a set of Internet-based services for servers using Microsoft Windows. It is the world's second most popular web server in terms of overall websites. As of May 2007 it served 31% of all websites according to Netcraft. The servers currently include FTP, SMTP, NNTP and HTTP/HTTPS.

2.6.2 Hardware Specification

- Processor : Intel Pentium III
- RAM : 256 MB
- Memory : 64 Megabytes
- Hard Disk : 20 GB
- Keyboard : Standard-104 Keys
- Mouse : Serial- Scroll mouse

CHAPTER 3

SYSTEM DESIGN

3.1 INTRODUCTION

Design is the first step in the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term "design" is defined as "the process of applying various techniques and principle for the purpose of defining a process or a system in sufficient details to permit its physical realization". Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used.

From the project management point of view, software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements in to a data and software architecture. Detail design focuses on refinement to the architectural representation that lead to detail algorithm data structure and representation of software.

Design starts with the system requirement specification and converts it to a physical reality during the development. Important design factors such as reliability, response time, throughput of the system, maintainability, expandability, etc should be taken into account.

3.2 DATA FLOW DIAGRAM

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.

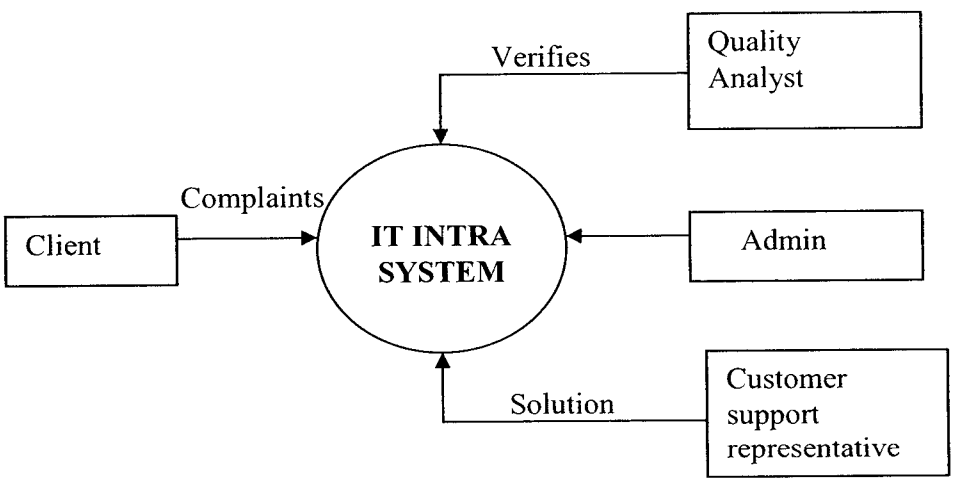

Fig 3.1. Context Level DFD

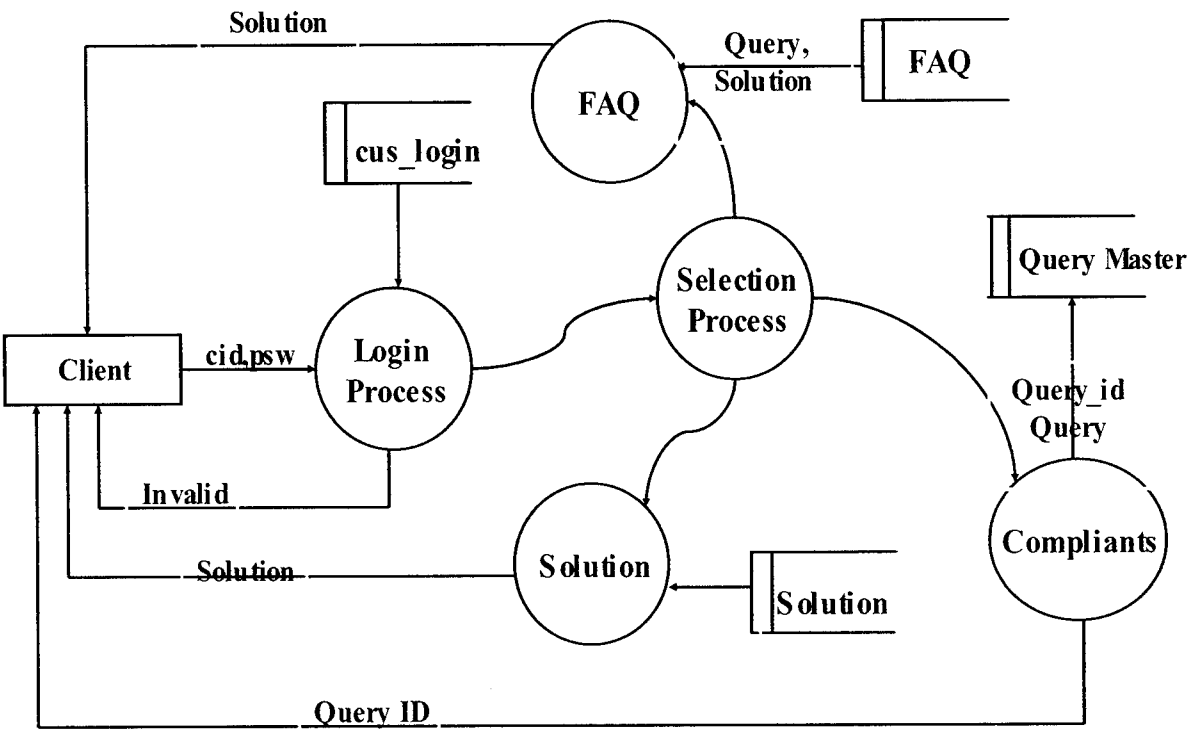

Fig 3.2. Level 1 DFD-Client

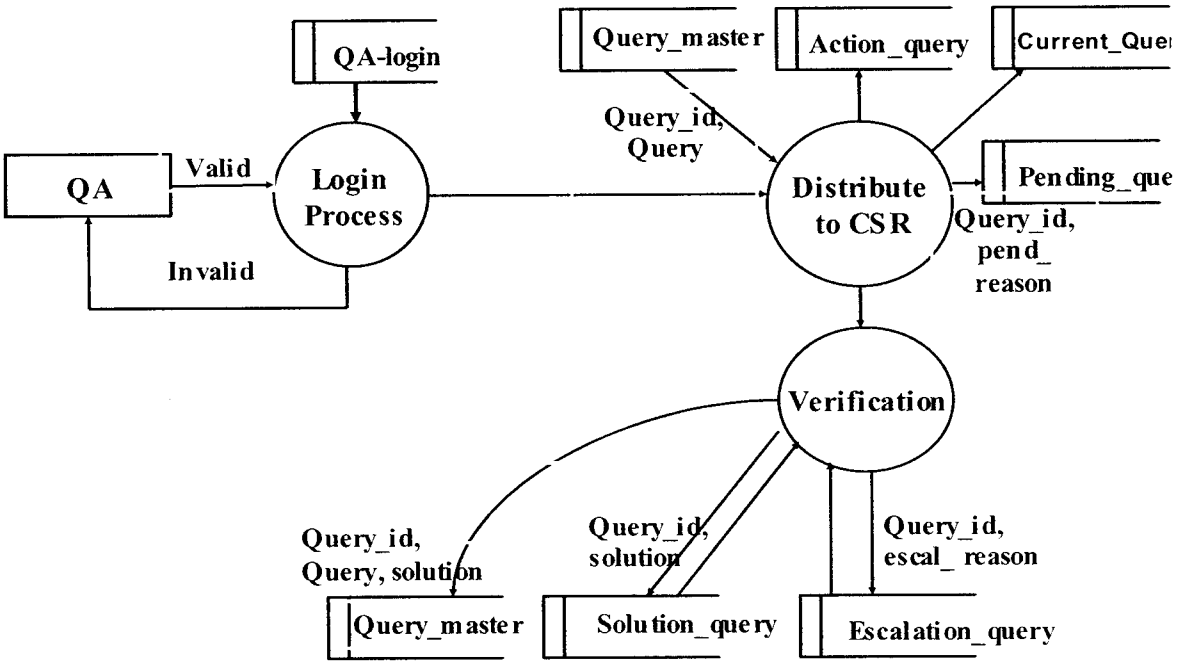

Fig 3.3. Level 1 DFD-Quality Analyst

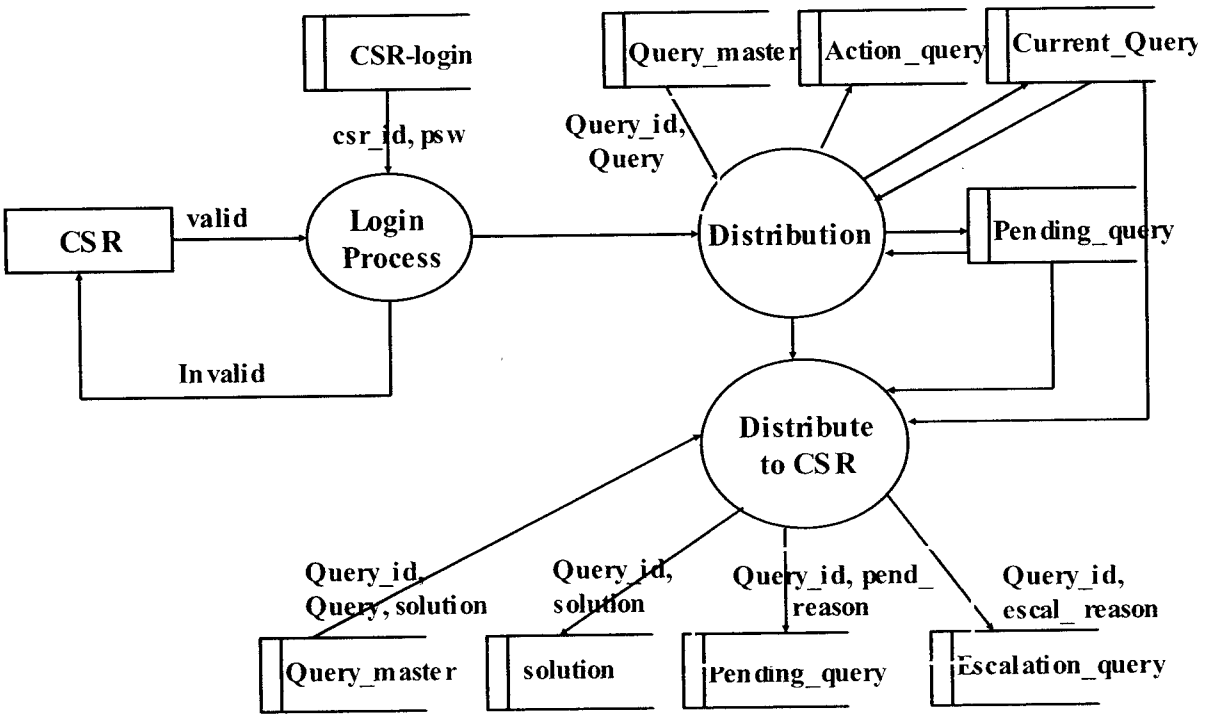

Fig 3.4. Level 1 DFD-Customer Support Representative

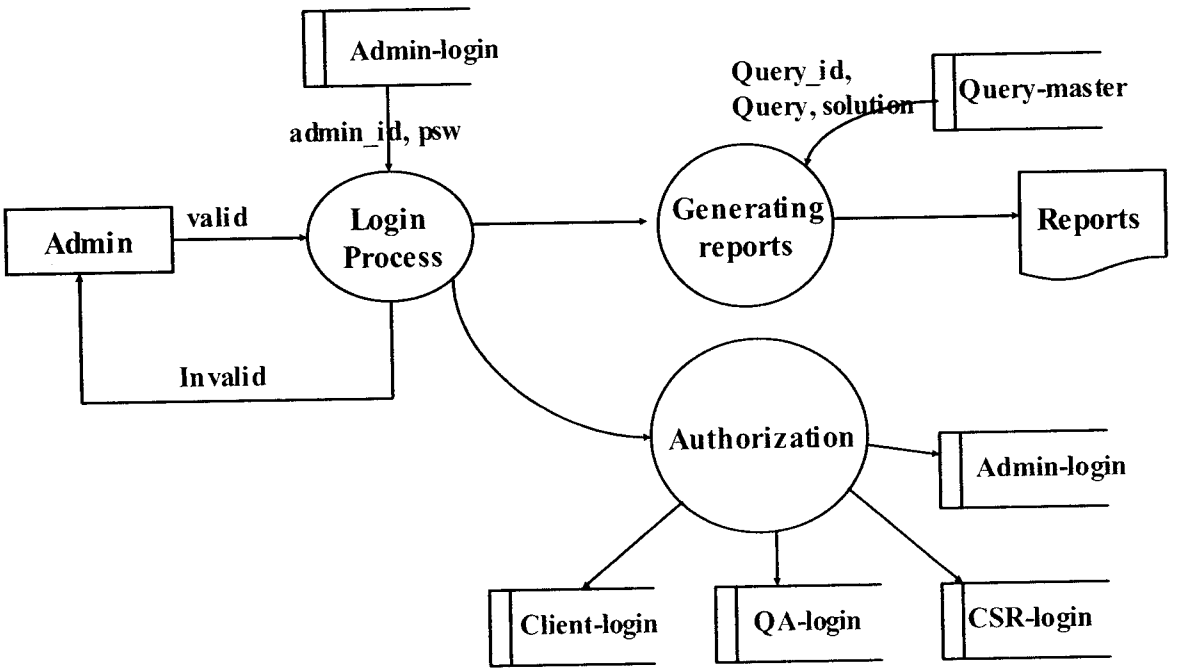

Fig 3.5. Level 1 DFD-Administrator

3.3 INPUT DESIGN

Input design is part of the overall system design which requires lot of attention. Computers require necessary data for their functioning. Hence, input design becomes an essential part of computer-oriented system. The activity of putting data into the computer for processing can be activated by instructing the computer to read data from a written printed document or it can occur by keying data directly into the system.

The goal of designing input data is to make data entry easy, logical and free from errors, avoid delays extra steps, and keeping the process simple. Input are raw data that are accepted by the system and are processed to produce the output will magnify these errors.

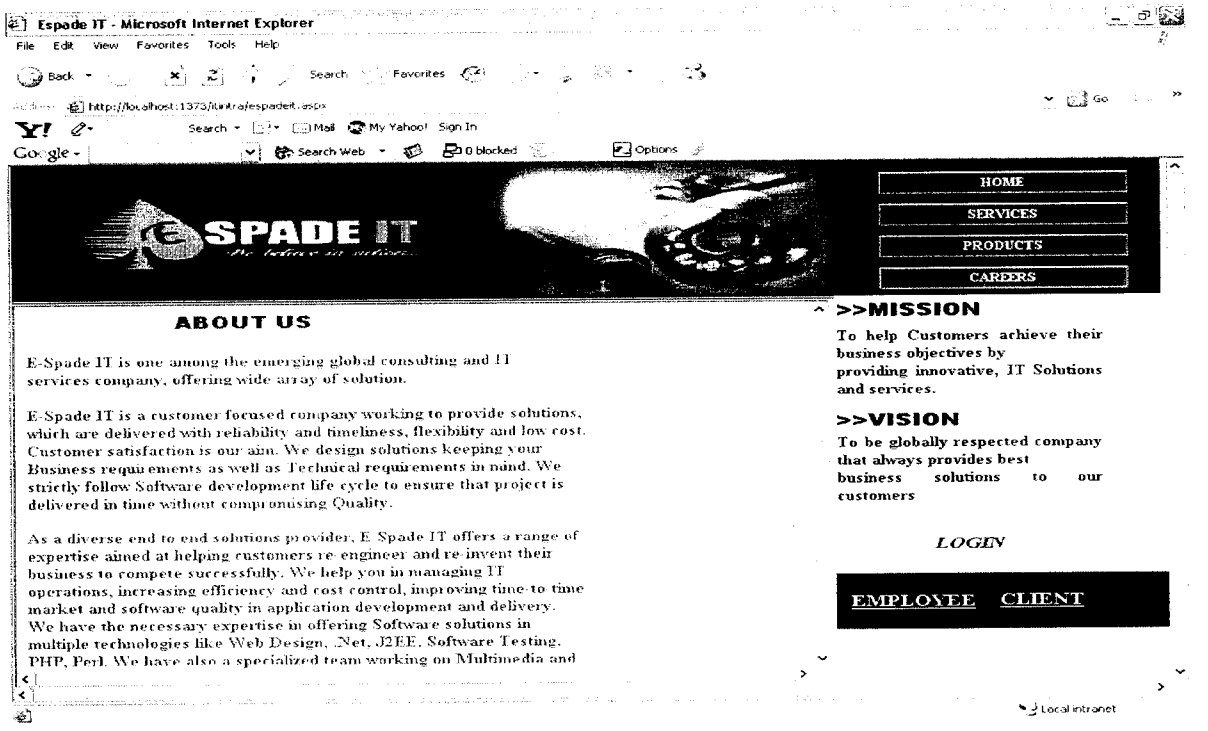

Fig 3.6. Main Screen

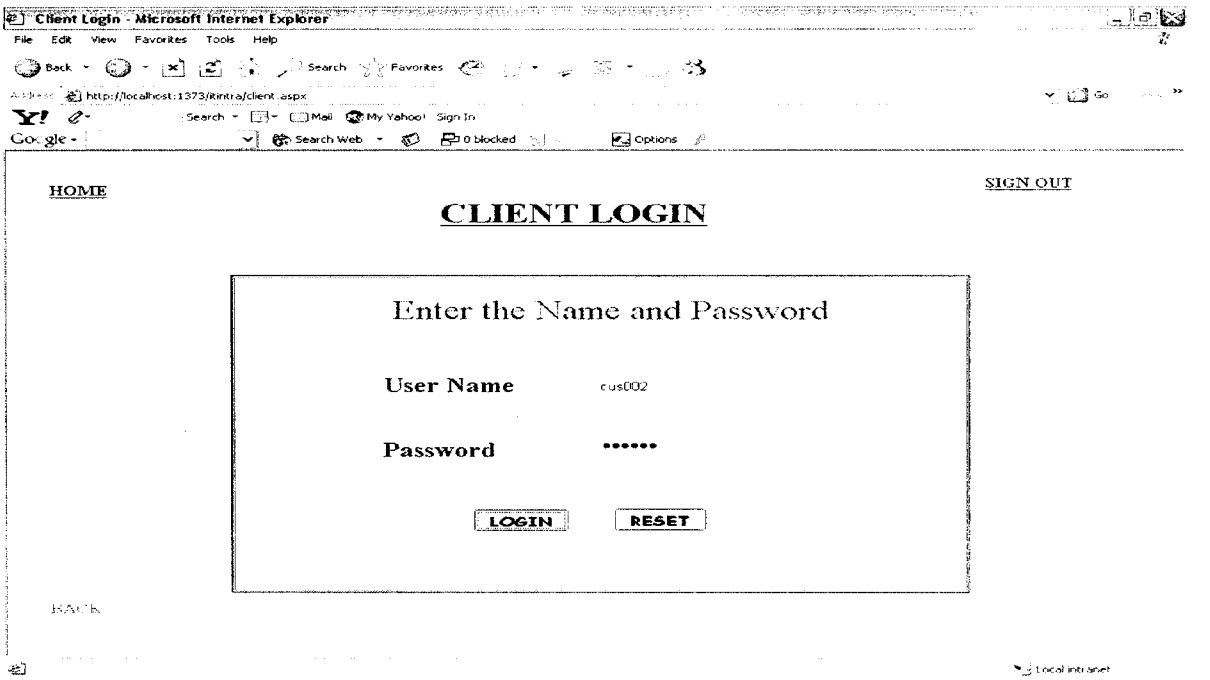

Fig 3.7. Client Login Screen

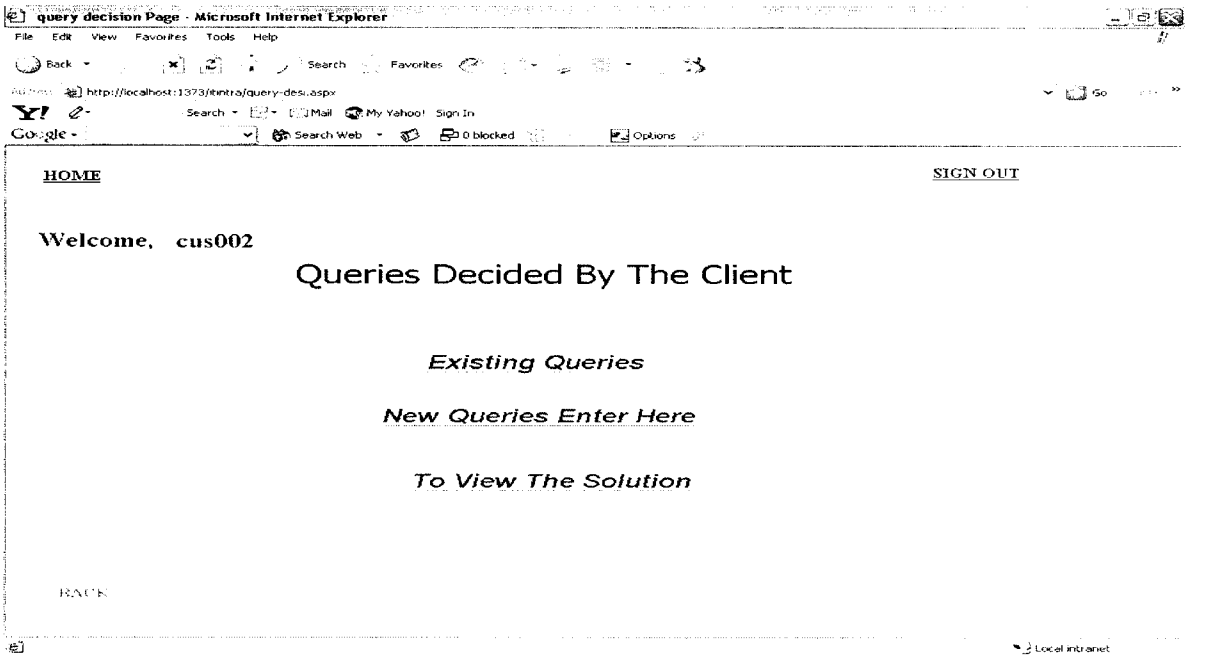

Fig 3.8. Main Screen for Client

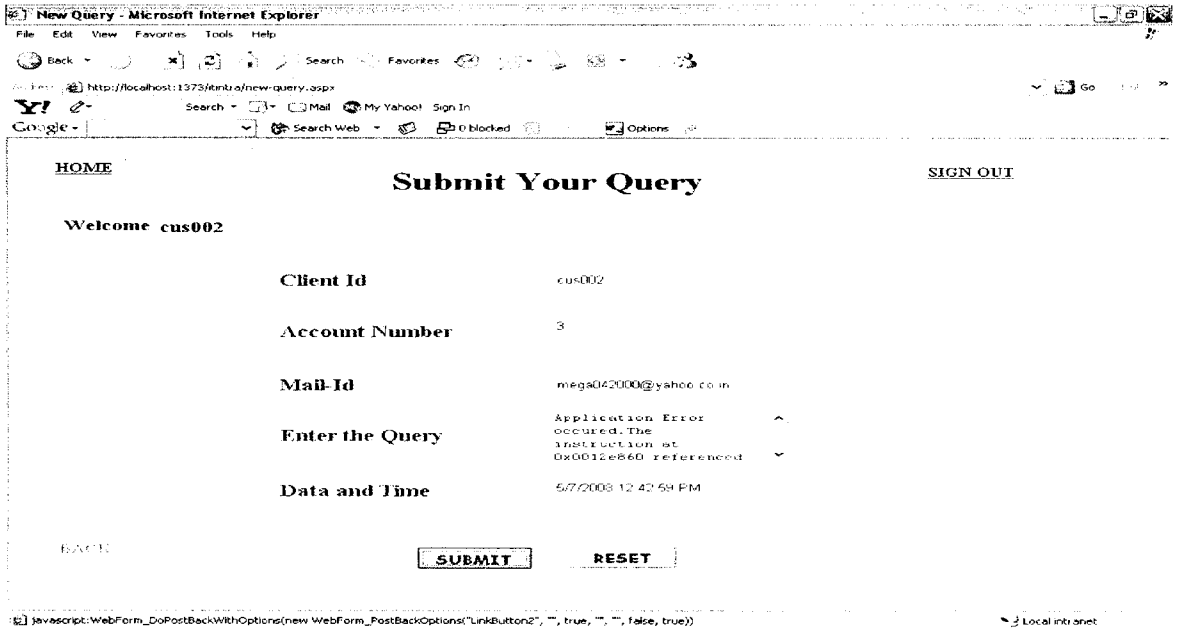
New Query - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
http://localhost:1373/dmtra/new-query.aspx
Go
Search Mail My Yahoo! Sign In
Google Search Web 0 blocked Options
HOME
Submit Your Query
SIGN OUT
Welcome cus002
Client Id
cus002
Account Number
3
Mail-Id
mega042000@yahoo.co.in
Enter the Query
Application Error occured. The instruction at 0x0012e860 referenced
Data and Time
5/7/2008 12:40:59 PM
BACK
SUBMIT
RESET
javascript:WebForm_DoPostBackWithOptions(new WebForm_PostBackOptions("LinkButton2", "", true, "", "", false, true))
Local intranet

Fig 3.9. New Query Submit Screen

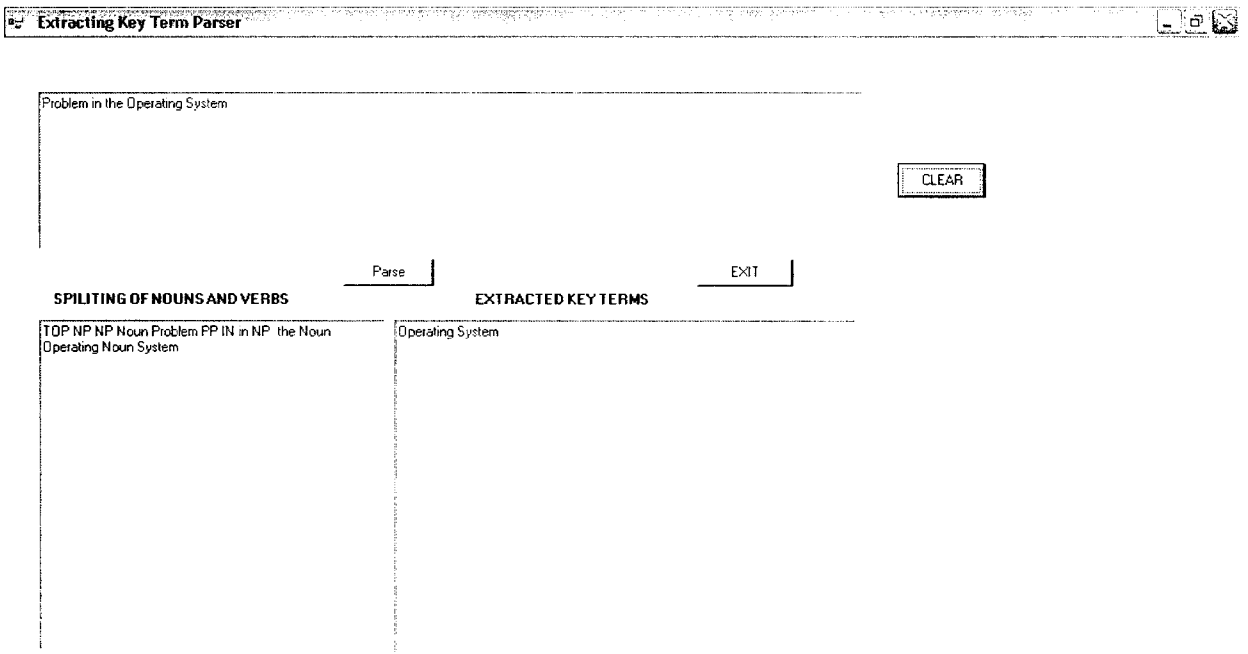

Fig 3.10. Key Word Finding Screen

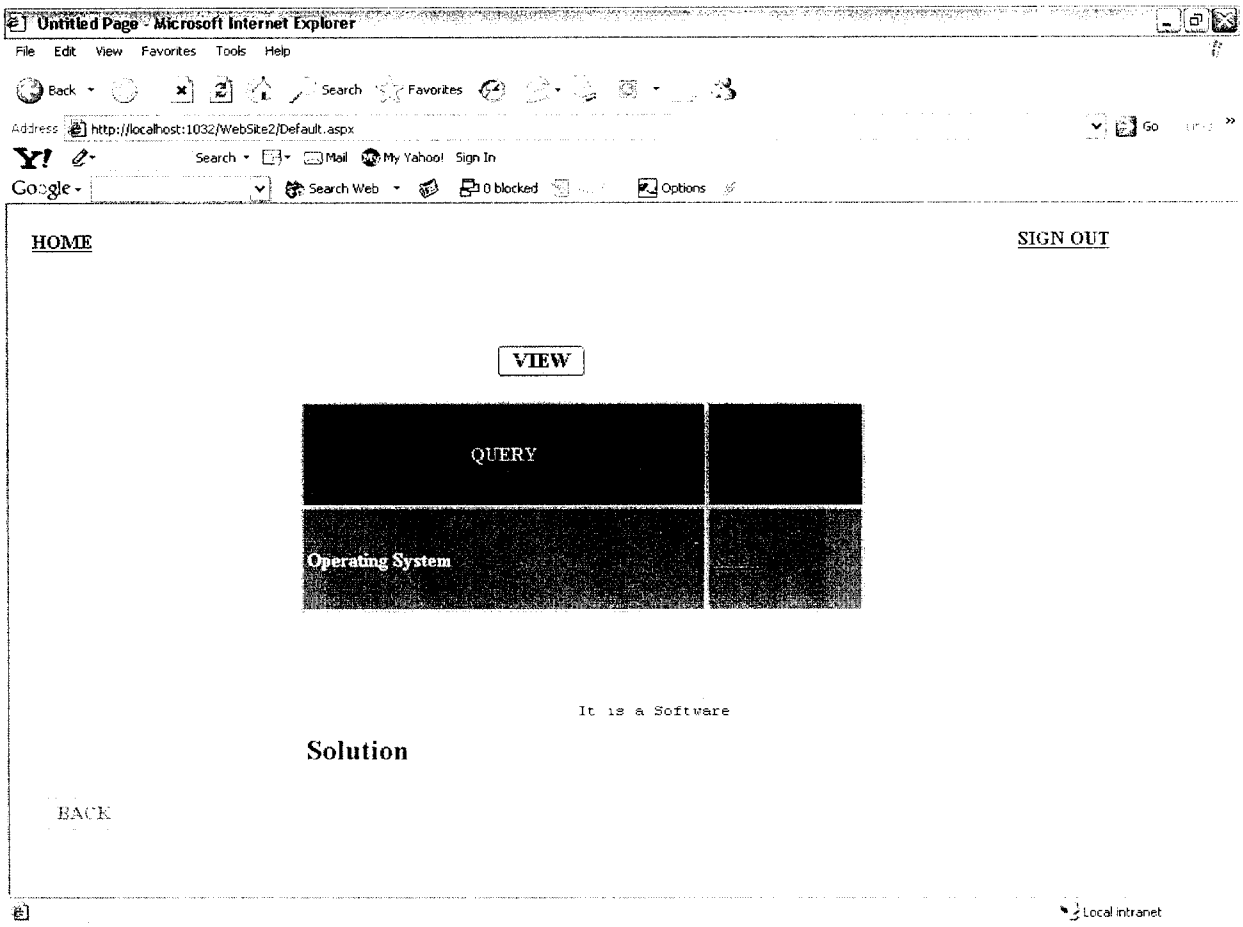

Fig 3.11. Solution Screen

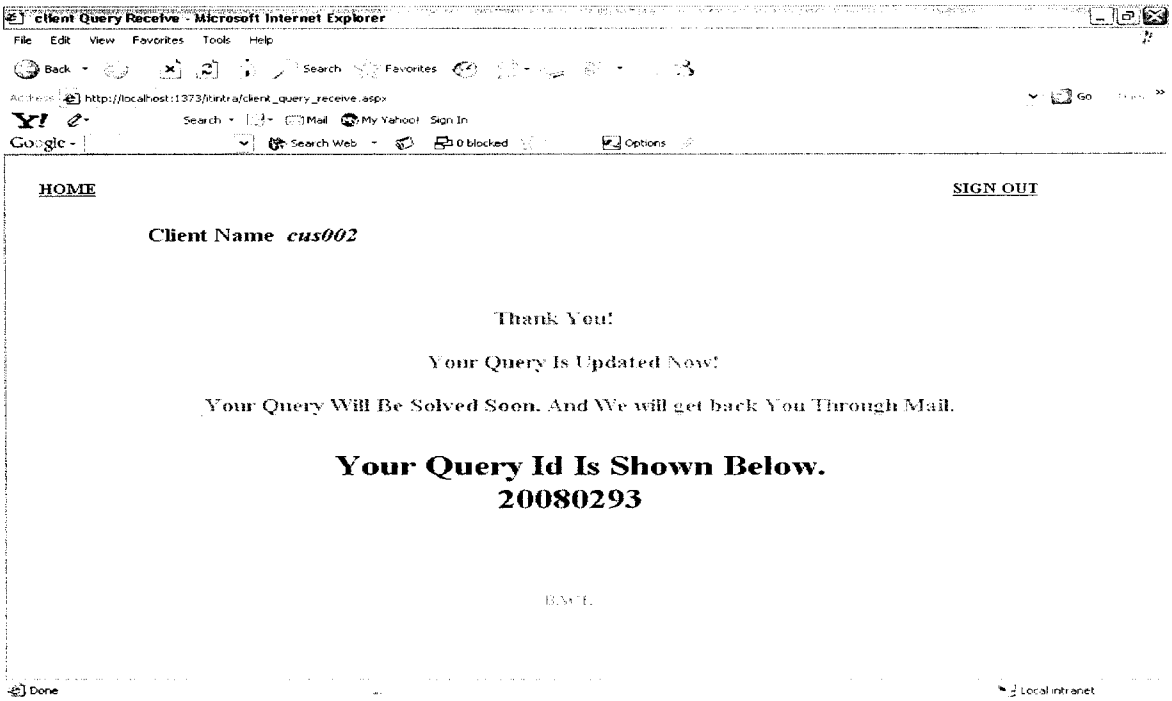

Fig 3.12. Query ID Receive Screen

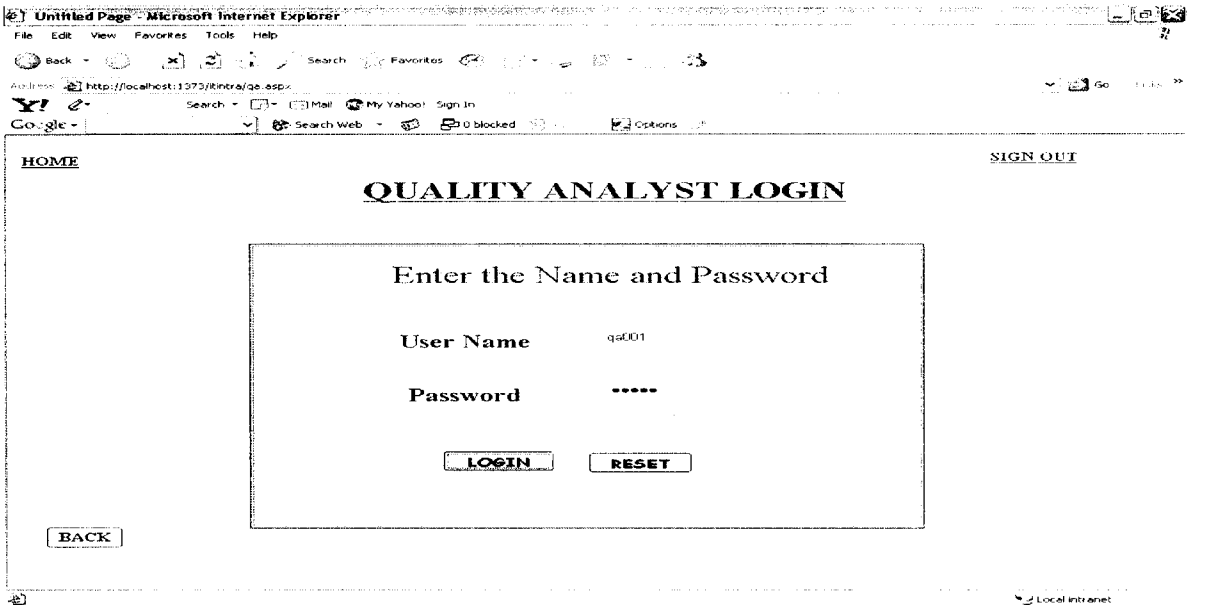

Fig 3.13. Quality Analyst Login Screen

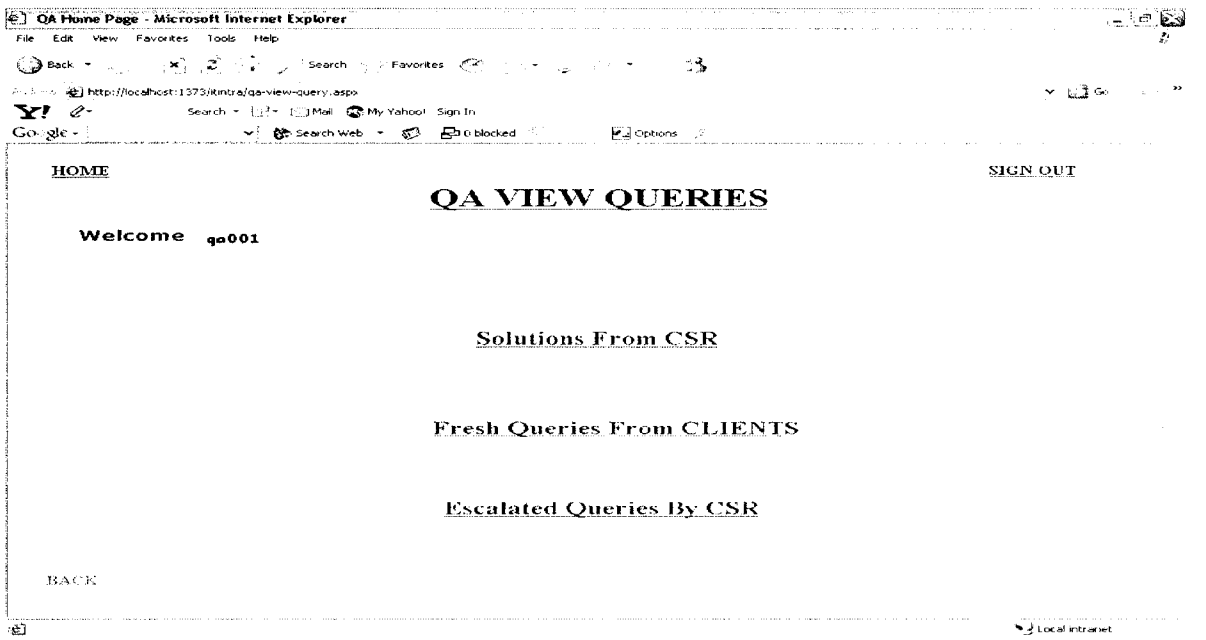

Fig 3.14. Quality Analyst Main Screen

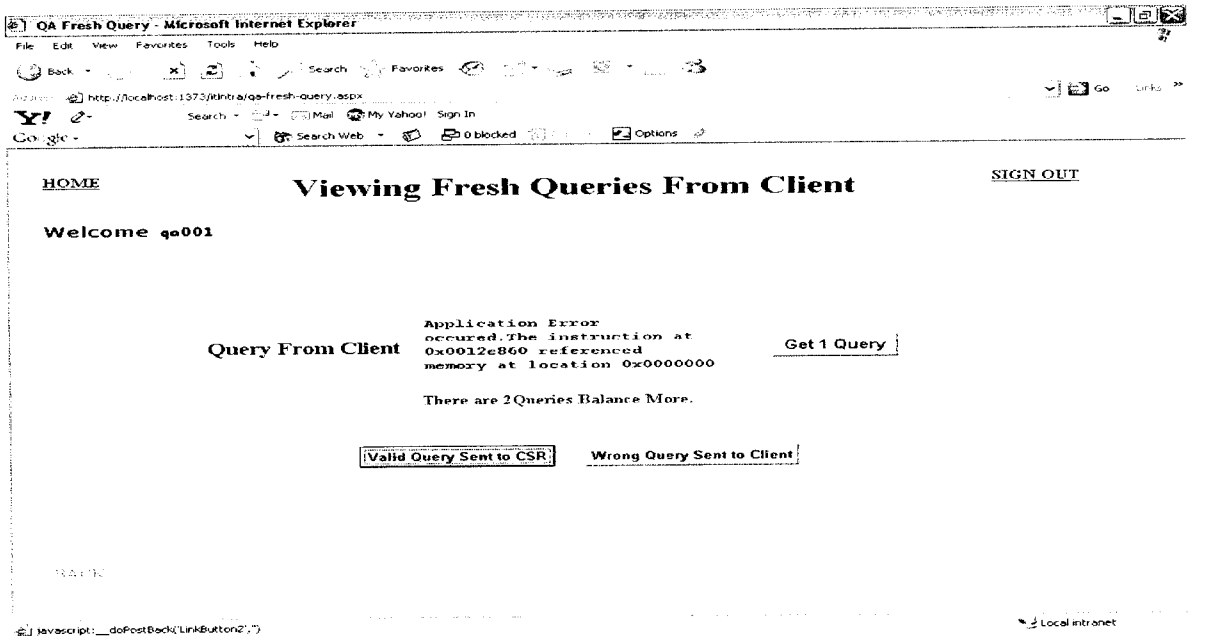

Fig 3.15. QA View Fresh Query

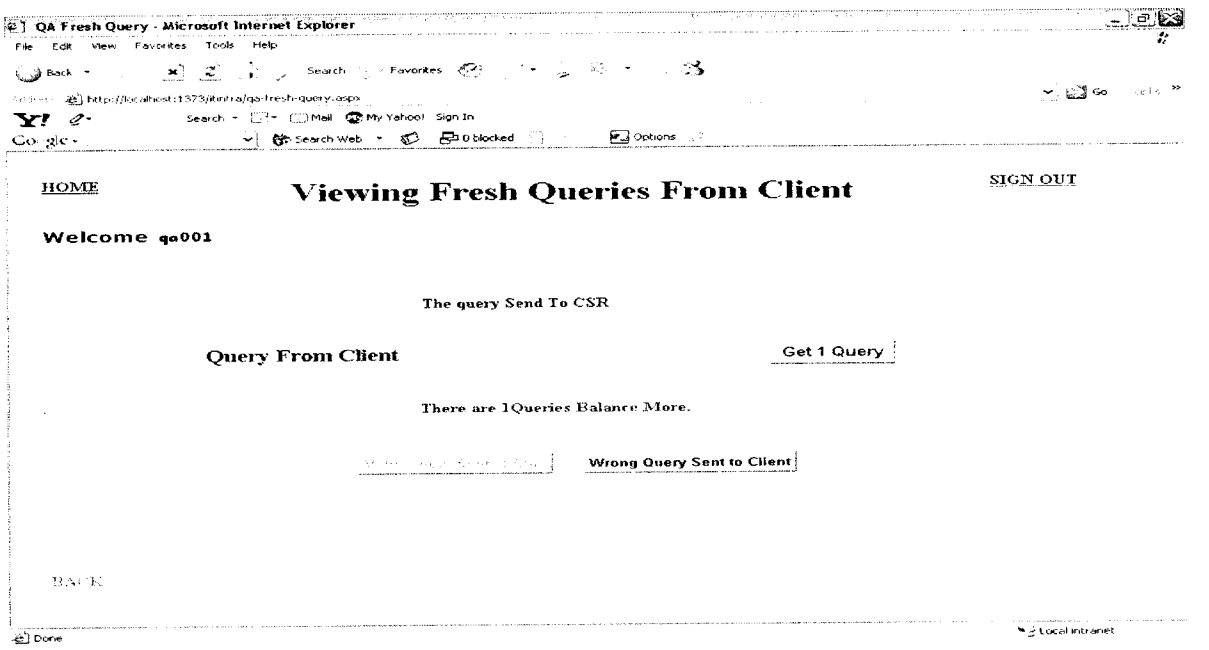

Fig 3.16. Query Send to CSR

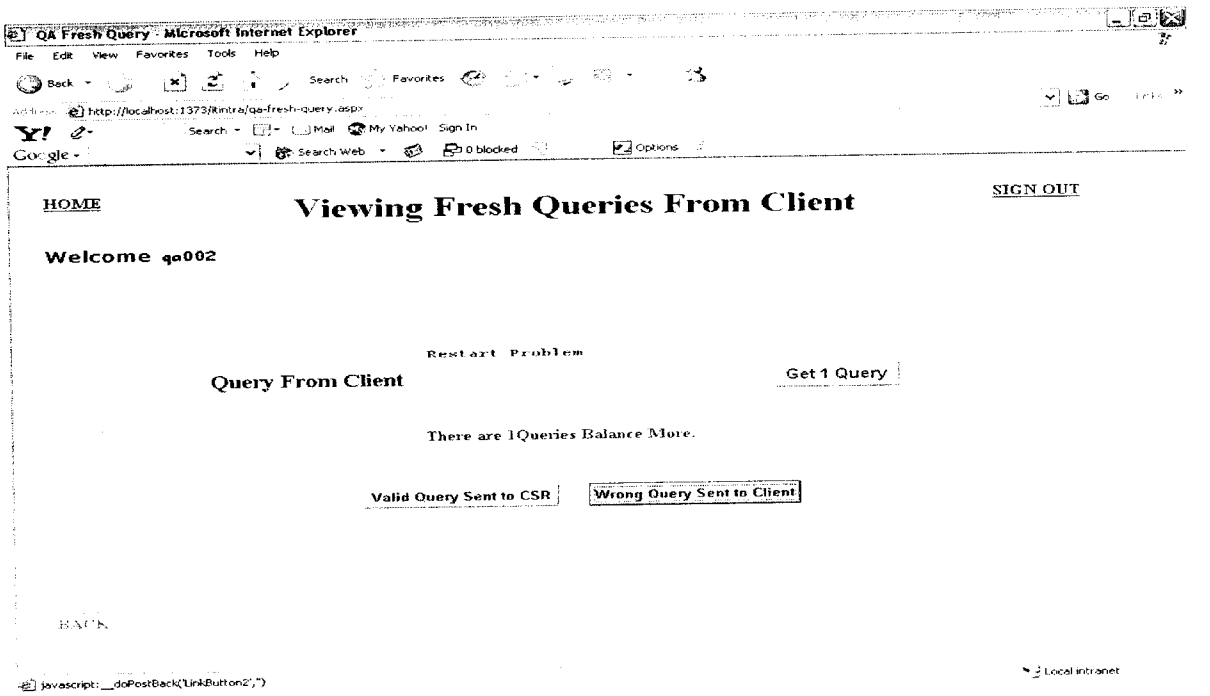

Fig 3.17. QA View Fresh Query

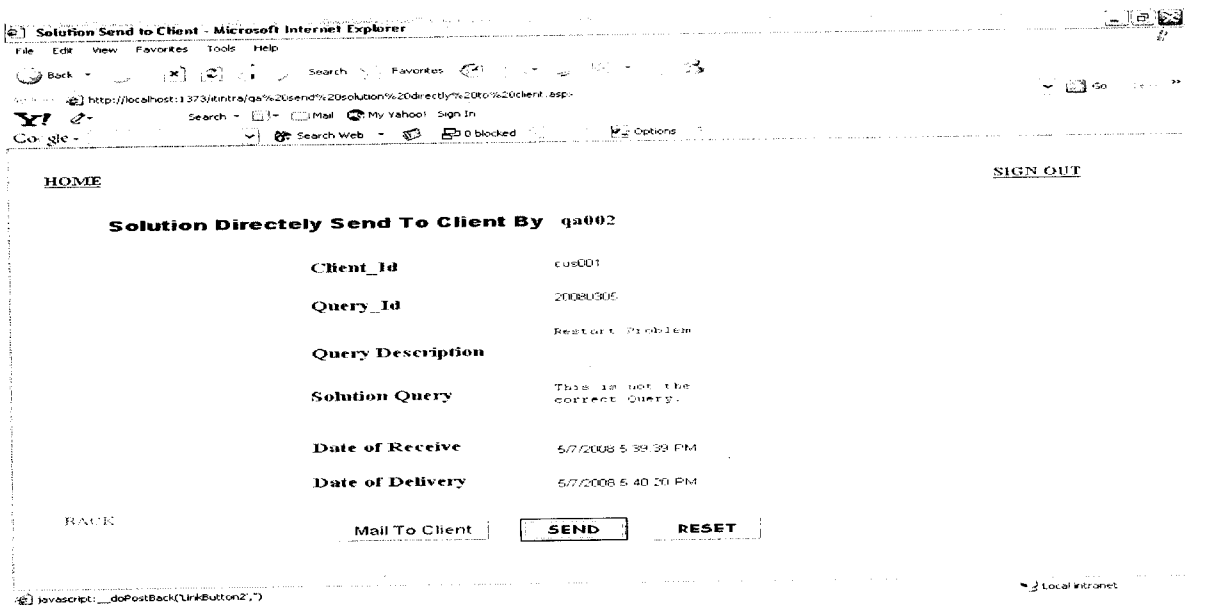

Fig 3.18. Wrong Query Send to Client

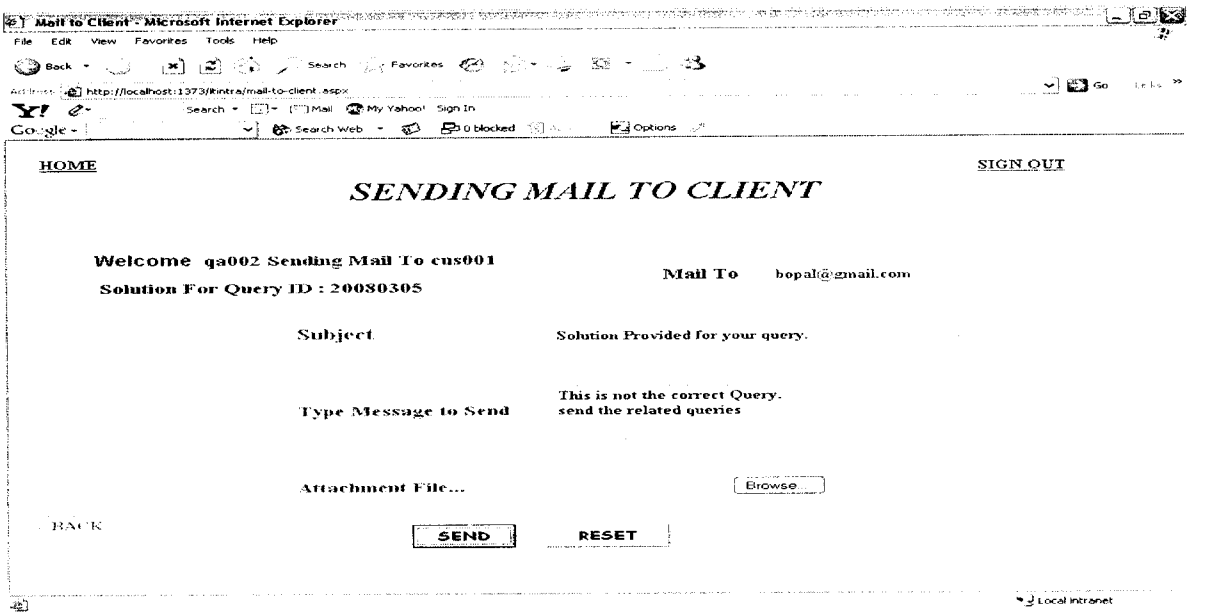

Fig 3.19. QA Directly Send Mail to Client

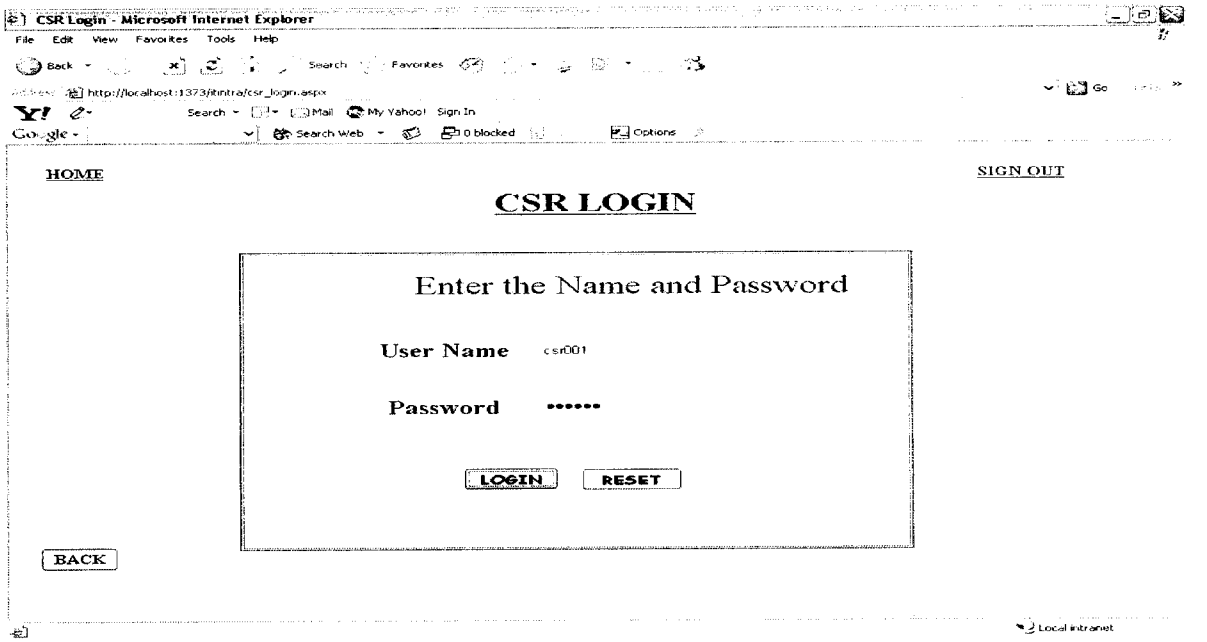

Fig 3.20. CSR Login Screen

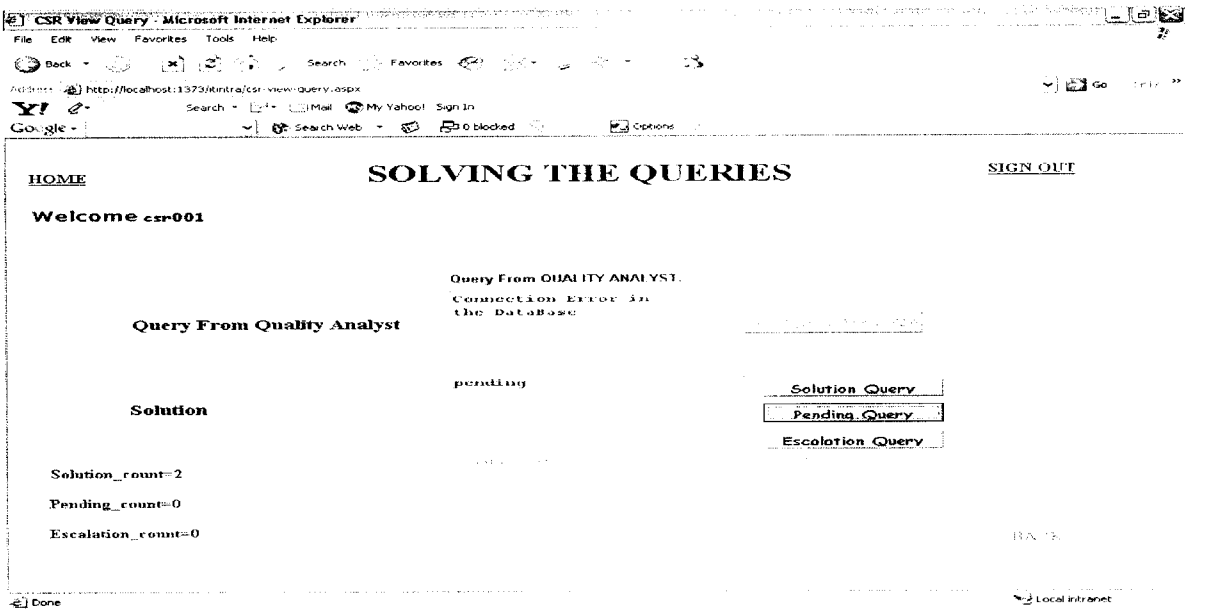

Fig 3.21. CSR View Query

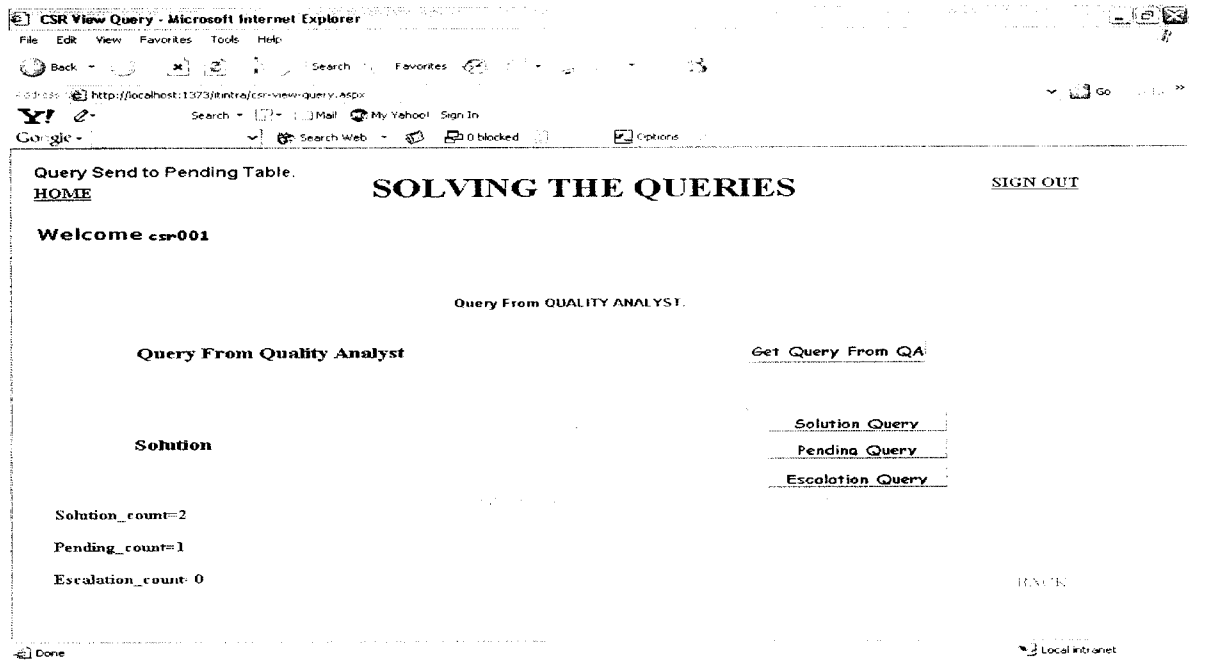

Fig 3.22. Query Send to Pending Table

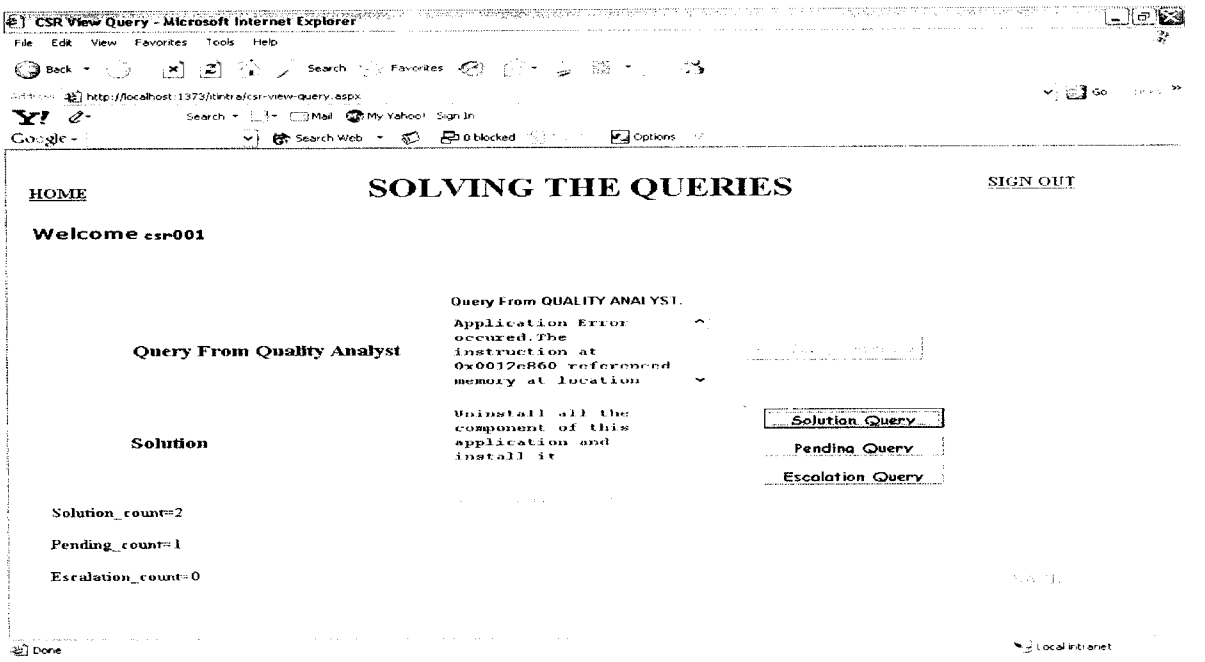

Fig 3.23. CSR Finding Solution

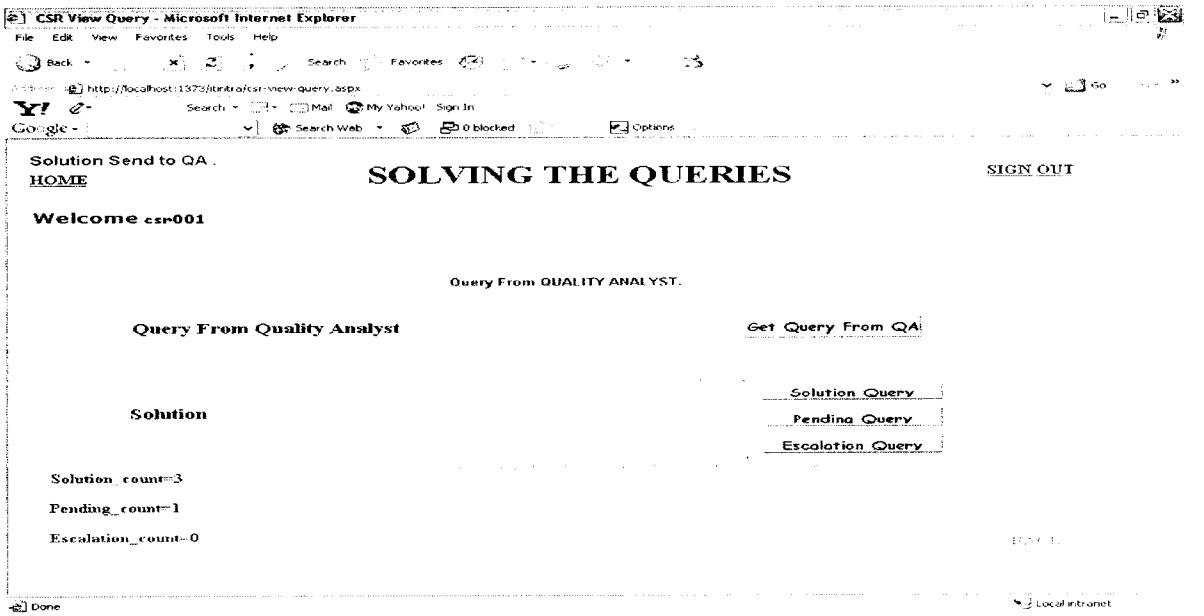

Fig 3.24. Query Send to Solution Table

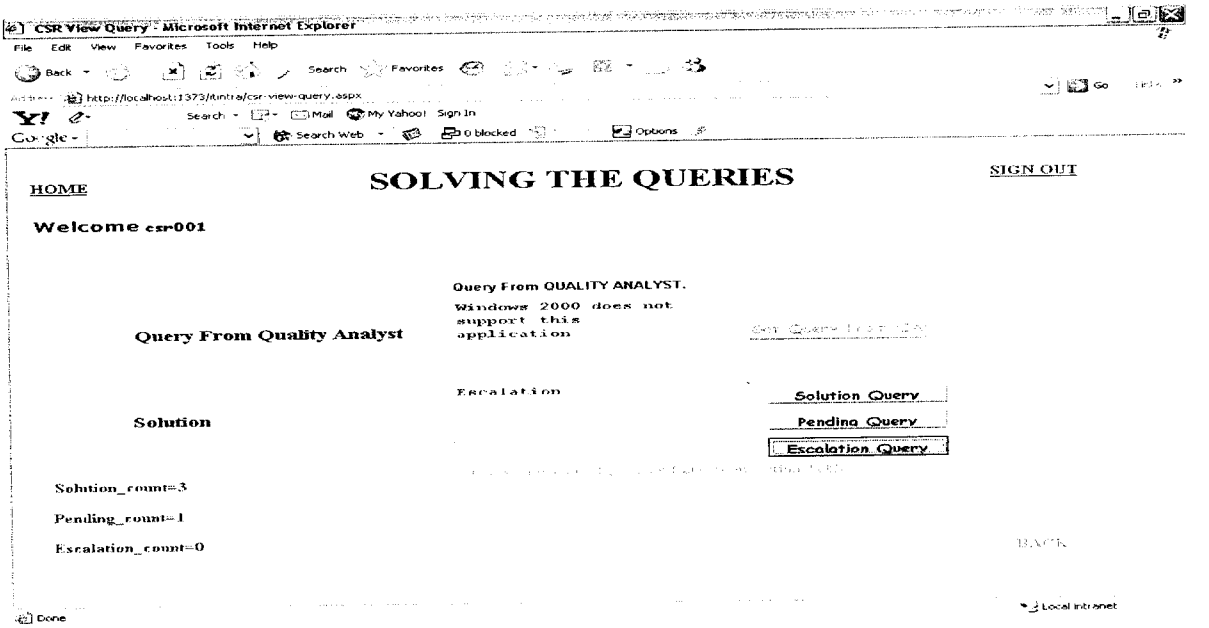

Fig 3.25. CSR Viewing Query

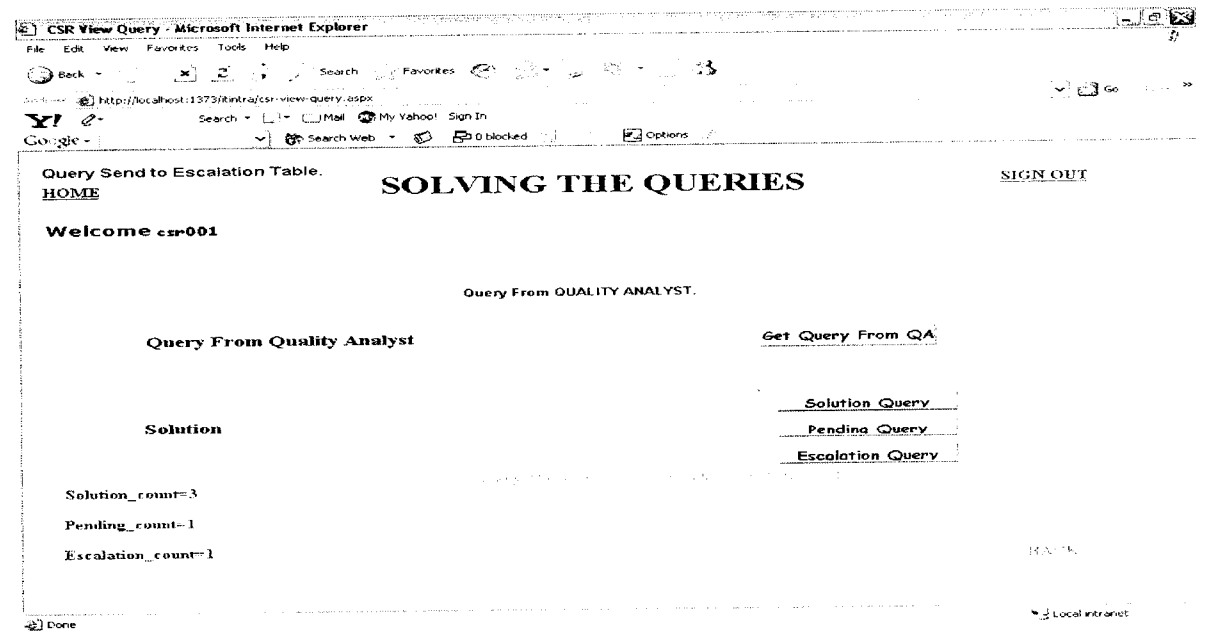

Fig 3.26. Query Send to Escalation Table

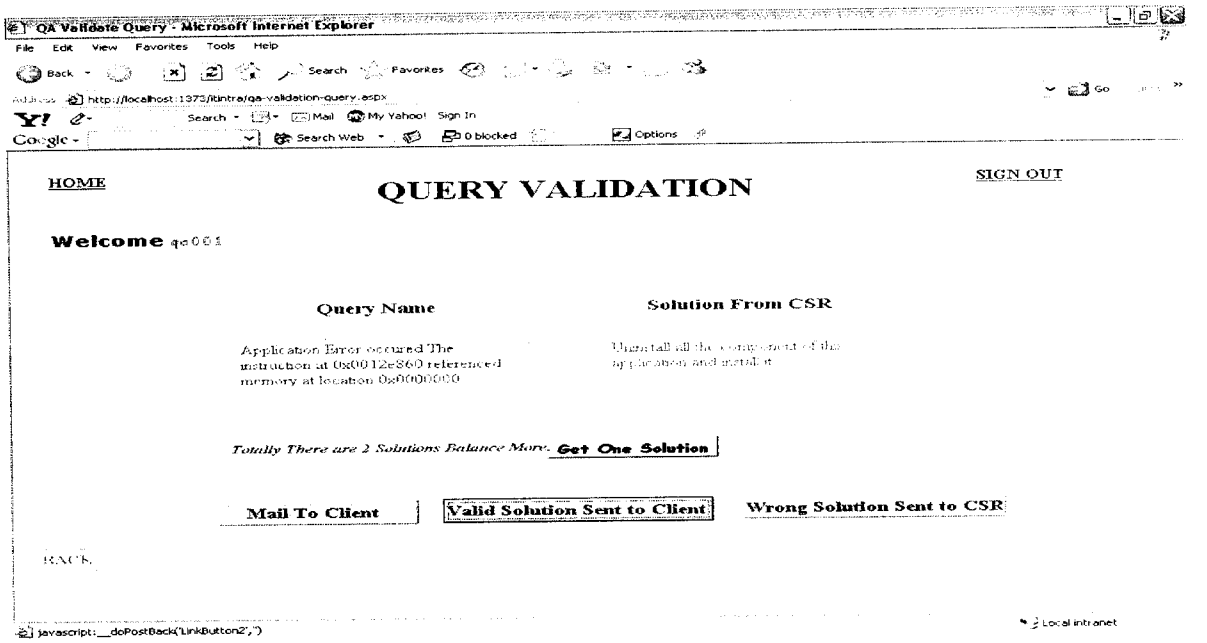

Fig 3.27. QA Validation the Solution Query

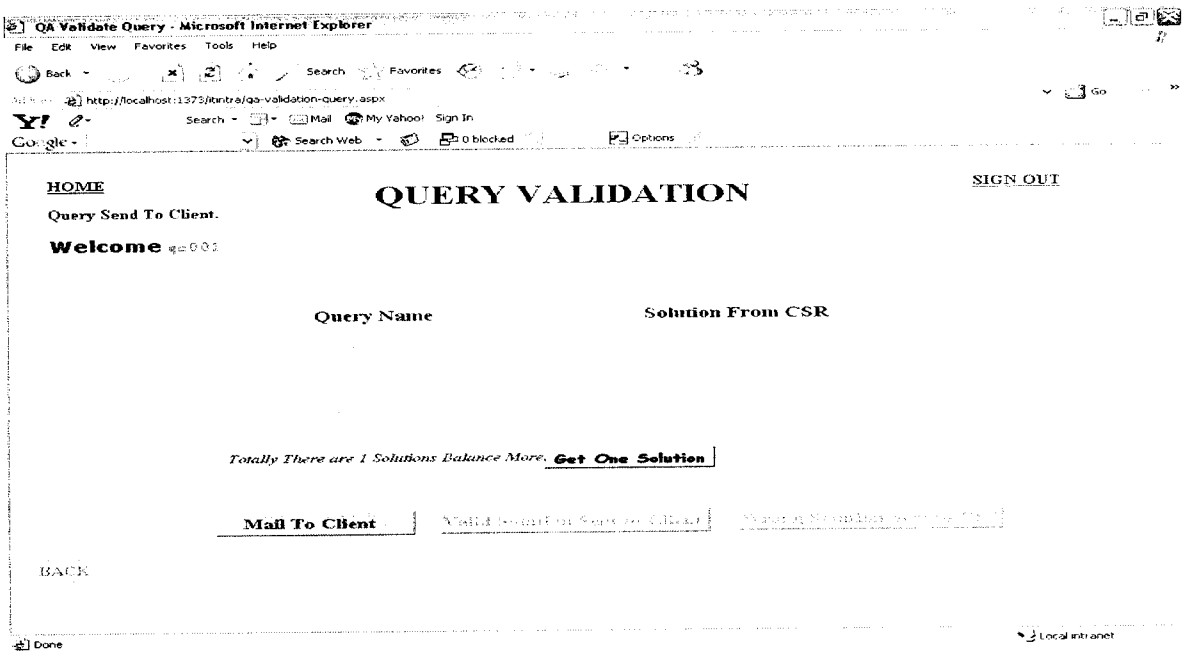

Fig 3.28. Query Send to Client

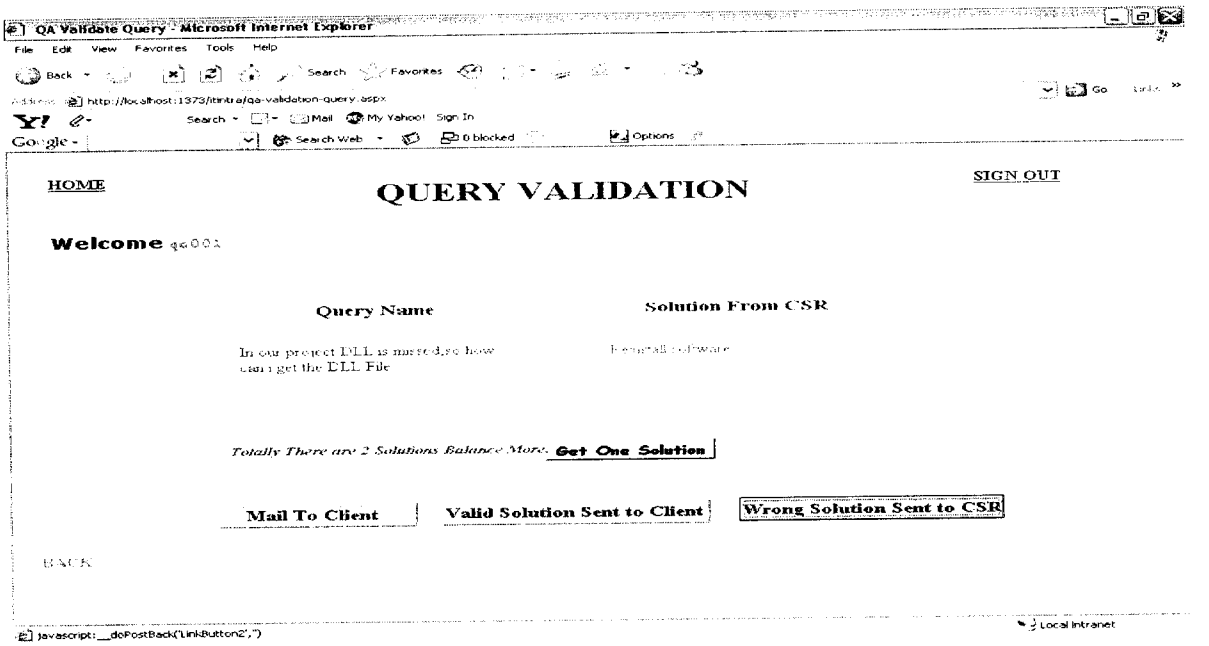

Fig 3.29. QA Viewing wrong Query

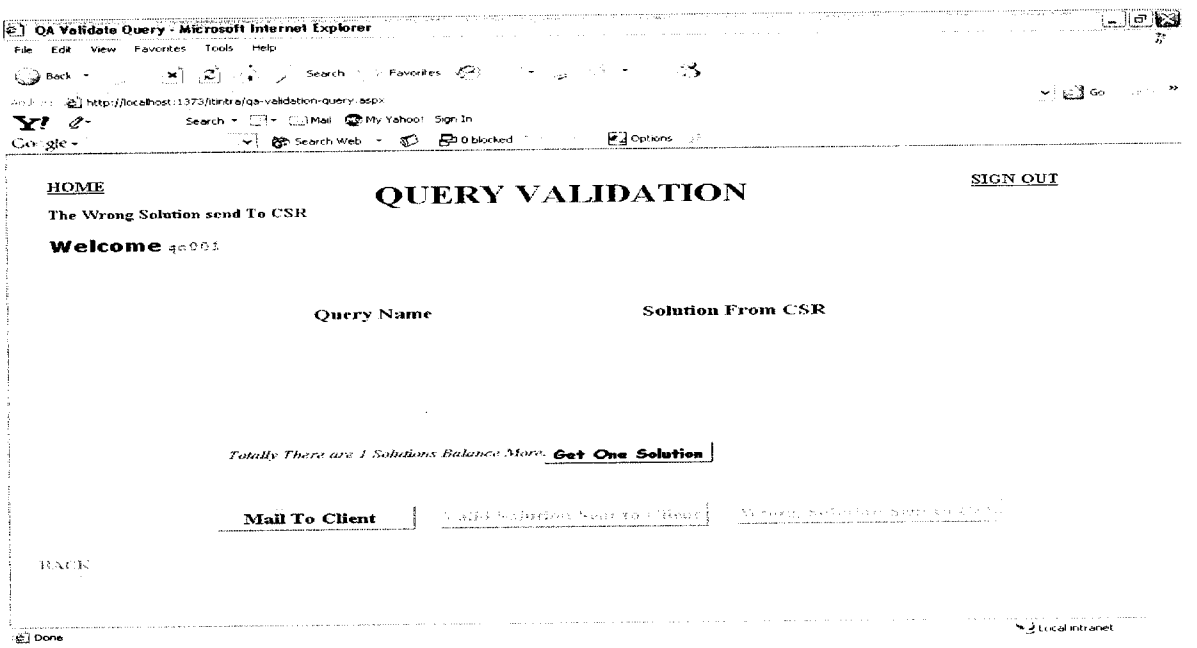

Fig 3.30. Wrong Query Send to CSR

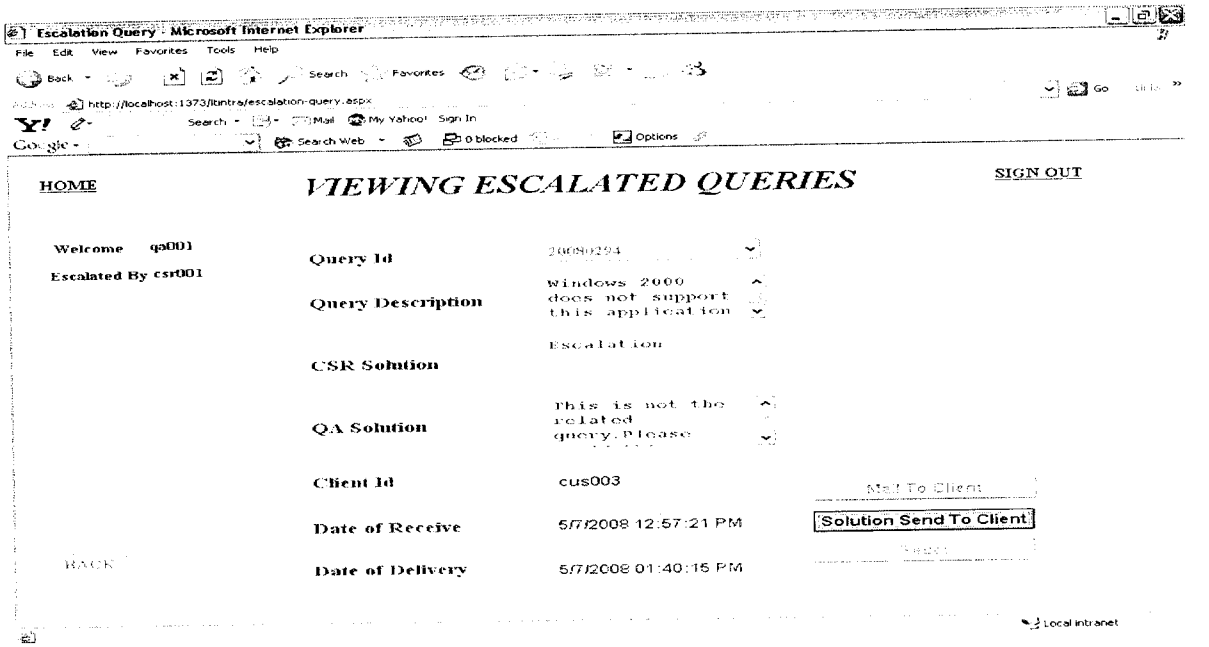

Fig 3.31. QA Viewing Escalation Query

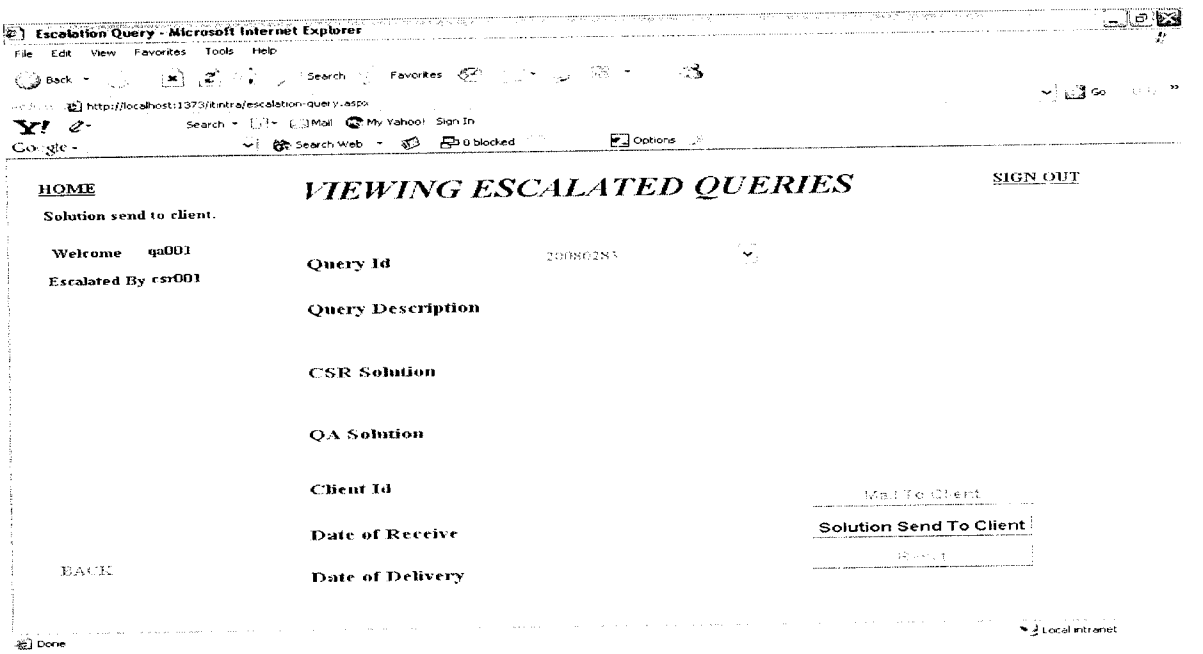

Fig 3.32. Wrong Query Send to Client

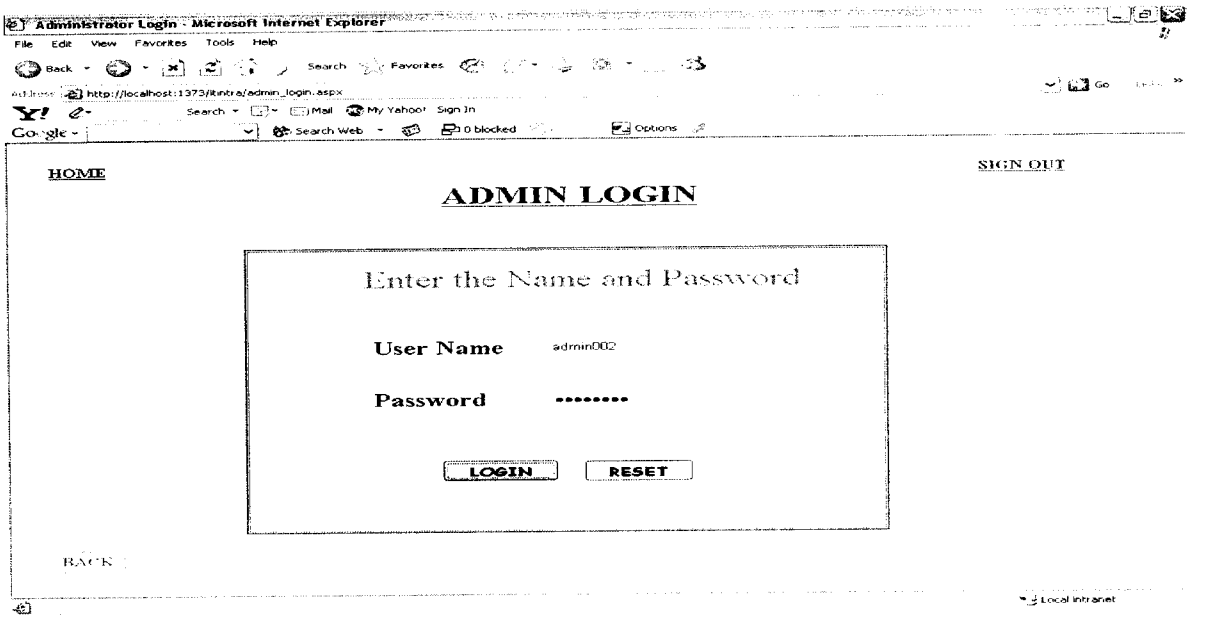

Fig 3.33. Admin Login Screen

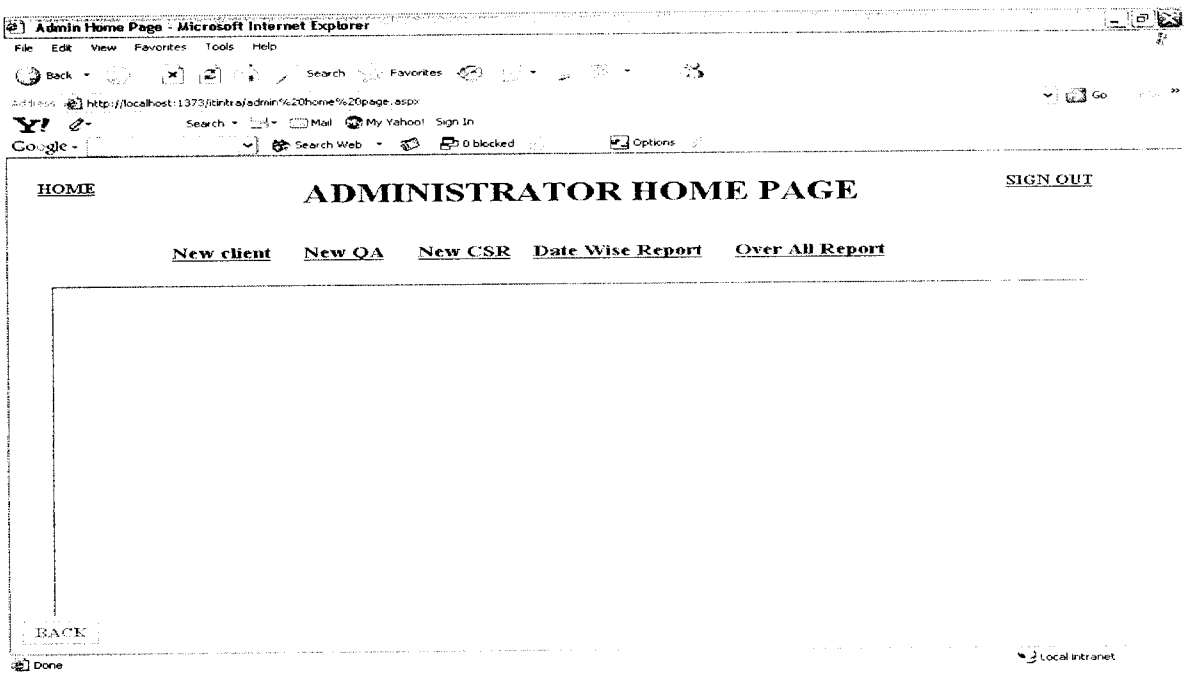

Fig 3.34. Admin Home Page

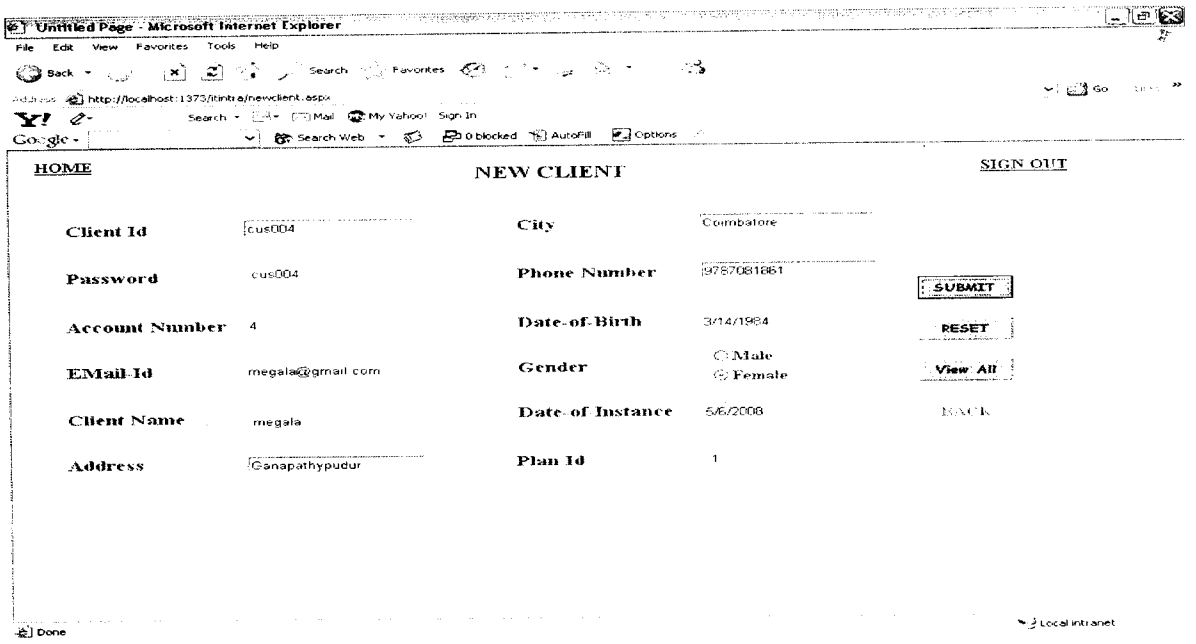

Fig 3.35. New Client Registration

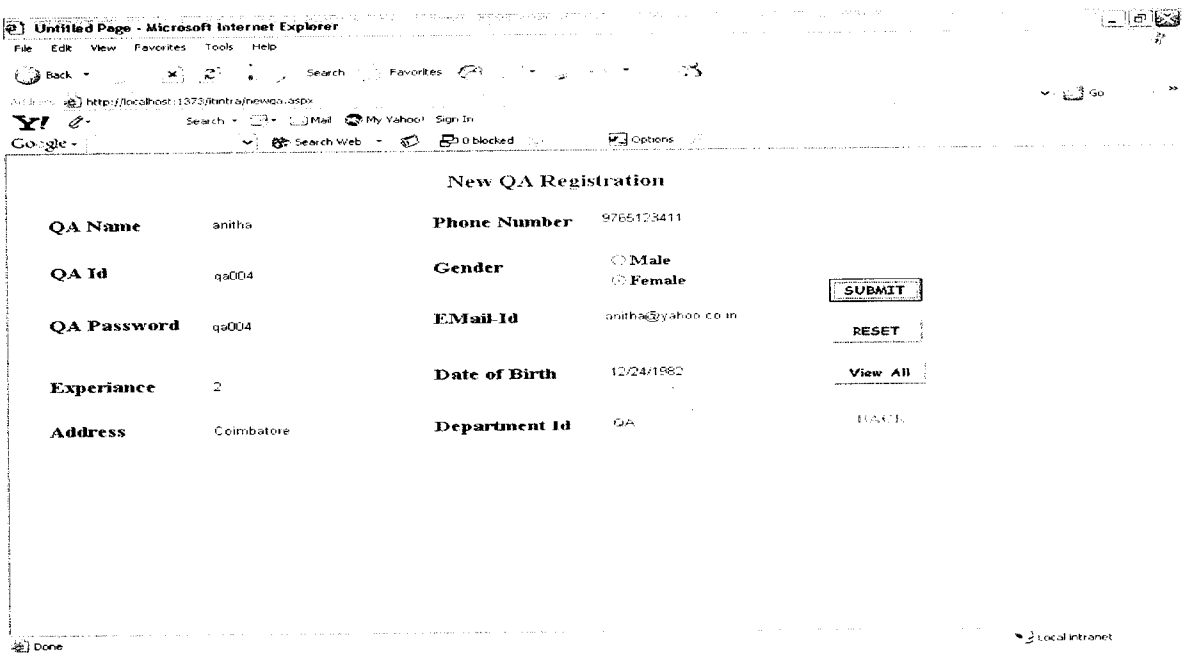

Fig 3.36. New QA Registration

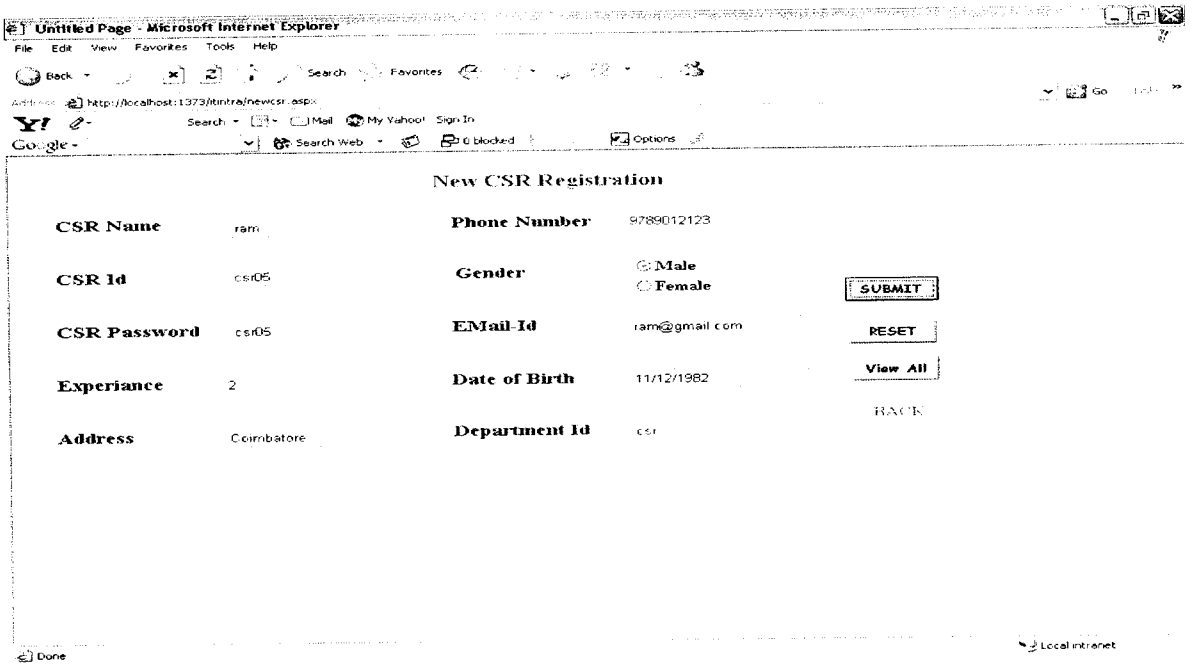

Fig 3.37. New CSR Registration

3.4 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any systems results of processing are communicated to the user and to other systems through outputs. In the output design it is determined how the information is to be displayed for immediate needs. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship with the user and helps in decision-making.

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.

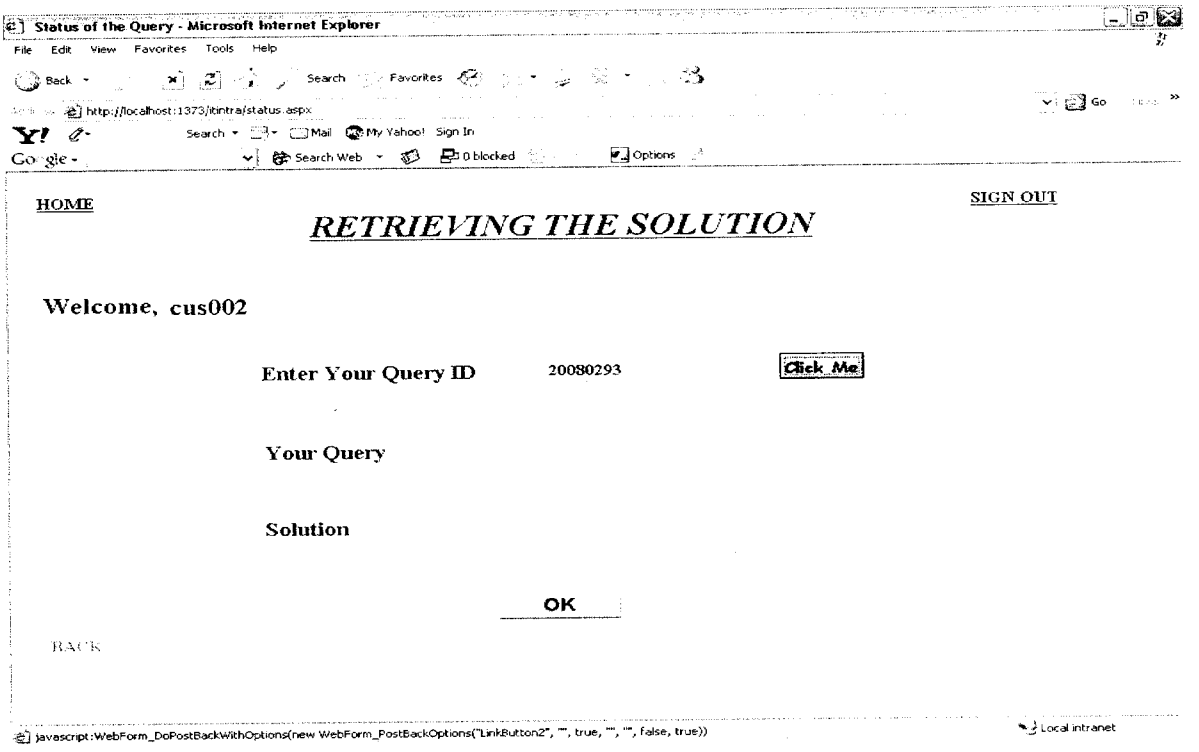

Fig 3.38. Query ID Screen

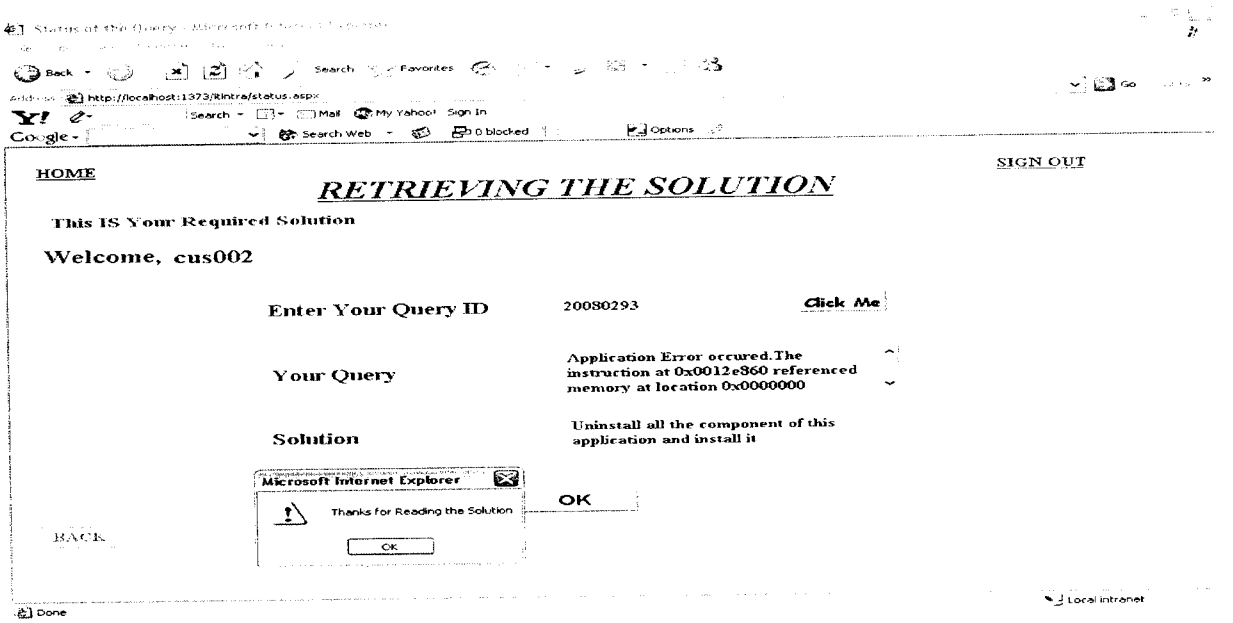

Fig 3.39 Client Viewing the Solution

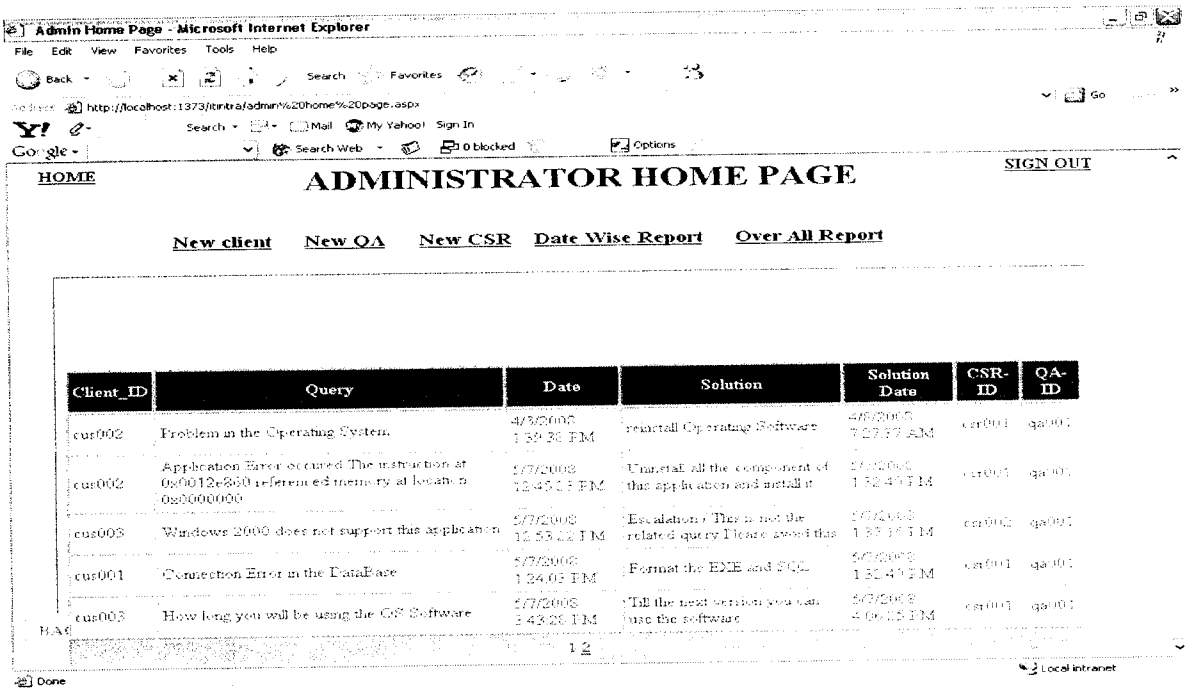

Fig 3.40 Over All Report

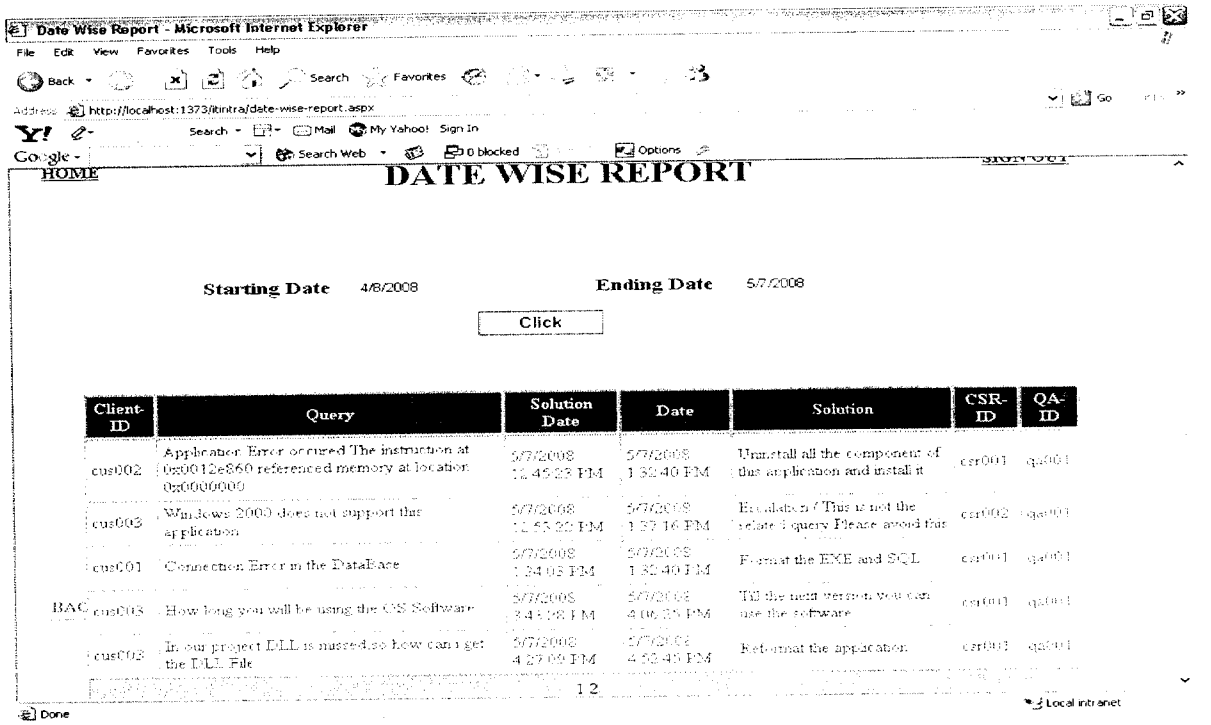

Fig 3.41 Date Wise Report

3.5 DATABASE DESIGN

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

NORMALIZATION

Normalization, sometimes referred to as canonical synthesis, is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. When multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity.

A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

Accordingly, more highly normalized tables are typically used in database applications involving many isolated transactions , while less normalized tables tend to be used in database applications that need to map complex relationships between data entities and data attributes.

A table that is not sufficiently normalized can suffer from logical inconsistencies of various types, and from anomalies involving data operations. In such a table:

- The same information can be expressed on multiple records; therefore updates to the table may result in logical inconsistencies. This leads to the update anomaly.

- There are circumstances in which certain facts cannot be recorded at all which leads to the insertion anomaly.

- There are circumstances in which the deletion of data representing certain facts necessitates the deletion of data representing completely different facts which leads to the deletion anomaly.

Ideally, a relational database table should be designed in such a way as to exclude the possibility of update, insertion, and deletion anomalies.

The normal forms of relational database theory provide guidelines for deciding whether a particular design will be vulnerable to such anomalies. It is possible to correct an unnormalized design so as to make it adhere to the demands of the normal forms: this is called normalization.

Normalization typically involves decomposing an unnormalized table into two or more tables that were they to be combined (joined), would convey exactly the same information as the original table.

Table design

Name : Client_Master

Description : Used to store Client details

Fields Name	Type	Constraints	Description
Client_Id	Varchar(10)	Primary Key	Client ID
Client_psw	Varchar(8)	Foreign Key	Client Password
Name	Varchar(20)	Not Null	Client Name
Mail-Id	Varchar(20)	Not Null	Client Mail ID
DOB	Date	Not Null	Date of Birth
Gender	Varchar(1)	Not Null	Gender
Plan-ID	Number(4)	Not Null	Plan ID
Phone	Varchar(10)	Not Null	Client Phone Number
Address	Varchar(25)	Not Null	Client Address
City	Varchar(15)	Not Null	City

Table 3.1. Client_Master

Name : Client_Account

Description : Table contain Client account number, and Client ID

Fields Name	Type	Constraints	Description
Client_acc_no	Number(5)	Primary Key	Client Account Number
Client_Id	Varchar(10)	Foreign Key	Client ID

Table 3.2. Client_Account

Name : Client_Login

Description : Used to maintain client confidential

Fields Name	Type	Constraints	Description
Client_Id	Varchar(10)	Foreign Key	Client ID
Client_psw	Varchar(8)	Primary Key	Client password

Table 3.3. Client_Login

Name : CSR_Login

Description : Used to maintain Customer Support Representative confidential

Fields Name	Type	Constraints	Description
CSR_Id	Varchar(10)	Foreign Key	CSR ID
CSR_psw	Varchar(8)	Primary Key	CSR password

Table 3.4. CSR_Login

Name : CSR_Personal

Description : Used to store Customer Support Representative Details

Fields Name	Type	Constraints	Description
CSR_Id	Varchar(10)	Primary Key	CSR ID
Name	Varchar(20)	Not Null	CSR Name
Mail-Id	Varchar(20)	Not Null	CSR Mail ID
DOB	Date	Not Null	Date of Birth
Gender	Varchar(1)	Not Null	Gender
Experience	Number(2)	Not Null	Experience
Phone	Varchar(10)	Not Null	CSR Phone Number

Address	Varchar(25)	Not Null	CSR Address
City	Varchar(15)	Not Null	City

Table 3.5. CSR_Personal

Name : QA_Login

Description : Used to maintain Quality Analyst confidential

Fields Name	Type	Constraints	Description
QA_Id	Varchar(10)	Foreign Key	Quality Analyst ID
QA_psw	Varchar(8)	Primary Key	Quality Analyst password

Table 3.6. QA_Login

Name : QA_Personal

Description : Used to store Quality Analyst details

Fields Name	Type	Constraints	Description
QA_Id	Varchar(10)	Primary Key	QA ID
Name	Varchar(20)	Not Null	QA Name
Mail-Id	Varchar(20)	Not Null	QA Mail ID
DOB	Date	Not Null	Date of Birth
Gender	Varchar(1)	Not Null	Gender
Experience	Number(2)	Not Null	Experience
Phone	Varchar(10)	Not Null	QA Phone Number
Address	Varchar(25)	Not Null	QA Address
City	Varchar(15)	Not Null	City

Table 3.7. QA_Personal

Name : Admin_Login

Description : Used to maintain Administrator confidential

Fields Name	Type	Constraints	Description
Admin_Id	Varchar(10)	Foreign Key	Administrator ID
Admin_psw	Varchar(8)	Primary Key	Administrator password

Table 3.8. Admin_Login

Name : Admin_Personal

Description : Used to store Administrator details

Fields Name	Type	Constraints	Description
Admin_Id	Varchar(10)	Primary Key	Admin ID
Name	Varchar(20)	Not Null	Admin Name
Mail-Id	Varchar(20)	Not Null	Admin Mail ID
DOB	Date	Not Null	Date of Birth
Gender	Varchar(1)	Not Null	Gender
Experience	Number(2)	Not Null	Experience
Phone	Varchar(10)	Not Null	Admin Phone Number
Address	Varchar(25)	Not Null	Admin Address
City	Varchar(15)	Not Null	City

Table 3.9. Admin_Personal

Name : Query_Master

Description : Used to store Query details

Fields Name	Type	Constraints	Description
Query_ID	Number(8)	Primary Key	Query ID
Client_ID	Varchar(10)	Foreign Key	Client ID
Query_Desc	Varchar(600)	Not Null	Query Description
Date_Submit	DateTime	Not Null	Date of Submit
status	Number(1)	Allow Null	Status
Solution_Query	Varchar(600)	Allow Null	Solution Query
Date_Sol	DateTime	Allow Null	Solution Date
CSR_ID	Varchar(10)	Foreign Key	CSR ID
QA_ID	Varchar(10)	Foreign Key	QA ID
Final_Solution	Varchar(600)	Allow Null	Final Solution

Table 3.10. Query_Master

Name : Current_Query

Description : Used to store Current Query details

Fields Name	Type	Constraints	Description
Query_ID	Number(8)	Foreign Key	Query ID
client_ID	Varchar(10)	Foreign Key	Client ID
Query_Desc	Varchar(600)	Not Null	Query Description
Query_Date	DateTime	Not Null	Date of Submit

Table 3.11. Current_Query

Name : Action_Query

Description : Used to store Action Query details

Fields Name	Type	Constraints	Description
Query_ID	Number(8)	Foreign Key	Query ID
Client_ID	Varchar(10)	Foreign Key	Client ID
Query_Desc	Varchar(600)	Not Null	Query Description
Action_Query_Date	DateTime	Not Null	Date of Submit
QA_ID	Varchar(10)	Foreign Key	Quality Analyst ID

Table 3.12. Action _Query

Name : Solution _Query

Description : Used to store Solution Query details

Fields Name	Type	Constraints	Description
Query_ID	Number(8)	Foreign Key	Query ID
Client_ID	Varchar(10)	Foreign Key	Client ID
Solution_Query_Desc	Varchar(600)	Allow Null	Solution Query Description
Solution_Query_Date	DateTime	Allow Null	Solution Query Date
QA_ID	Varchar(10)	Foreign Key	Quality Analyst ID

Table 3.13. Solution_Query

Name : Pending_Query

Description : Used to store Pending Query details

Fields Name	Type	Constraints	Description
Query_ID	Number(8)	Foreign Key	Query ID
Client_ID	Varchar(10)	Foreign Key	Client ID
Pend_Query_Desc	Varchar(600)	Not Null	Pending Query Description
Pend_Query_Date	DateTime	Not Null	Pending Query Date
Pend_reason	Varchar(50)	Allow Null	Pending Reason
QA_ID	Varchar(10)	Foreign Key	Quality Analyst ID

Table 3.14. Pending _Query

Name : Escalation_Query

Description : Used to store Escalation Query details

Fields Name	Type	Constraints	Description
Query_ID	Number(8)	Foreign Key	Query ID
Client_ID	Varchar(10)	Foreign Key	Client ID
Escalation_reason	Varchar(50)	Not Null	Escalation Reason
Escal_Query_Date	DateTime	Not Null	Escalation Query Date

Table 3.15. Escalation_Query

CHAPTER 4

SYSTEM DEVELOPMENT

4.1 INTRODUCTION

System development is a series of operations performed to manipulate data to produce output from computer system. This aim at translating the design of the system produced during the design phase into code in user programming language. A modular approach is used for the development of the software.

The development phase for the project was created from the specification created during the design phase. A principal activity of the development phase is coding and testing the computer program that make up the computer program component of the overall system. Other important activities include implementation, planning, equipment acquisition and system. The development phase concludes with the report and review.

4.2 MODULE DESCRIPTION

The proposed system is designed in such a way so as to meet the requirements. The modules which are used in the system are

- Client
- Frequently Asked Queries
- Quality Analyst
- Customer Support Representative
- Administrator

4.2.1 Client Module

The client is the customer for this software company. While client purchasing software the company will provide an account number. Through this number username and password can create. Using this client can logon to the website. Clients will have to submit their complaints through online. First the client can view the solutions from FAQ list and if the solution is not in the FAQ list then client can post new Complaints. For each new complaints query ID is generated. Client can get the solution back through the website itself.

4.2.2 Frequently Asked Queries Module

This module consists of a list, which is FAQ list. It contains Frequently Asked Queries with solutions. The client can view this list and if the query exists then client can obtain the solution directly. In this Module we will categories the queries based on Software. If the query does not exist the client will send a query to the company.

4.2.3 Quality Analyst Module

This module is to supervise the queries and the solutions. The Quality Analyst is like supervisors. The query from the client will come to the QA. QA will verify the queries. If it is valid it will forward to the CSR otherwise it will be discarded there itself.

QA will obtain the solution from CSR and check the quality of the solution. If it is quality solution then it will send to the Client else it will be resend back to the CSR.

4.2.4 Customer Support Representative Module

CSR module is the place where the actual processing takes place. The CSR will receive the queries from the QA and he will provide solution to the queries and will forward it to QA. Here the CSR has to satisfy three constraints, If CSR is opening a query then CSR should provide the solution within 10 minutes, per day CSR should provide solution to 50 queries. Per month a CSR should provide solution to 1200 queries.

4.2.5 Administrator Module

Authentication process for the CLIENT, CSR, and QA will be provided by administrator. The process and activities of the project are maintained and controlled by the administrator. Report for the whole project likely day report, monthly report, yearly report, CSR report will also be generated by the generator.

CHAPTER 5

SYSTEM TESTING AND IMPLEMENTATION

5.1 INTRODUCTION

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. System testing is the stage of implementation that we aimed at assuring that the system works accurately and efficiently before live operation commences.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. The user tests the developed system and changes are made according to their needs. The testing phases involve the testing of developed system using various kinds of data.

The candidate system may subject to variety of tests. In the response security and usability is tested. A series of testing is performed for the proposed system, before the system is ready for user acceptance testing.

The objectives of testing are

- Testing is the process of executing the program with the intention of finding an error
- A good test is one that has a high probability of finding an as-yet-undiscovered error
- A successful test is that which uncovers as-yet-undiscovered error

5.2 SYSTEM TESTING

For every software project, there is an inherent conflict of interest that occurs as testing begins. The people, who have built the software, are now asked to test the software. These same developers have a vested interest in demonstrating that the program is error free, that it will be completed on schedule and with in budget. From psychological point of view, software analysis and testing (along with coding) is constructed works. The software engineer creates a computer program, is demonstrated and data structure.

5.2.1 White Box Testing

White box focuses on the program control structure. Test cases are derived to ensure that all statements in the program have executed at least once during, testing and that all logical conditions implying that this test is typically applied to small program components. The system has been tested by providing variety of inputs to ensure that all the statements are executed at least once and that too in the expected manner. All topic and transaction path from origin to destination was tested to identify and correct the possible error.

5.2.2 Black Box Testing

Black box testing broadens our focus and might be called testing in the large. Black box tests are designed to validate functional requirements without regards to the internal working of the program. Black box techniques focus on information domain of the software deriving test cases by partitioning input and output in a manner that provides through test coverage. The requirement for higher quality software demands a more systematic approach to testing. The specification stating what a program should do, and it should perform under various conditions are examined. The test cases are developed for each condition or combination of conditions and submitted for processing. By examining the results, the performance of the program according to its specified requirements can be determined.

5.2.3 Unit Testing

Unit testing is the first level of testing in this different module is tested against the specifications produced during the design of the modules. Unit testing is done for the verification of the code produced during the coding phase and to test the internal logic of modules. It refers to the verification of the singe program module in an isolated environment. Unit testing first focused on the modules independently of one another to locate errors. After coding each dialogue is tested and run individually. All unnecessary coding were removed and it was ensured that all the modules works as the programmer would expect. Logical errors found were corrected.

5.2.4 Integration Testing

Data can be lost across an interface, one module can have adverse afford on another, sub functions, when combined, may not produce the desired major functions. Integration testing is a systematic testing for constructing the program structure while at the same time conducting tests to uncover errors associated with in the interface. The objective of it is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the vast expense of the entire program complicate the isolation of courses. In integration testing steps, all the errors un-covered are corrected for the next testing step.

5.2.5 Validation Testing

Validation testing provides the final assurance that the software meets all the functional, behavioral and performance requirements. The software is completely assembled on a package. Validation succeeds when the software functions in a manner in which the user expects. Validation refers to the process of using software in a live environment in order to find errors.

During the courses of validating the system, failures may occur and sometimes the coding has to be changed according to the requirements. Thus the feedback from validation phase generally precedes changes in the software. Once the application was made free of all logical and interface errors, inputting dummy data ensured that the software developed satisfies all the requirements of the user. This dummy data is usually known as test case.

5.2.6 User Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system. Users at time of development can make changes whenever required. This is done regarded to the following points.

- Input Screen design
- Output Screen design
- On-line message to guide the user
- Format of ad-hoc reports and other output

The above testing is done talking various kinds of test data; Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again un-covered and corrected by using above testing steps and corrections are also noted for future use.

5.3 SYSTEM IMPLEMENTATION

Implementation is the stage where theoretical design is converted into a working system. The following are steps involved in the implementation plan.

- Test the system with sample data
- Detection and correction errors
- Make the necessary changes in the system
- Check with the existing system
- Installation of hardware and software utilities
- Training and involvement of user personnel

User Training and Documentation

The implementation of the system includes the training of the system. Training of the system operators includes not only the instruction on how to use the system, but also how to diagnose system errors and mal-functions and ways to resolve the same. So proper training should be provided to system operators. No training is complete without familiarizing users with simple system maintenance activities.

Change Over

Implementation of a new computer system to replace an existing is a more difficult conversion. If not properly planned, there can be many problems. Some large computer systems have taken long time to convert. The accuracy of the conversion is of at most importance both to user confidence in the system and to effective operation. There are many types of change over, such as direct changeover, parallel running, and staged changeover.

CHAPTER 6

CONCLUSION AND SCOPE FOR FUTURE ENHANCEMENT

6.1 CONCLUSION

The project has been successfully developed and enabled to serve the purpose without any difficulty. The project has been designed and implemented and provides a full-fledged approach for proficient with best of results. The project satisfies each efficient user for saving his time and cost. The system is integration easily with other. Even a name user can operate the system, as the user interface is GUI based with a lot of help options.

Currently, the only way for the training department to know about their efforts on the training a program was conducted and feedbacks send towards them by the client. The organization depends on the comments send by the clients.

A client always expects services to be offered as soon as possible. Organization is responsible for making client's to be satisfied. **"NOTATION ANALYSIS AND SCRUTINY SYSTEM"** will prove an added advantage to them in this internet world.

6.2 SCOPE FOR FUTURE ENHANCEMENT

The system has the capability for easy integration with other systems. New modules can be added to the existing system with less effort. Providing extra facilities can enhance the planning module.

APPENDICES

SAMPLE CODING

// new query submit

```
protected void submit_Click1(object sender, EventArgs e)
{
      try
      {
      SqlConnection db = new
      SqlConnection("database=notation; server=.;
      Trusted_Connection=yes");
      SqlCommand cmd = new SqlCommand();
      SqlDataReader dr;
      db.Open();
      cmd.Connection = db;
      cmd.CommandText ="select query_desc from query_master where
      query_desc='"+TextBox4 .Text .ToString().Trim ()+"'";
      dr = cmd.ExecuteReader();
      if (dr.Read())
      {
       check_query_status_not_zero();
      }
     else
      {
       clientcheck();
      }
      db.Close();
      }
```

```
      catch (Exception ex)
      {
      Response.Write(ex.Message.ToString());
      }
}
```

// check query status

```
public void check_query_status_not_zero()
{
      try
      {
SqlConnection db = new
SqlConnection("database=notation;server=.;Trusted_Connection=yes");
      SqlCommand cmd = new SqlCommand();
      SqlDataReader dr;
      db.Open();
      cmd.Connection = db;
      cmd.CommandText = "select status from query_master where
      query_desc='" + TextBox4.Text + "'";
      dr = cmd.ExecuteReader();
      int query_status=int.Parse (dr.GetValue (0).ToString ());
      if (query_status == 0)
      {
      Response.Write("The Query U Entered is already in existing  query
 area with Solution. Please check iT!");
      }
      else
      {
      Session["processing_query_name"] =
      TextBox4.Text.Trim().ToString();
      Response.Redirect("client_query_processing.aspx");
      }
      dr.Close();
      db.Close();
      }
         catch (Exception ex)
         {
             Response.Write(ex.Message.ToString());
     }
 }
```

REFERENCES

BOOKS

- Matthew MacDonald, **"The ASP.NET: THE Complete Reference"**, *TataMacGraw Hill Publications, 2002*
- Mirdula Pariha, Jeff Webb, **"Microsoft ASP.NET"**, *TataMacGrawHill.*
- Paul Nielson, **"Microsoft SQL Server"**, *Tata Mac Graw Hill.*
- Rama Ramachandran, *Bill Evjen Hollis, Rockford* **"Professional VB.NET 2003"**, *Wrox Publications.*
- Roger.S.Pressman, **"Software Engineering"**, *TataMacgrawHill Publications, 2003*

WEBSITES

- http://www.aspexperts.com
- http://www.w3schools.com/aspnet/defaults.asp
- http://www.vbnotnetheaven.com
- http://en.wikipedia.org
- http://www.sql-server-performance .com