CBR-ESiCA

(Case-Based Reasoning – Expert System in Complaint Analysis)

By

Wahidah binti Padeli

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
Business Information Systems

JUNE 2006

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

CBR-ESiCA

(Case-Based Reasoning – Expert System in Complaint Analysis)

By

Wahidah binti Padeli

A project dissertation submitted to the
Business Information Systems (BIS) Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for
BACHELOR OF TECHNOLOGY (Hons)
(BUSINESS INFORMATION SYSTEMS)

Approved by,

____

(Mr. Jale Ahmad)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
June 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by any person.

Wahidah

WAHIDAH BINTI PADELI

ACKNOWLEDGEMENT

Thank Allah for giving me an opportunity to finish up my Final Year Project for this semester. During developing my FYP project, I have learned new knowledge, how to work under pressure and the real meaning of patience. I was able to develop my skill in programming and in managing databases.

I would like to express my deepest appreciation and sincere thank to my supervisor, Mr. Jale Ahmad for his valuable advices, guidance and close collaboration during my FYP. He is the best supervisor I ever met. This project could not have been finished without Mr. Jale who not only served as my supervisor but also encouraged and challenged me throughout my project development.

Not forgotten to my family who always encourage and be on my side through out my life. Special thank to my parent, Mrs. Siti Ashah binti Ali and Mr. Padeli bin Yahaya for their support, sacrifice, advice and encouragement; to Mr. Ang Hock Soon, who helps me a lot in providing information and guiding me in codes the system, I will never forget that. Not forgotten, to someone personal who always encourage, give an advice and be on my side, thanks a lot!

Besides that, I also would like to thank all people who are involved directly or indirectly in my FYP and in giving me ideas and advices. They are in particular below:

- Lecturers
- Classmates and friends

ABSTRACT

Many companies concern most about their level of customer complaints. The numbers of customer complaints are usually used as business performance indicator. A lot of companies handle a survey in order to get response or complaint from their customers and make an analytical study on it. In this research, the author has developed **Case-Based Reasoning Expert System in Complaint Analysis (CBR-ESiCA)** in order to automate all the processes involved in categorizing the customer complaints into particular complaint category. The scope of the research is customer complaints regarding to home care products. CBR-ESiCA categorizes customer complaints into one of eight key complaint categories. New complaints are solved by adapting previously successful solutions to similar complaints. The main objective of the system is to categorize customer complaints into particular complaint categories and generate report based on analytical study made.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND NOMENCLATURES

CBR	:	Case-Based Reasoning
DAS	:	Defect Analysis System
ES	:	Expert System
FL	:	Fuzzy Logic
PC	:	Personal Computer
RAM	:	Random Access Memory

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND OF STUDY

Customer satisfaction is a key driver of profits. One way to increase customer satisfaction is by reducing the number of customer's complaints from time to time. Nowadays, many companies require their customers to fill in the complaint form, submit the form to particular person, and then that person will identifies the complaints and categorizes them into particular category before takes any further action. Usually, complaint category with highest number of complaints will be given more attention. Automation of those processes may reduce much time besides reducing human-error. Thus, *CBR-ES in Complaint Analysis (CBR-ESiCA)* is developed to automate the processes. The target users for the system are Customer Care Officer and Customer Care Manager (trained users).

1.2 PROBLEM STATEMENT

1.2.1 Problem Identification

Study done on Bbraun Medical Industries Sdn. Bhd., Bayan Lepas, Penang shown that its Customer Care Officer is responsible to key in all customer complaints into a system called, DAS (Defect Analysis System). The officer will review the complaints and categorizes them into particular complaint category before takes any further actions

[Ang]. The current process consumes much time because the officer need to key-in the complaints, and identify the complaint category each time the complaint is received even though the complaint is similar to the previous. As a result, the author has developed *CBR-ESiCA* to automate all those processes.

1.2.2 Significance of the Project

1. Paperless transaction
    - Customers submit the complaints through company's website. Thus, they do not need to fill in the form anymore. All the complaints are stored in table in database called dbTextCompare.

2. Simplifies Customer Care Officer's task
    - The officer does not to need enter the customer complaints into the system because the system captures the complaints from the database.

3. Automatically categorizes the complaints
    - The system categorizes the complaints into particular problem category only with single click.

4. Automatically generates the report
    - The system generates the report based on data stored in database. The report provides information on total complaints for each complaint category. This information can assist Customer Care Manager in order to identify which complaint category needs more priorities.

1.3 OBJECTIVES

1. To capture customer complaints directly from the database

- The system will captures customer complaints from the database. Customer Care Officer does not need to enter the complaints each time he/she receives the complaints.

2. To categorize customer complaints into particular category

- The system compares new complaints to the previous complaints stored in the database. The system will suggest the category of the new complaints based on the comparison made.

3. To provide solutions for new customer complaints based on the previous solutions

- After the system categorizes the complaints, the system will suggest a few solutions based on previous similar complaints.

4. To generate report for the Customer Care Manager

- The system generates the report to summarize the result of the analysis. The report consists of total number of complaints for each complaint category. This report will allow Customer Care Manager to identify which complaint category contributes most complaints.

1.4 SCOPE OF STUDY

CBR-ESiCA focuses more on Case-Based Reasoning approach whereby previous similar solutions will be used to solve new complaints. This system is categorized as Expert System because it imitates human expert in order to solve the new complaints.

Even though the original idea of the system came from the system used by Bbraun Medical Industries, the author choose home care products business as scope of study because medical field involves many technical terms. Some of terms are difficult to be understood. Besides that, the expert from Bbraun Medical Industries was not able to provide the list of their product complaints because of confidential matters.

In this system, the scope of customer complaints is limited only to eight complaint categories regarding to home care products business in Malaysia (Refer to Table 1.0). The products include Vitamin, Shampoo, Skincare, Hand Moisturizer and Water Supply Products. However, the system can be modified in order to be applied in other businesses areas. The 'virtual company' (company that doest not exist in the real world) is used in this research, named AsFird Medicare Sdn. Bhd.

Complaint Category	Description/Action taken	Example of complaint
1. overdoses	- Give an advice to customer. Ask them to consume based on prescriptions given	- Always vomits within two hours of taking the pills. - Itchy, dry, and peeling skin - Irresistible desire to sleep - Experiencing headache after 3 hours using the product - My skin become flushing. The redness of my skin is very obvious after using the product
2. sensitivity	- Product must be gentle enough for daily use on even the most sensitive skin. Its gentle formula should help the skin retain moisture, and give a smoother, healthier and younger-looking appearance.	- Skin effects were evaluated by visual grading (redness, dryness and smoothness) - Skin become red, dry and not smooth - Temporary irritation (itching) or redness occur - various skin problems (such as color changes on the skin, and hard patches on the palms
3. extreme	- Reduce the contents of chemicals that may damage human hair. Ask customer to use according to prescriptions	- I am experiencing some degree of significant hair loss - I am 15 years old and my hair has gone through a lot. My hair is naturally thick, but lately it has thinned down a lot. No visible bald spot of course, but I'm still concerned
4. unclear	- To provide safe drinking water, the various components of the water supply system—from protection at the source to treatment and distribution of drinking water to consumers—must be understood and managed as a whole	- Drinking water has objectionable tastes and odors - Consuming your drinking water over a long period results in various health effects including skin problems and high blood pressure. - Experiencing diarrhea due to infection that last a few days

5. moisturizer	- Add a moisturizer that holding moisture in the skin and eliminating fine lines - Product should be enriched with 4% antioxidant Vitamin E (tocopheryl acetate) and chamomile extract to help deep moisturize, soften and give user's skin a firmer, smoother appearance.	- Does not protect the skin against damage caused by elements such as heat, sun, wind, cold and pollution - does not soothes and heals skin irritations and sunburn
6. reflection	- This has been called the discontinuation or "rebound" effect. It can be completely avoided by a continuous consumption of vitamin or supplementary pills at regular basis.	- People may experience the onset of a cold, fatigue, or other temporary unpleasant feelings. - Not prominent spots but they are there and one can see circular areas (the size of a coin) where the skin has become a little lighter.
7. labeling	- As well as medicinal ingredients, labels will list non-medicinal ingredients. Labels also will warn consumers of potential adverse effects and will provide contact information so consumers can report problems with the product	- The label does not provides enough information on how much to use to treat a problem - No expiry date on the product label - The label does not provides medical advices - Blur text on the label
8. expired	- Expiry date indicates the date after which medicines are considered no longer safe for consumption or ingestion. Some medication may become toxic after a certain date, or lose their effectiveness in the case of medication. This feature on Labels, therefore, serves to protect consumers' interests, to ensure that they get products of high quality and products that are safe.	- The medicine loses its effectiveness. - When I first got the product, it tastes great but now it tastes differently.

Table 1.0: Complaint's Category

CHAPTER 2
LITERATURE REVIEW

2.1 WHAT IS AN EXPERT SYSTEM?

The category of expert system is a very successful approximate solution to the classic AI problem of programming intelligence. Professor Edward Feigenbaum of Stanford University, an early pioneer of expert systems technology, has defined an expert system as "an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solutions." [Feigenbaum, 1982]. That is, an expert system is a computer system that emulates the decision-making ability of a human expert. The term *emulates* means that the expert system is intended to act in all respects like a human expert. Emulation is much stronger than a simulation, which is only required to act like real thing in some respects.

Expert system is a branch of AI that makes extensive use of specialized knowledge to solve problems at the level of a human expert. An expert is a person who has expertise in a certain category. That is, the expert has knowledge or special skills that are not known or available to most people. An expert can solve problems that most people cannot solve or can solve them much more efficiently (but not as cheaply).

The knowledge in expert systems may be either expertise or knowledge that is generally available from books, magazines, and knowledgeable persons. The term *expert systems*, *knowledge-based system*, or *knowledge-based expert system* are often used synonymously [Giarratano,1994].

The *knowledge base* of expert systems contains both factual and heuristic knowledge. Factual knowledge is widely shared knowledge, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable in the particular field. Heuristic knowledge is the less rigorous, more experiential, more judgmental knowledge of performance. In contrast to factual knowledge, heuristic knowledge is rarely discussed, and is largely individualistic. It is the knowledge of good practice, good judgment, and plausible reasoning in the field. Knowledge in the knowledge base will be manipulated and used by inference engines to form a line of reasoning [S.Engelmore,1993].

2.2 HOW AN EXPERT SYSTEM WORKS?

Figure 2.0 illustrates the basic concept of an expert system. The user supplies facts or other information to the expert system and receives expert advice or expertise in response. Internally, the expert system consists of two main components, knowledge base and inference engine. The knowledge base contains the knowledge with which the inference engine draws conclusions. These conclusions are the expert system's responses to the user's queries for expertise.

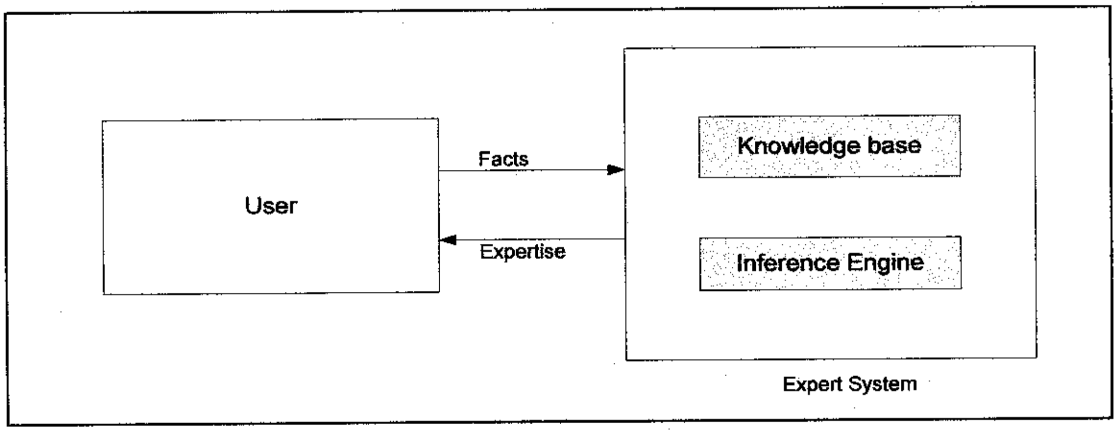


Figure 2.0: Basic Concept of an Expert System Function

2.3 EXPERT SYSTEM DEVELOPMENT LIFE CYCLE

Figure 2.1 shows the linear model of expert system development life cycle. The objective of each stage of the life cycle is shown in Table 2.0.

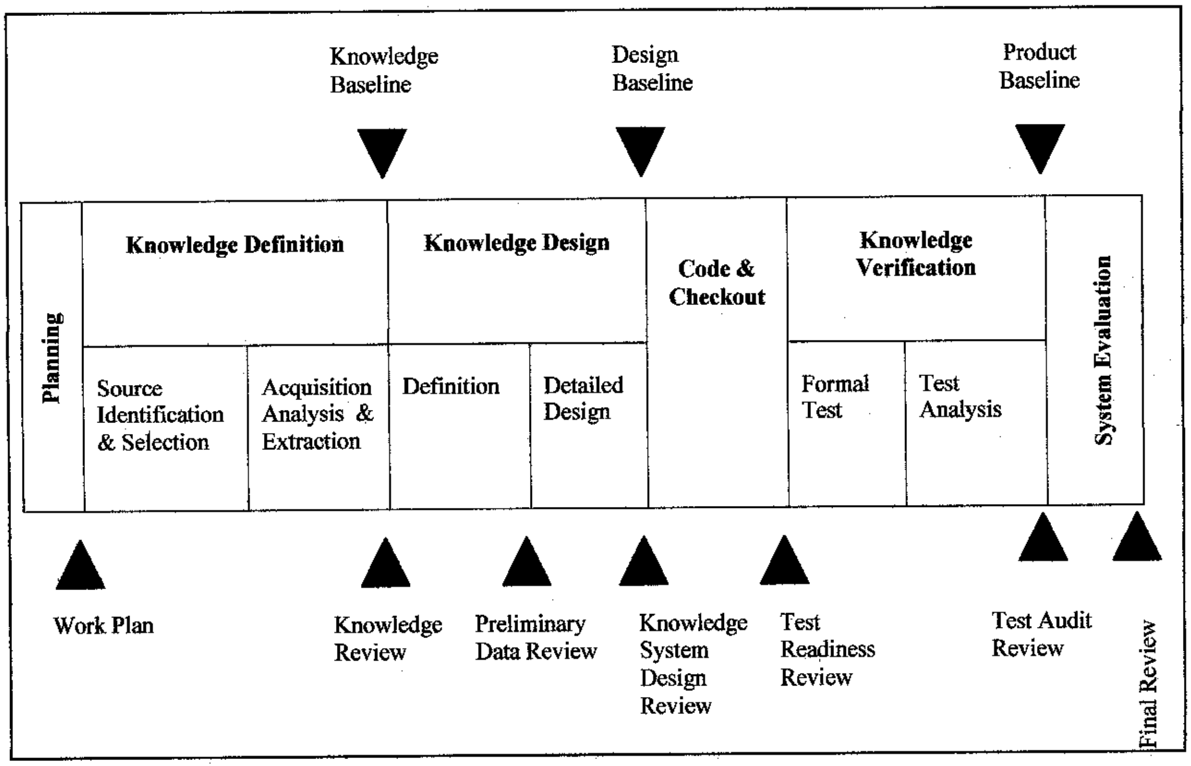


Figure 2.1: The Linear Model of Expert System Development Life Cycle

Stage	Objective
1. Planning	- To produce a formal work plan for the expert system development Note: The work plan is a set of documents that will be used to guide and evaluate development
2. Knowledge Definition	- To define the knowledge requirements of the expert system
3. Knowledge Design	- To produce the detailed design for an expert system
4. Code and Checkout	- Begins the actual code implementation
5. Knowledge Verification	- To determine the correctness, completeness, and consistency of the system
6. System Evaluation	- To summarize what has been learned with recommendations for improvements and corrections

Table 2.0: Objective of each stage in Expert System Development Life Cycle

2.4 EXPERT SYSTEM DEVELOPMENT TOOLS

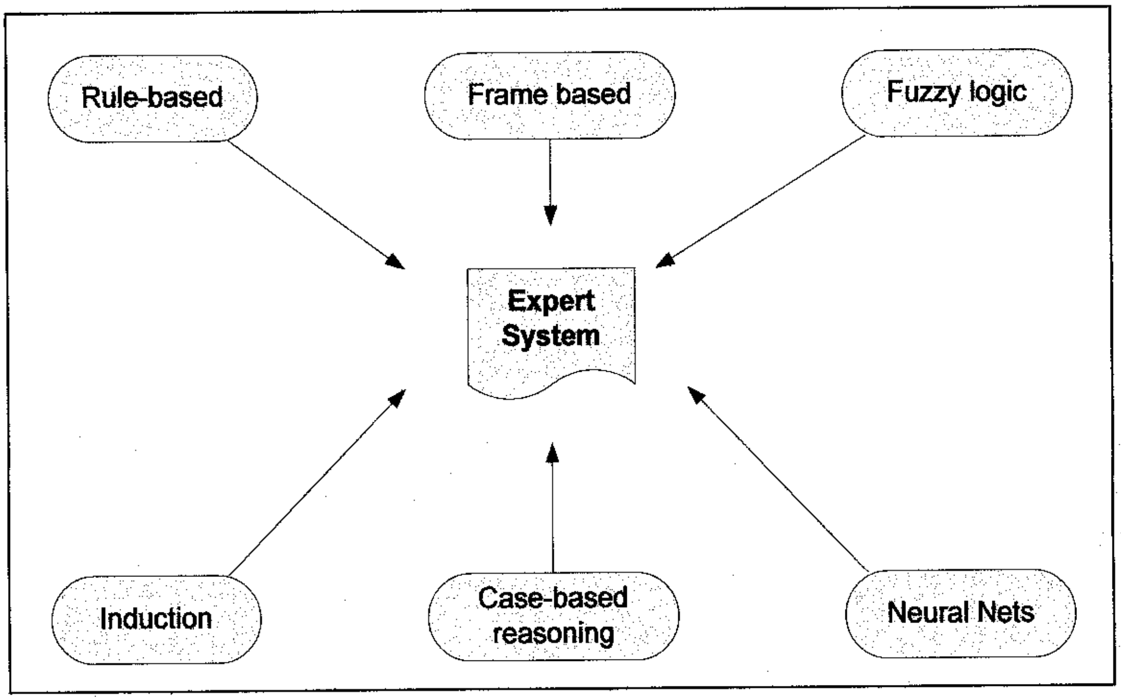


Figure 2.2: Expert System Development Tools

An expert system tool, or shell, is a software development environment containing the basic components of expert systems. Associated with a shell is a prescribed method for building applications by configuring and instantiating these components. Figure 2.2 illustrates tools available in developing Expert System.

1. Rule-based
   - Rule-based systems are a relatively simple model that can be adapted to any number of problems. Rule-based systems are really only feasible for problems for which any and all knowledge in the problem area can be written in the form of if-then rules and for which this problem area is not large. If there are too many rules, the system can become difficult to maintain and can suffer a performance hit.

2. Frame based
   - For architecture to be frame-based means that it is organized around the representation of all of its knowledge in frames. This is important because it allows a uniform representation of knowledge. In a frame-based system, all knowledge is represented as slots of some concept, which describe properties of that concept.

3. Fuzzy Logic
   - Fuzzy Logic (FL) is a problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems. It can be implemented in hardware, software, or a combination of both. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information [Steven D. Kaehler]

4. Induction
   - Inductive reasoning results in the addition of semantic information. The reasoning starts with cases and ends with general conclusion. In induction, assertions do not necessarily lead to true conclusions. Induction is more difficult than deduction because of both the addition of new semantic information and because the inferred concept may not be the correct one.

5. Case-Based Reasoning
   - Case-Based Reasoning solves new problems by adapting previously successful solutions to similar problems (Discussed in Section 2.5).

6. Neural Nets

   - Neural Nets is very sophisticated modeling techniques capable of modeling extremely complex functions. Neural Networks are analytic techniques modeled after the processes of learning in the cognitive system and the neurological functions of the brain. It is capable to new observations from other observations.

2.5 WHAT IS CASE-BASED REASONING (CBR)?

CBR is one of an expert system's development tools. It is a recent approach to problem solving. In CBR terminology, *case* usually denotes a *problem situation.* Thus, basically: CBR solves new problems by adapting previously successful solutions to similar problems [Watson]. Illustrations below illustrate the application of Case-Based Reasoning in Drilling (Illustration 1) and Medical (Illustration 2) area:

<u>Illustration 1:</u>
A drilling engineer, who has experienced two dramatic blows out situations, is quickly reminded of one of these situations when the combination of critical measurements matches those of a blow out case. In particular, he may get a reminding to a mistake he made during a previous blow-out, and use this to avoid repeating the error again [A.Aamodt,1994].

<u>Illustration 2:</u>
A doctor examined a particular patient – gets a reminding to a patient that he treated two weeks ago. Because of both patients have similarity of important symptoms; the doctor uses the diagnosis and treatment of previous patient to determine the disease and treatment for the current patient [A.Aamodt,1994].

2.6 HOW THE CBR WORKS?

CBR is a technology that uses past experiences (cases) to solve current problems. Much in same way as our older doctor treats a patient, a CBR system takes information on some current problem and looks for the most similar past case. If a reasonable match is found, the system suggests the same solution used in the past. When a search fails to locate a similar case, the search itself becomes the basis for a new case. In effect, a CBR system can learn from new experiences. A CBR system derives its power from its ability to retrieve relevant cases quickly and accurately from its case library, and its ability to learn [T.Leondes]. The basic operation of CBR system follows a four-step process (as shown in Figure 2.3) :

1. Input information about a current problem
2. Find a similar case
3. Adapt this case to fit the current problem
4. Provide a suggested solution

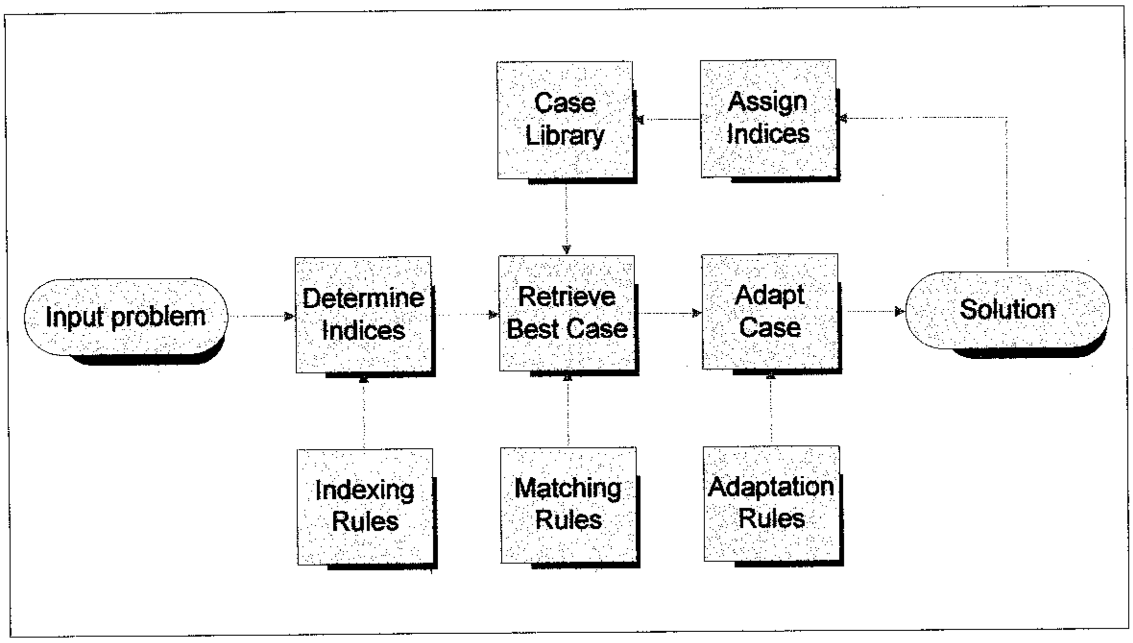


Figure 2.3: CBR Flow-chart

2.7 CBR CYCLE

All CBR methods have to deal with: identifying the current problem situation, finding a past case similar to the new one, using that case to suggest a solution to the current problem, evaluating the proposed solution, and updating the system by learning from this experience. These processes are known as CBR cycle. CBR Cycle consists of four processes: Retrieve, Reuse, Revise and Retain [Althoff,1989]:

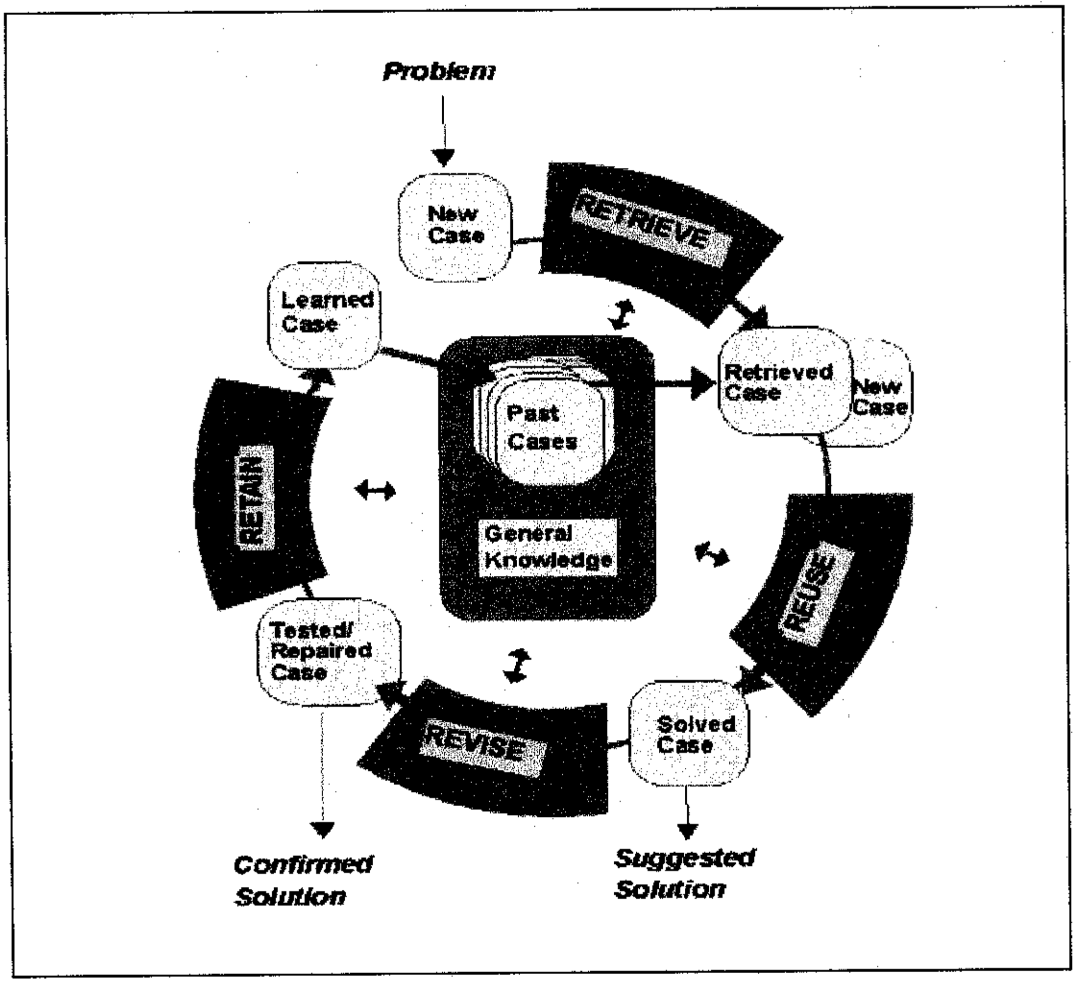


Figure 2.4: The CBR Cycle

Figure 2.4 illustrates CBR Cycle which consists of four processes:

1. RETRIEVE

- The system determines, usually using set of rules, the features of the current problem that are relevant to finding similar cases. System retrieves a set of potentially relevant cases from the case library.

2. REUSE

- The system then matches the current problem with the most similar ones. The similarity of cases is based on how well they match on the relevant features. The system will reuse the solutions of similar cases and apply the same solutions to the new problems.

3. REVISE

- The system adapts the most similar case to fit the current problem, creating a new case that is then added to the library. This is accomplished by the first determining what is different between input and retrieved case and then modifying the retrieved case by taking into account the differences. In some instances, no adaptation is needed because what was done in the past is perfectly fine for the current problem.

4. RETAIN

- Once the solution has been evaluated, and if necessary repaired, a decision is made as to what information from the problem solving should be retained in order to assist problem-solving in the future.

2.8 EXAMPLES OF CBR SYSTEM

1. CASCADE
   - An example of a successful CBR help-desk application
   - A system developed at DEC to assist in recovering from VMS device driver failure [Simoudis,1991]. Information provided by a caller about a current problem is entered into an on-line form by a help-desk technician and the CASCADE retrieves relevant cases that suggest solution. The system provides quick and effective solutions, increases customer satisfaction, and avoids the need to pass along problems to design engineers

2. HYPO
   - Another category that provides fertile ground for CBR technology is the legal profession, which thrives on judicial precedents established in landmark cases The similarities, as well as the differences, between the current and the landmark case are considered to guide legal decision-making
   - HYPO supports legal reasoning in category of patent law [Rissland,1986]. Provided a case description involving a given patient claim violation, the system uses a library of precedent cases and looks for those that are most similar to the given case that were decided in favor of the defense. HYPO then generates plausible argument for the prosecution or defense. It can take the plaintiffs or defendant's side in a dispute and is equally good at creating argument for each

3. JUDGE
   - Assists a judge during the sentencing phase of trail
   - JUDGE models a judge who is responsible for establishing a sentence for an individual convicted of a crime [Bain,1986]. The system uses as input the charge, events that occurred and legal statuses concerning related crimes. The system's case library contains previous crimes and the sentences imposed for each. After locating the most similar past case, JUDGE adapts the case to fit the current crime by making the sentence longer or shorter

CHAPTER 3
METHODOLOGY

In *CBR-ESiCA* system development, the author used Linear Model of Expert System Development Life Cycle (as discussed in Chapter 2) in order to manage the project processes. This model consists of six stages: Planning, Knowledge Definition, Knowledge Design, Code and Checkout, Knowledge Verification, and System Evaluation.

3.1 CBR-ESiCA SYSTEM DEVELOPMENT LIFE CYCLE

STAGE 1: PLANNING

During this stage, the author prepared Project Title Proposal and got an approval from respective supervisor. After the supervisor approved the proposal, the author continued developing the project by doing some researches and getting an overview of DAS from expert in Bbraun Medical Industries Sdn. Bhd. The author also named the system as *CBR-ESiCA* (Case-Based Reasoning-Expert System in Complaint Analysis).

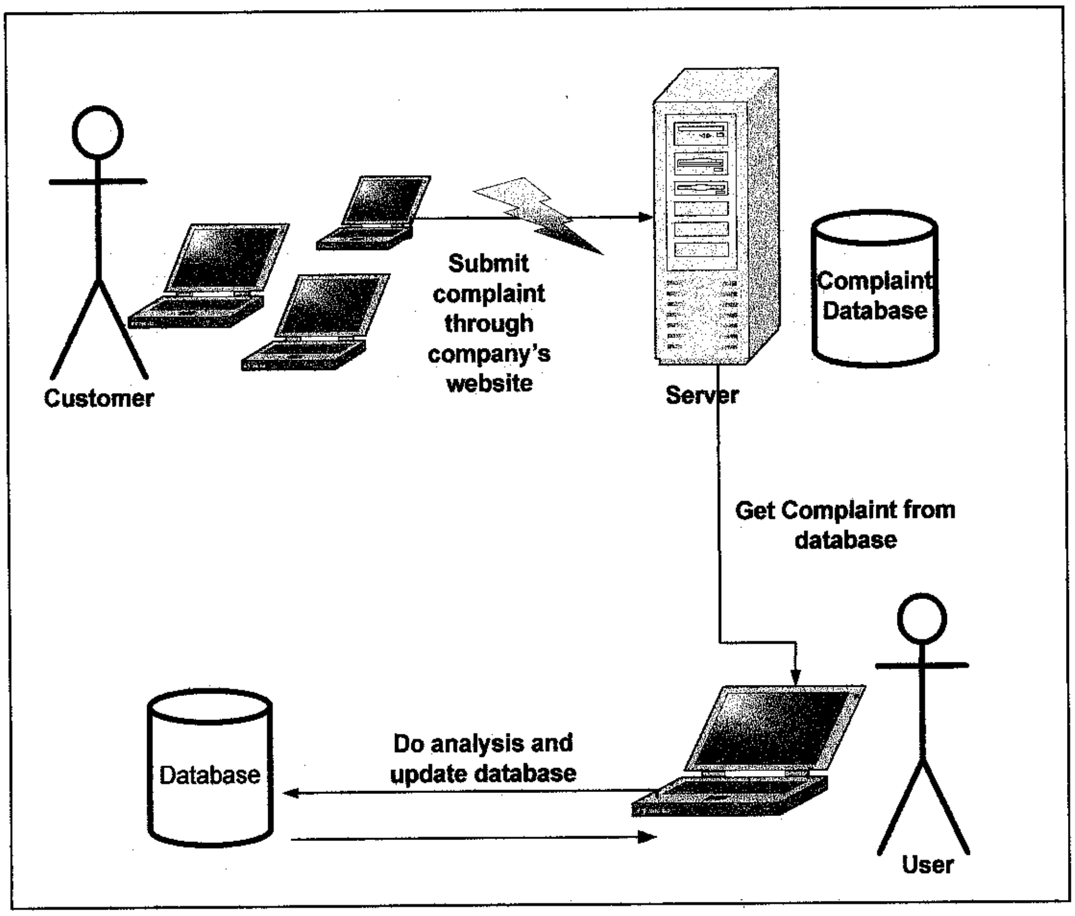
STAGE 2: KNOWLEDGE DEFINITION

The author started to collect knowledge requirements of the system. The author did some researches on customer complaints regarding to Home Care products. The information regarding to the complaints were gained from the expert, Internet, journals, and books. Then, the author documented all the collected data.

STAGE 3: KNOWLEDGE DESIGN

In this stage, the author produced detailed design for the system. The author developed system architecture (Figure 3.0), process flow diagram (Figure 3.1), and data flow diagram (Figure 3.2) of the system to ease implementation stage later.

In Figure 3.0 below, it shows that customer will submit complaints through company's website. The complaints will be stored in dbTextCompare database. The *CBR-ESiCA* will capture all these complaints after the user login into the system.



Customer
Submit complaint through company's website
Server
Complaint Database
Get Complaint from database
Database
Do analysis and update database
User

Figure 3.0: System Architecture of *CBR-ESiCA*

Figure 3.1 illustrates the flow of the system. The user needs to log in into the system before proceeds with analyzing the complaints. After the access is granted, the user is allowed to analyze the complaint. The system will displays all relevant categories if similar cases or complaints found in the database. If no similar complaints found in the database, the user has two options: either to categorize the new complaint into new category or to categorize it under existing category. After the analysis is completed, user may view the report and display the description of the particular complaint category.

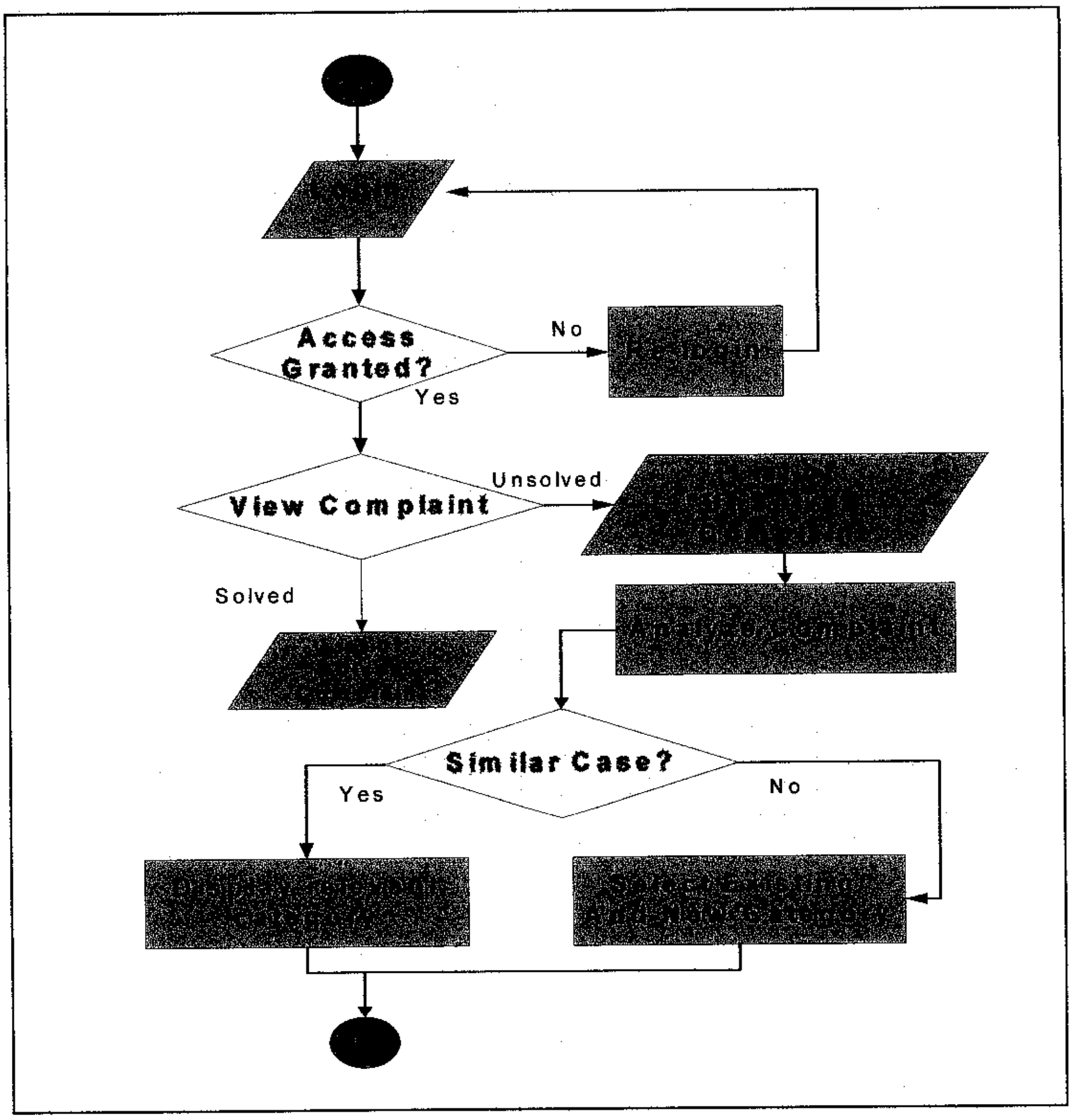


Figure 3.1: ***CBR-ESiCA*** **flow chart**

Figure 3.2 illustrates the flow of data for *CBR-ESiCA*. Data flow begins with storing customer complaints into tblComplaint table. During Analysis Process, the data from this database table will be captured and compared with existing cases in tblCases table. If the system found any similar complaints, it will trigger the category of the complaints from tblCategory table and display the result.

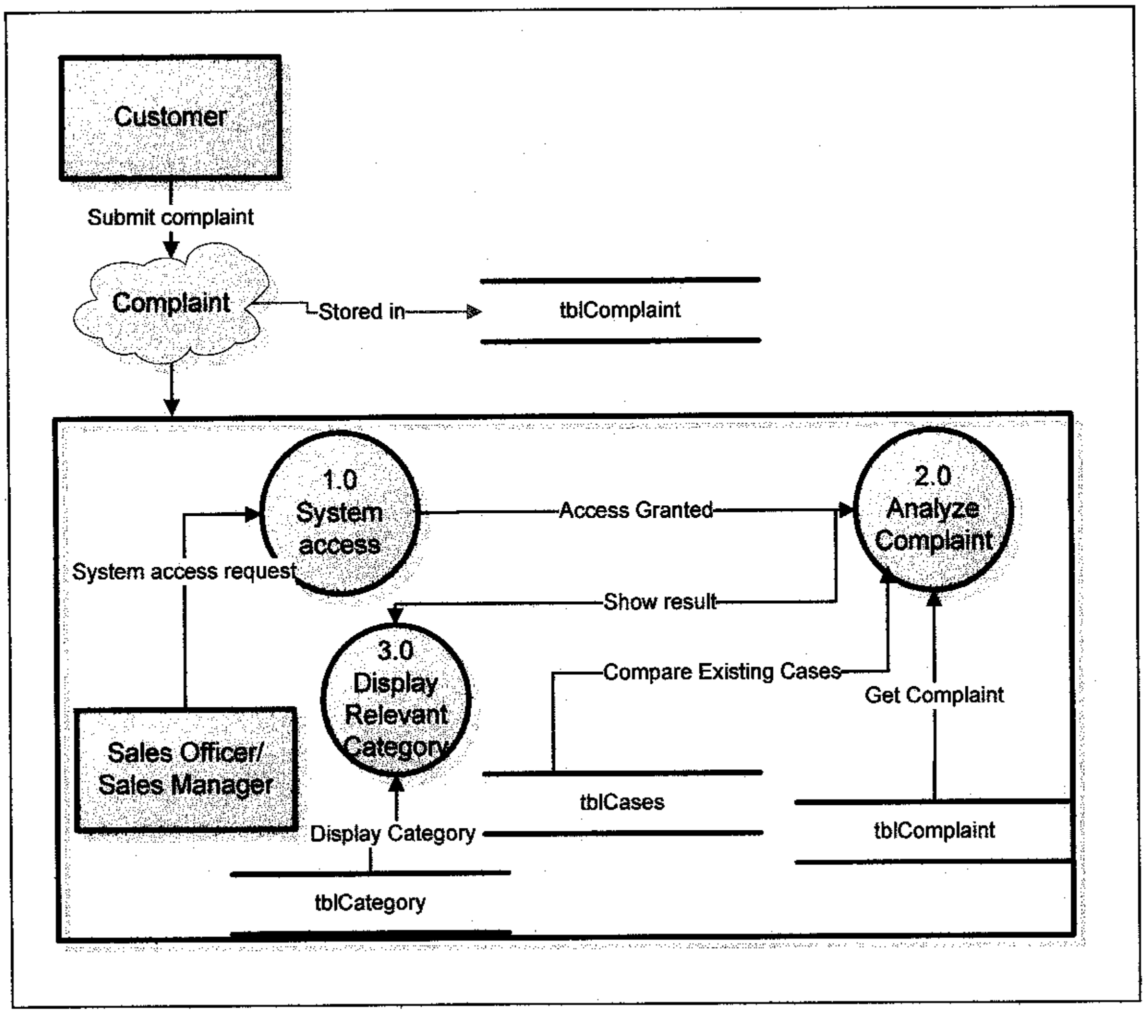


Figure 3.2: Data Flow Diagram for *CBR-ESiCA*

STAGE 4: CODE AND CHECKOUT

This stage is similar to Implementation stage in other System Development Life Cycle. The author started to code the system using Visual Basic 6.0. Besides the flow of the process, the User Interface also needs to be considered. The author concentrated on the user-friendliness and the attractiveness of the system.

During the system development, the author tested the codes unit by unit. For example, after completed Login function, the author tested the codes to ensure Login function works well. Then, the author will continue coded the other functions and tested those functions. After all the functions have been completed, the author tested the system as a whole. The last testing approach is called as Integrated Unit Testing.

Processes in CBR-ESiCA:

Process 1: User login

Users need to login into the system before they can access it. "View Report" function only accessible if the user login as 'Customer Care Manager'.

Process 2: Complaint Analysis

The system captures the customer complaints from tblComplaint. User need to set value for benchmark before start analysis process. If not, the system will display all the complaints even though the percentage of similarities is 0%. After that, click "Analyze" button to do the analysis.

Analysis process:

1. The system splits the complaint and all cases in database according to space " " into the array.

```
trimmedcomplaint = Split(tmpComplaint, " ")
trimmedcases = Split(Caseslist(num), " ")
```

Example:
Complaint – "Any idea why your water source become yellow?"

Word array for complaint:

```
trimmedcomplaint(0) = "Any"
trimmedcomplaint(1) = "idea"
trimmedcomplaint(2) = "why"
trimmedcomplaint(3) = "your"
trimmedcomplaint(4) = "water"
trimmedcomplaint(5) = "source"
trimmedcomplaint(6) = "become"
trimmedcomplaint(7) = "yellow?"
```

2. Check whether there are a symbol and “noise words” in the Complaint – remove them if any.

Replace symbols with “”

```
'array for symbols
symbol(0) = "."
symbol(1) = ","
symbol(2) = "-"
symbol(3) = "!"
symbol(4) = "?"
symbol(5) = "_"
symbol(6) = "("
symbol(7) = ")"
symbol(8) = ":"
symbol(9) = ";"
symbol(10) = ""
symbol(11) = """

For i = 0 To UBound(symbol)
tmpComplaint =
Trim(Replace(tmpComplaint, symbol(i),
""))
Next i
```

Remove “noise words”

```
'list of noise word

nix(0) = "a"
nix(1) = "an"
nix(2) = "the"
nix(3) = "of"
nix(4) = "in"
nix(5) = "for"
nix(6) = "with"
nix(7) = "to"
nix(8) = "from"
nix(9) = "is"
nix(10) = "was"
nix(11) = "are"
nix(12) = "were"
nix(13) = "and"
nix(14) = "that"
nix(15) = "which"
nix(16) = "or"
nix(17) = "by"
nix(18) = "i"
nix(19) = "my"
nix(20) = "have"
nix(21) = "has"
nix(22) = "been"
nix(23) = "am"
nix(24) = "on"
nix(25) = "but"
nix(26) = "and"
nix(27) = "because"
```

3. Then the system will compare whether there is a shared words (same words) between

- simpan() - array that store complaint after symbols and "noise words" are removed
- simpan2() - array that store cases after symbols and "noise words" are removed

*Note: technique used in comparison process is "Text Analysis-shared words". System will calculate total number of same words between sentences.

```
'find shared words between sentences
 For f = 1 To Count1
   For fc = 1 To Count2
     Compare = InStr(1, simpan2(fc), simpan(f), vbTextCompare)
     If Compare <> 0 Then
       samewords = samewords + 1
       Exit For
     Else
     End If
   Next fc
 Next f
```

4. After calculate the number of same words, the system will show the percentage of similarities for each cases using the following formula:

```
percentage = Format$(((samewords / ((count1 + count2) / 2)) * 100), "fixed")

*NOTE: count1 – total word of trimmedcomplaint array
       count2 – total word of trimmedcases array
```

<u>Process 3:</u> Display result

After analysis process completed, the system will display chart that show the number of relevant cases for each category based on percentage set earlier (user need to click "Analysis Result" tab). All relevant cases that similar with the complaint will be shown when user click "Relevant Cases" tab.

*Note: If the complaint does not match to any cases in database, either the user can choose to "Select Existing" (categorize the complaint under existing category) or to "Add New" (add new category into the database).

<u>Process 4:</u> View Report

Customer Care Manager may view the report that consists of all customer complaints. From the report, Customer Care Manager will be able to analyze which category contributes most complaints. The corrective action will be taken based on that report.

STAGE 5: KNOWLEDGE VERIFICATION

The author got an advice and help from expert from Bbraun Medical Industries Sdn. Bhd to ensure the correctness, completeness, and consistency of the system. The author gone through the codes with the expert and did manual calculation to ensure that the calculation made by the system is correct. The error-tolerance value is less than 5% (illustrated in Table 3.0).

EXAMPLE	
Formula = Format$(((samewords / ((count1 + count2) / 2)) * 100), "fixed")	
Complaint	= "Why my skin become red after using your product??!"
Cases in database	= "My skin become flushing. The redness of my skin is very obvious after using the product"
Similar words after removing *"noise words" (samewords)	= skin, become, red, using, product = 5
Total words in complaint after removing *"noise words" (count1)	= why, skin, become, red, after, using, product = 7
Total words in cases after removing *"noise words" (count2)	= skin, become, flushing, redness, very, obvious, after, using, product =9
CBR-ESiCA result	= 66.67% similarities
Manual Calculation	= (5/((7+9))/2) *100 = (5/8) * 100 = 62.50% similarities
Error tolerance value	= 66.67% – 62.50% = 4.17%

Table 3.0: Example of Calculation

*NOTE: "Noise words" – "I", "my", "a", "an", "the", "of", "in", "to", "from", "is", "was", "which", "your", etc.

STAGE 6: SYSTEM EVALUATION

After completed Stage 5, the author evaluated the system in order to summarize what has been learned with recommendations for improvements. The authors will discuss on the recommendations for the system in Chapter 5 later on.

3.2 DESIGN TOOLS

Table 3.1 describes the hardware and software used in *CBR-ESiCA* development.

Hardware/Software	**Description**
PC Microprocessor	Pentium 4 1.5 MHz
Minimum RAM	10.0 MB
Visual Basic	VB 6.0
Crystal Report	Version 10
Microsoft Access	2003
Microsoft Visio	2003

Table 3.1: Hardware/Software Description

3.2.1 Codes and Interface Design

Visual Basic 6.0 is used in coding and designing GUI (Graphical User Interfaces) for *CBR-ESiCA*. Visual Basic evolves from ***BASIC*** (an acronym for **B**eginner's **A**ll-Purpose **S**ymbolic **I**nstruction **C**ode), which was created in the 1960's. Its syntax is similar to FORTRAN (**For**mula **Tran**slation). VB is an event-driven rather than procedure-oriented language. In event-driven programs, activity was carried out only after some "prerequisite" actions have been taken or data are ready.

3.2.2 Database design

The author developed the database for *CBR-ESiCA* by using Microsoft Access 2003. Currently, this system consists of three tables:

1. tblComplaint - Stores customer complaints

   tblComplaint (Complaint: char, Status: number)

2. tblCategory – Stores all complaint categories with their ID and Description. This table links to tblCases through categories ID.

   tblCategory (ID: number, Category: char, Description: char)

3. tblCases - Stores all previous complaints. Links to tblCategory through categories
   ID.

   tblCases (Cat_ID: number, Cases: char)

3.2.3 Report Design

The author used Crystal Reports v10 in designing the report for this system. Crystal Reports v10, the release of Seagate Software's widely used and highly acclaimed report writer, provides a sophisticated graphical report designer that can be used with many databases. Crystal Reports v10 is compatible with prior Crystal Reports releases. It comes with several new features including server-side processing, which can reduce client-processing requirements; an ActiveX report designer that allows reports to be designed within Visual Basic (VB); and the ability to directly enter SQL expressions and execute stored procedures for higher-performance reporting.

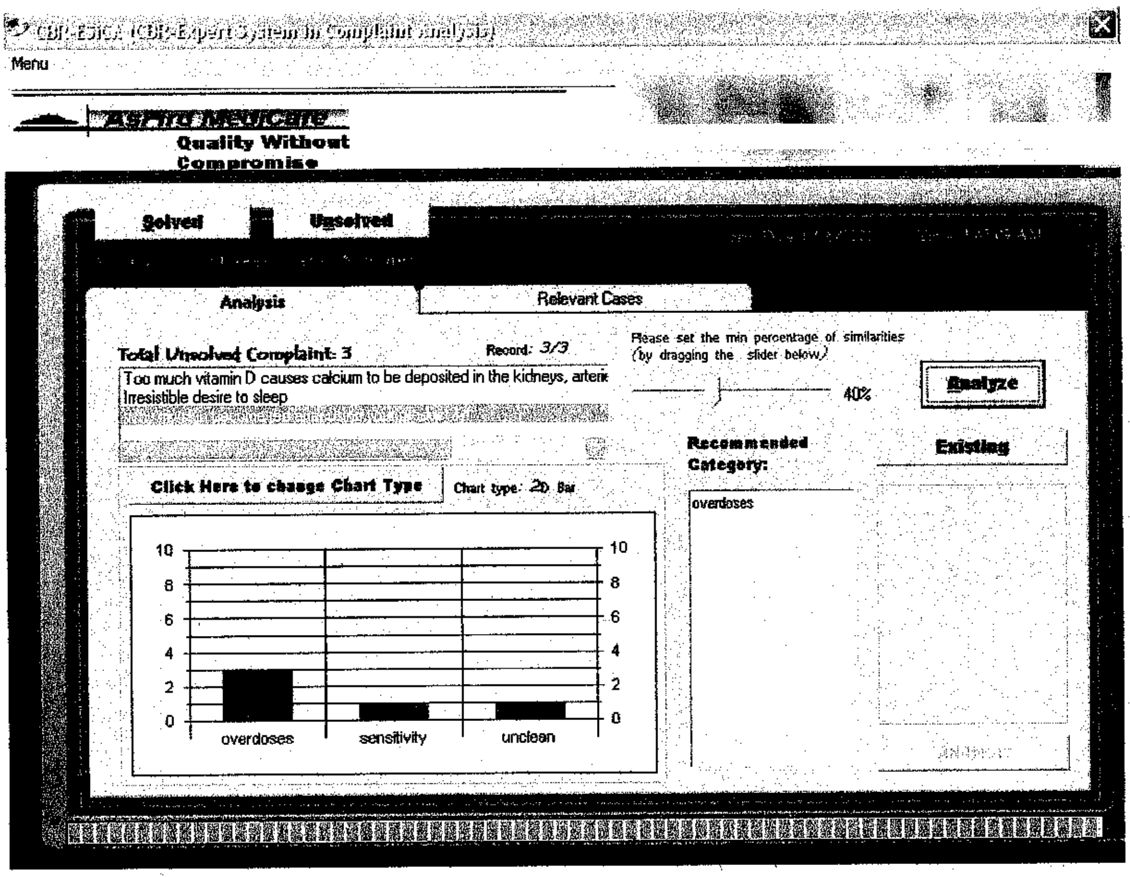
CHAPTER 4
RESULTS AND DISCUSSION

4.1 EXAMPLE OF QUERIES RESULT

Query 1: "why my skin becomes very red after using your product??!"

Benchmark value: 40%

Result: overdoses


CBR-ESICA (CBR-Expert System in Complaint Analysis)

Total Unsolved Complaint: 3   Record: 3/3

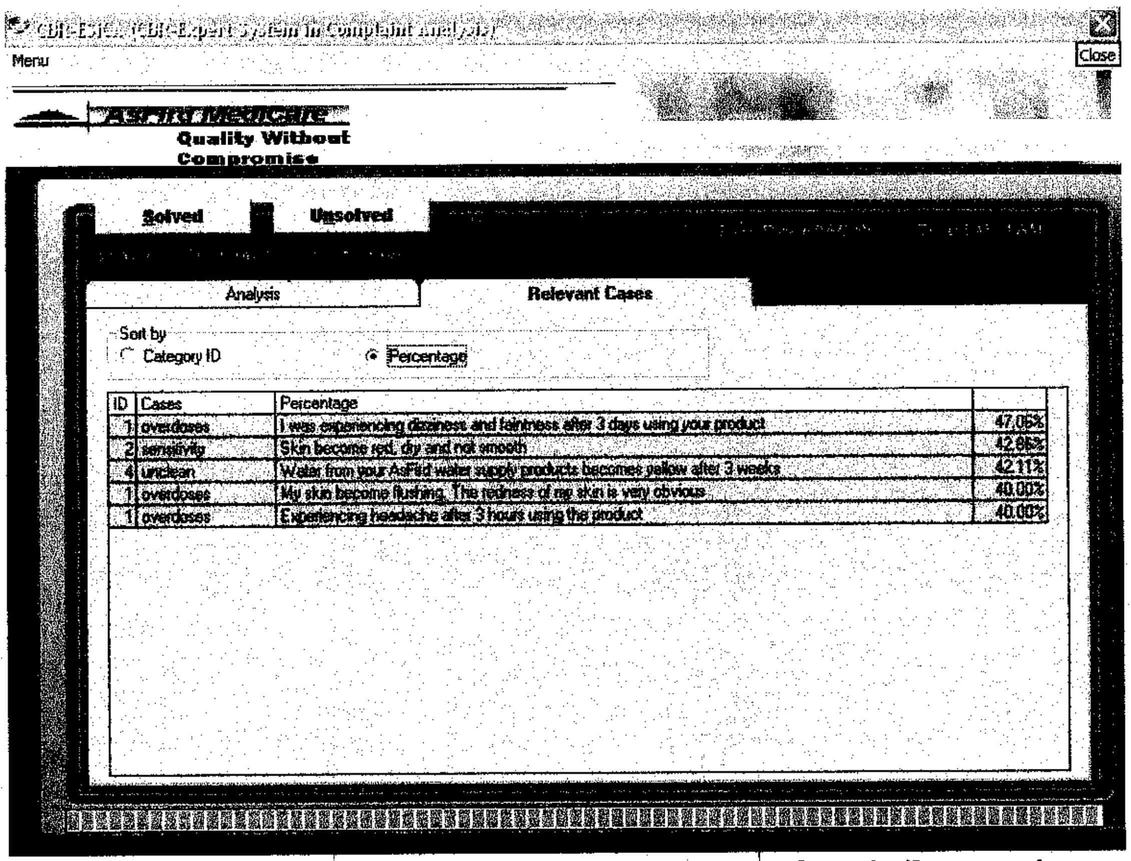
Too much vitamin D causes calcium to be deposited in the kidneys, arterie
Irresistible desire to sleep

Please set the min percentage of similarities (by dragging the slider below) 40%

Chart type: 2D Bar

Recommended Category: overdoses

Figure 4.0 (i): Example of Complaint Analysis – match to similar cases in database.


CBR-ESICA (CBR-Expert System in Complaint Analysis)
Menu
Close
AsFird Medicare
Quality Without Compromise
Solved
Unsolved
Analysis
Relevant Cases

Sort by
Category ID [✓ Percentage]

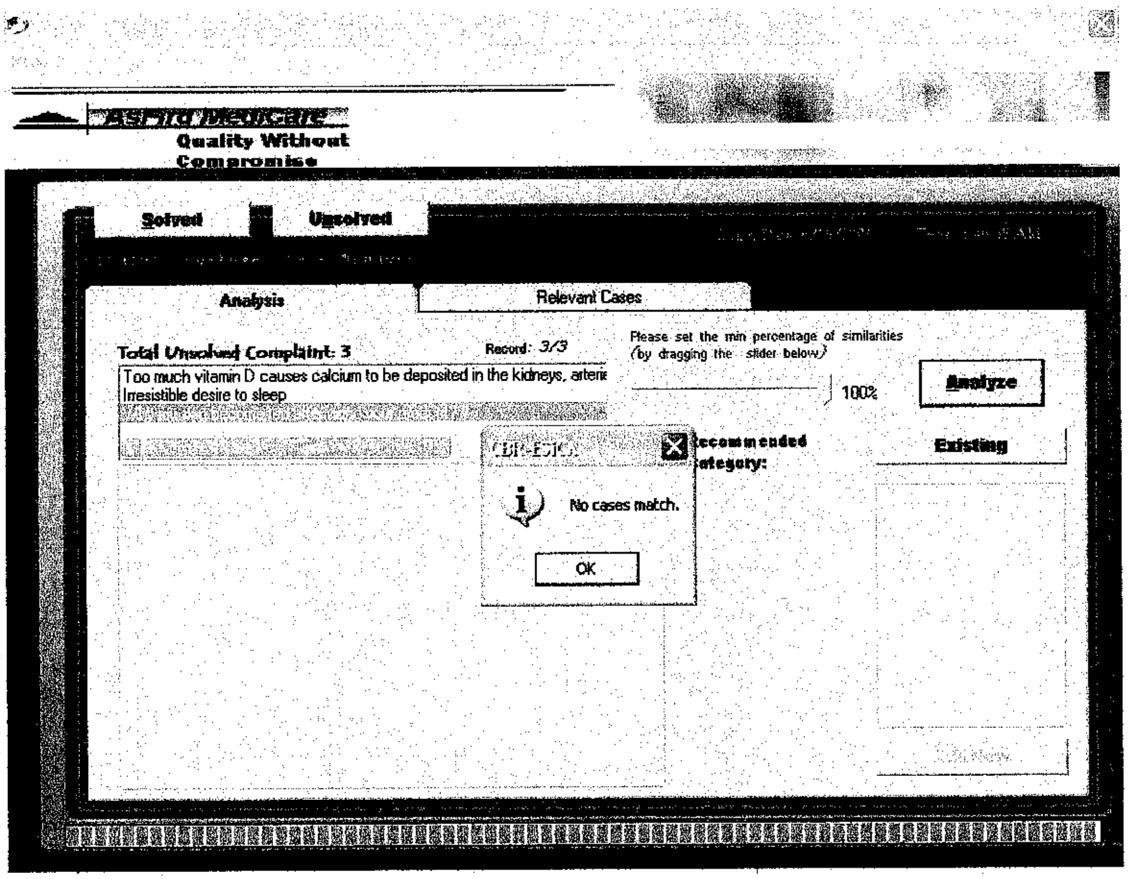
ID	Cases	Percentage	
1	overdoses	I was experiencing dizziness and faintness after 3 days using your product	47.06%
2	sensitivity	Skin become red, dry and not smooth	42.86%
4	unclean	Water from your AsFird water supply products becomes yellow after 3 weeks	42.11%
1	overdoses	My skin become flushing. The redness of my skin is very obvious	40.00%
1	overdoses	Experiencing headache after 3 hours using the product	40.00%

Figure 4.0 (ii): Example of Complaint Analysis – match to similar cases in database.

Query 2: "why my skin become very red after using your product??!".

Benchmark value: 100%

Result: No cases match.



Total Unsolved Complaint: 3 Record: 3/3

Too much vitamin D causes calcium to be deposited in the kidneys, arteri
Irresistible desire to sleep

Please set the min percentage of similarities (by dragging the slider below)

100%

No cases match.

Figure 4.1 Example of Complaint Analysis – no match.

4.2 DISCUSSION

Based on the queries made, the number of relevant cases returned by the system depends on the benchmark value set by the user. Low benchmark value will give more results but the accuracy is lower whilst high benchmark value will return less results. The user can rely on the results given because the error-tolerance value is only 5%. With this small value, the results given by the system is assumed as same as results given by the experts.

After the completion of this research, all the objectives as stated in Chapter 1 are accomplished. The system is able to capture customer complaints directly from the database, then categorize the complaints into particular category, and provide the solutions or descriptions for new complaints based on the previous successful solutions. The system is also able to do an analytical study on the data from the database and create the report for Customer Care Manager.

4.2.1 Challenges and Limitation of the System

The biggest challenge when dealing with expert system is, the difficulties to elicit *tacit knowledge* from expert and explicitly documented it. It is hard to verbalize the knowledge because it is expressed through action-based skills and cannot be reduced to rules and recipes. Thus, in order to develop this system, the author referred to only one expert, Mr. Ang Hock Soon, an expert from Bbraun Medical Industries Sdn Bhd.

The complaints used for *CBR-ESiCA* database are not the real ones. The author failed to get real complaints from Bbraun Medical Industries because of confidential matter. As a result, the company and the complaints used in *CBR-ESiCA* are 'virtual' ones. In addition, the system cannot ensure 100% accuracy. However, the percentage of similarities is reliable and can be proved by manual mathematical calculations.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSION

CBR-ESiCA is an expert system that was proposed in order to categorize new customer complaint based on the previous complaints. The knowledge (tacit knowledge) of previous complaints is elicited from experts. The system is developed to assist Customer Care Officer or Customer Care Manager in categorizing customer complaints into particular complaints category. The system also creates the report to assist Customer Care Manager to analyze which complaint category should be given more attention. In addition, *CBR-ESiCA* can be applied to many business areas. It is not limited or designed just for one particular party.

5.2 RECOMMENDATIONS

1. Increase the system accuracy
   - The system should be able to ensure 100% accuracy in analyzing the complaints.

2. Customer's language and typography should not be a problem for the system.
   - System should have some methods that will enable it to response to the complaint regardless how customer defines the problem.

3. Ability to analyze all complaint at one time

   - Currently the user needs to select the complaint, and click the button to solve the complaint and repeat the same things for next complaints. For future enhancement, the system should be able to solve all the complaints in the database at 1-click.

4. Automatically set the percentage of similarities

   - Default percentage of similarities should be 100%. If there is no cases match in the database, the system should be able display the recommended complaint category that has percentage of similarities near to 100%.

REFERENCES

Journals, Articles, Books:

A.Aamodt, E.Plaza, 1994. *AICom – "Artificial Intelligence Communication"*, IOS Press, Vol. 7:1, pp. 39-59.

Althoff, K.D, 1989. *"Knowledge acquisition in the domain of CNC machine centers"*; Third European Workshop on Knowledge-Based Systems, Paris, July 1989. pp 180-195.

Bain, W.M, 1986. *Case-based reasoning: "A computer model of subjective assessment"*. Ph.D. Thesis, Yale University.

Cornelius T.Leondes, "*The Technology of Knowledge Management and Decision Making for the 21st Century*", Volume 1, Academic Press, A Horcourt Science and Technology Company.

Edward A. Feigenbaum, 1982. "*Knowledge Engineering in the 1980s,*" Dept. of Computer Science, Stanford University, Stanford, CA.

Garry Robinson's. *"Microsoft Chart – A How to Guide For Handling The VB 6 Charting"*. <http://www.vb123.com/toolshed/99_vbchart/>

Ian Watson and Farhi Marir, *"Case-Based Reasoning: A Review"*. <http://www.ai-cbr.org/classroom/cbr-review.html>

Jack Lynch. "*Text Analysis with Compare*", Free Software Foundation Inc, Boston, USA.

James Chapman, 2005. "*Software Development Methodology*", Washington DC. <http://www.hyperthot.com/pm_sdm.htm>

James Crowley. "*Scrolling Text*", Developer Fusion, the UK developer community <http://www.developerfusion.co.uk/>

Joseph Giarratano and Gary Riley, 1994. *Expert Systems:* "*Principles and Programming*, 3rd Edition, PWS Publishing Company.

Kiran. "*Creating Graph using MS Chart Control with an array*", Visual Basic Source Code. <http:/kiranreddys.com/>

Paul Harmon and David King, 1985. "*Expert Systems*", John Wiley and Sons, Inc., p. 21.

Rissland, E.L. and Ashley, 1986. "*Hypothetical as heuristic device*". In Proceedings of AAAI-89, pp. 289-297. Morgan Kaufmann, San Mateo, CA.

Robert S.Engelmore and Edward Feigenbaum, 1993. *Chapter 1:* "*Expert System and Artificial Intelligence*".

Rod Stephens, 1997-2003. "*Sort the columns in an MSFlexGrid control*", Mountain Computer Consulting, Inc. <http://www.vb-helper.com/index.html>

Simoudis, 1991. "*E.Knowledge acquisition in validated retrieval*". Int. J. Expert Syst. 4(3), 299-315.

Steven D. Kaehler, *"Fuzzy Logic – An Introduction (Part 1)"*, Newsletter of the Seattle Robotics Society.
<http://www.seattlerobotics.org/encoder/mar98/fuz/fl_part1.html>

Thomas Connolly, Carolyn Begg, 2002. *Database Systems; "A Practical approach to Design, Implementation, and Management"*, 3rd Edition, Addison Wesley.

Personal Interview:

Ang Hock Soon, DAS Developer and Database Administrator, Bbraun Medical Industries Sdn. Bhd., Penang, , 24th September 2005.

Internet links or business web site:

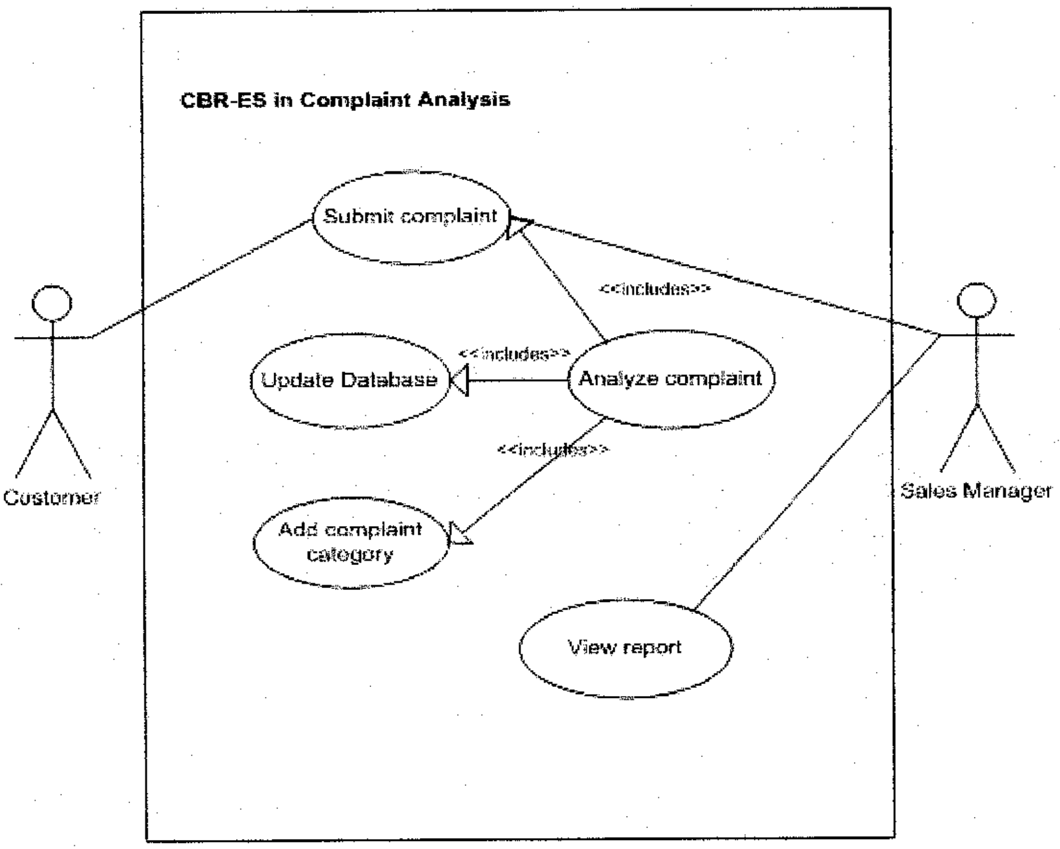
<http://www.bbraunap.com/>

APPENDICES

CBR-ESiCA system requirement:

Minimum RAM	3.0 MB
Visual Basic	VB 6.0
Crystal Report	Version 10
Microsoft Access	2003

Use Case for *CBR-ESiCA*



SYSTEM MANUAL

Customer side:

<u>Step 1:</u>

- Customer submits complaint through company's website.

Companies Side:

<u>Step 1:</u>

- Customer Care Officer or Customer Care Manager need to login into the system.

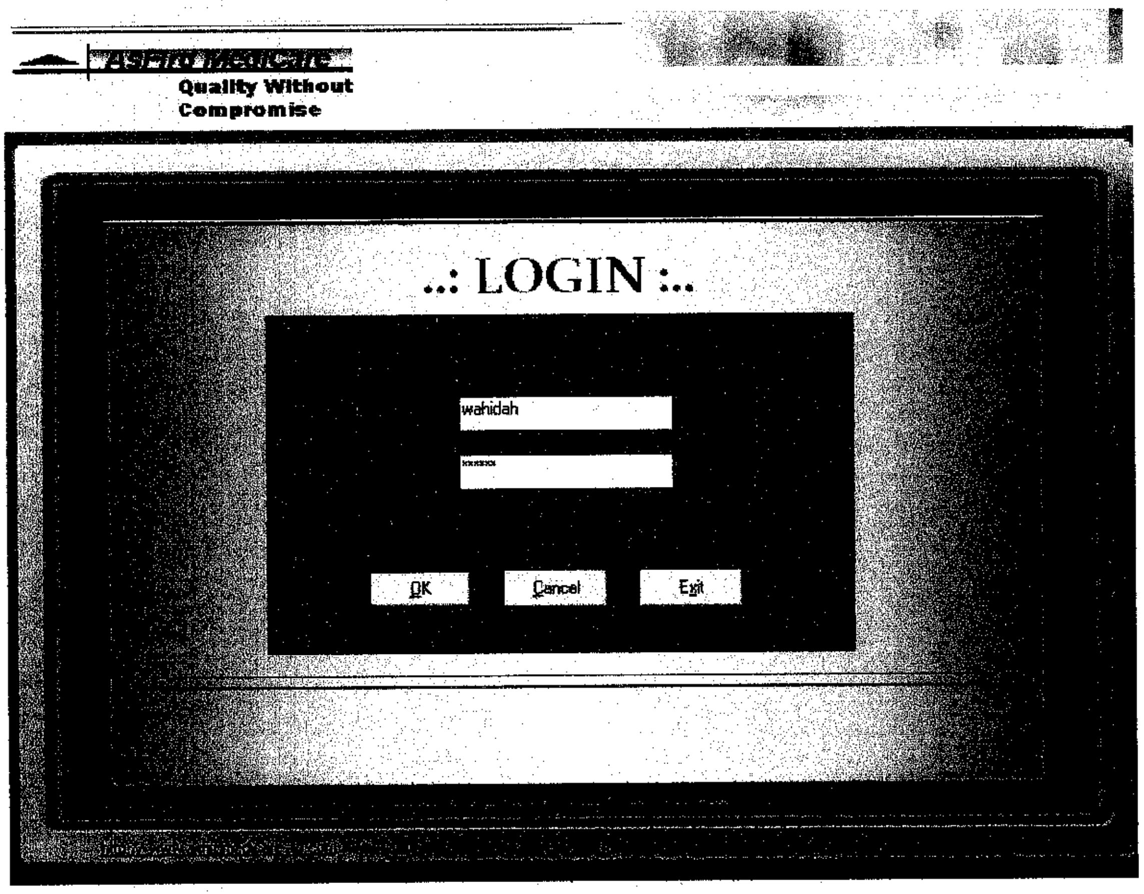
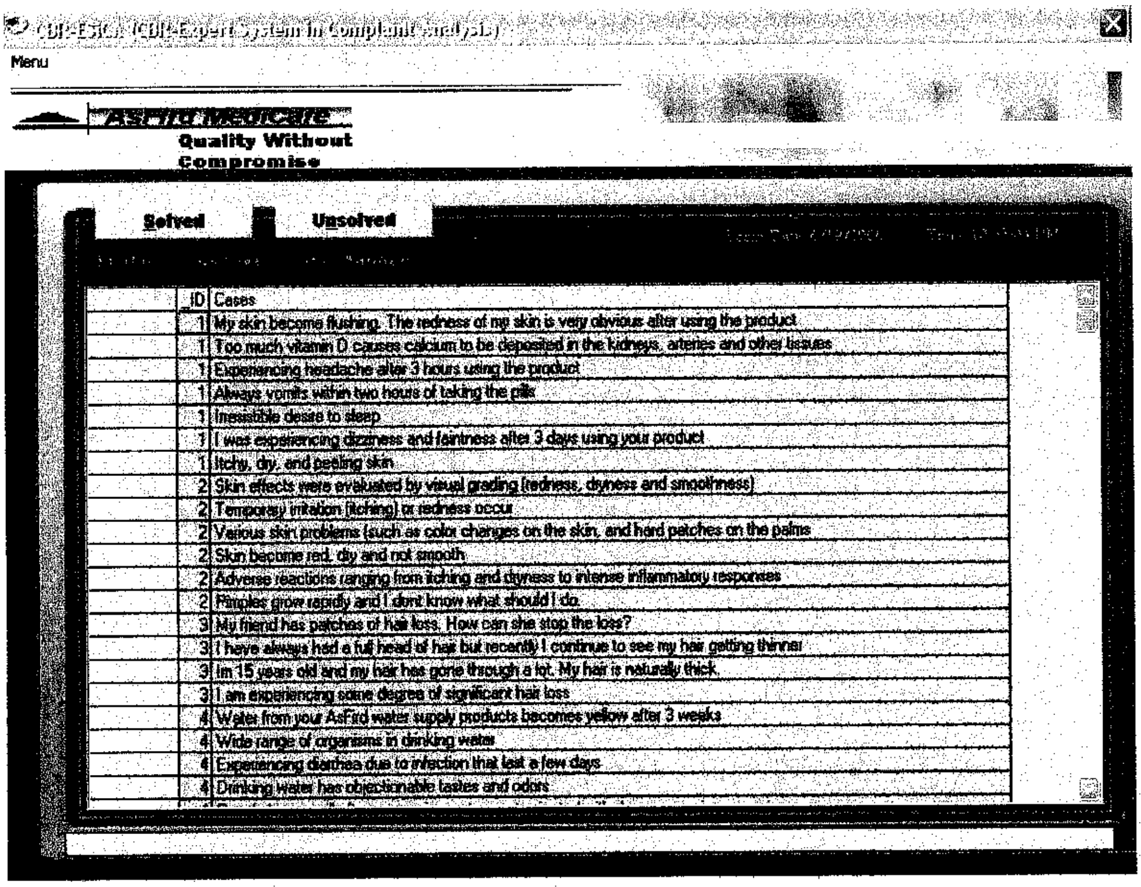


Figure 1.0: Login page

Step 2:

- Display and Analyze Complaint

1. To display solved complaint, click "Solved" button.

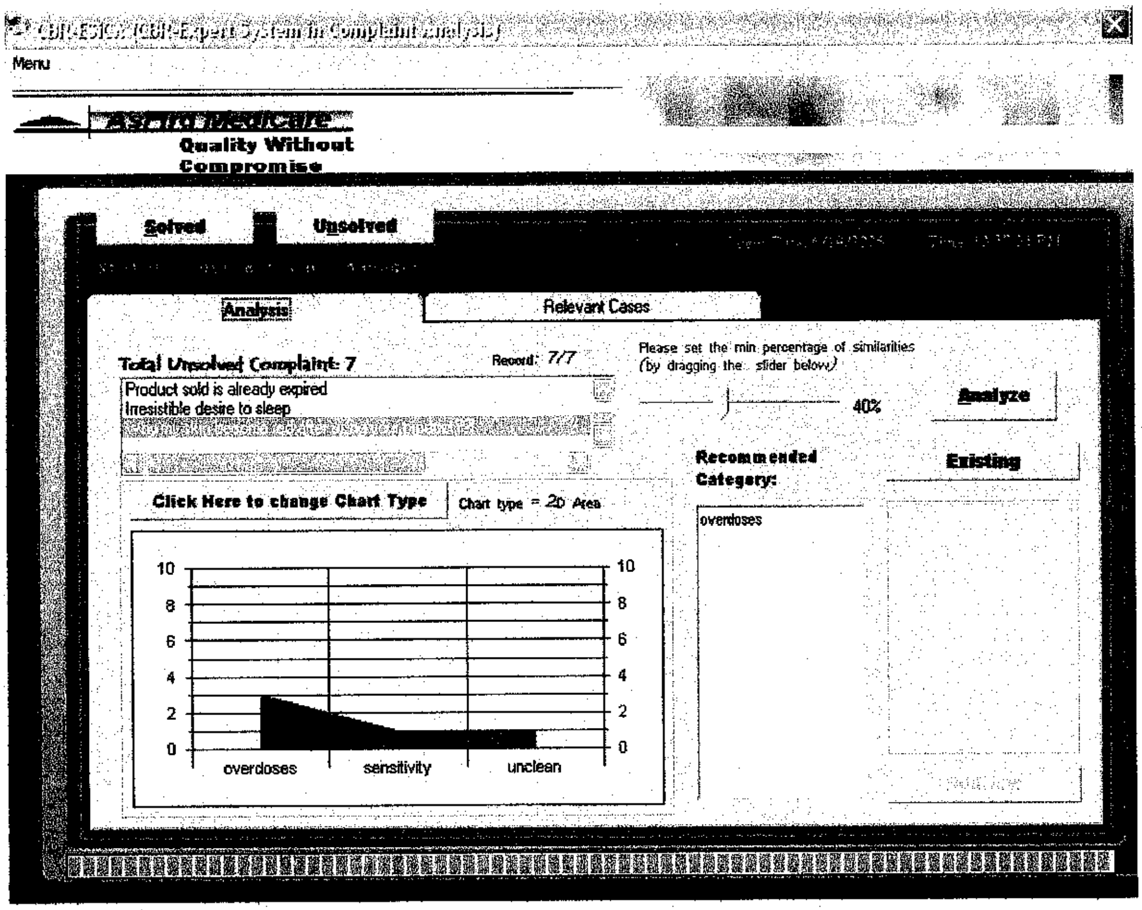


ID	Cases
1	My skin become flushing. The redness of my skin is very obvious after using the product
1	Too much vitamin D causes calcium to be deposited in the kidneys, arteries and other tissues
1	Experiencing headache after 3 hours using the product
1	Always vomits within two hours of taking the pills
1	Irresistible desire to sleep
1	I was experiencing dizziness and faintness after 3 days using your product
1	Itchy, dry, and peeling skin
2	Skin effects were evaluated by visual grading (redness, dryness and smoothness)
2	Temporary irritation (itching) or redness occur
2	Various skin problems (such as color changes on the skin, and hard patches on the palms
2	Skin become red, dry and not smooth
2	Adverse reactions ranging from itching and dryness to intense inflammatory responses
2	Pimples grow rapidly and I dont know what should I do.
3	My friend has patches of hair loss. How can she stop the loss?
3	I have always had a full head of hair but recently I continue to see my hair getting thinner
3	Im 15 years old and my hair has gone through a lot. My hair is naturally thick.
3	I am experiencing some degree of significant hair loss
4	Water from your AsFird water supply products becomes yellow after 3 weeks
4	Wide range of organisms in drinking water
4	Experiencing diarrhea due to infection that last a few days
4	Drinking water has objectionable tastes and odors

Figure 1.1: Display "Solved" complaint

2. To analyze the complaint, click "Unsolved" button, set the benchmark percentage and click "Analyze" button.

3. Click on "Analysis" tab to display number of relevant cases per category through 2D bar chart. User can change chart format to 2D Area (Figure 1.2).



Total Unsolved Complaint: 7

Record: 7/7

Product sold is already expired

Irresistible desire to sleep

Chart type = 2D Area

Please set the min percentage of similarities (by dragging the slider below)

40%

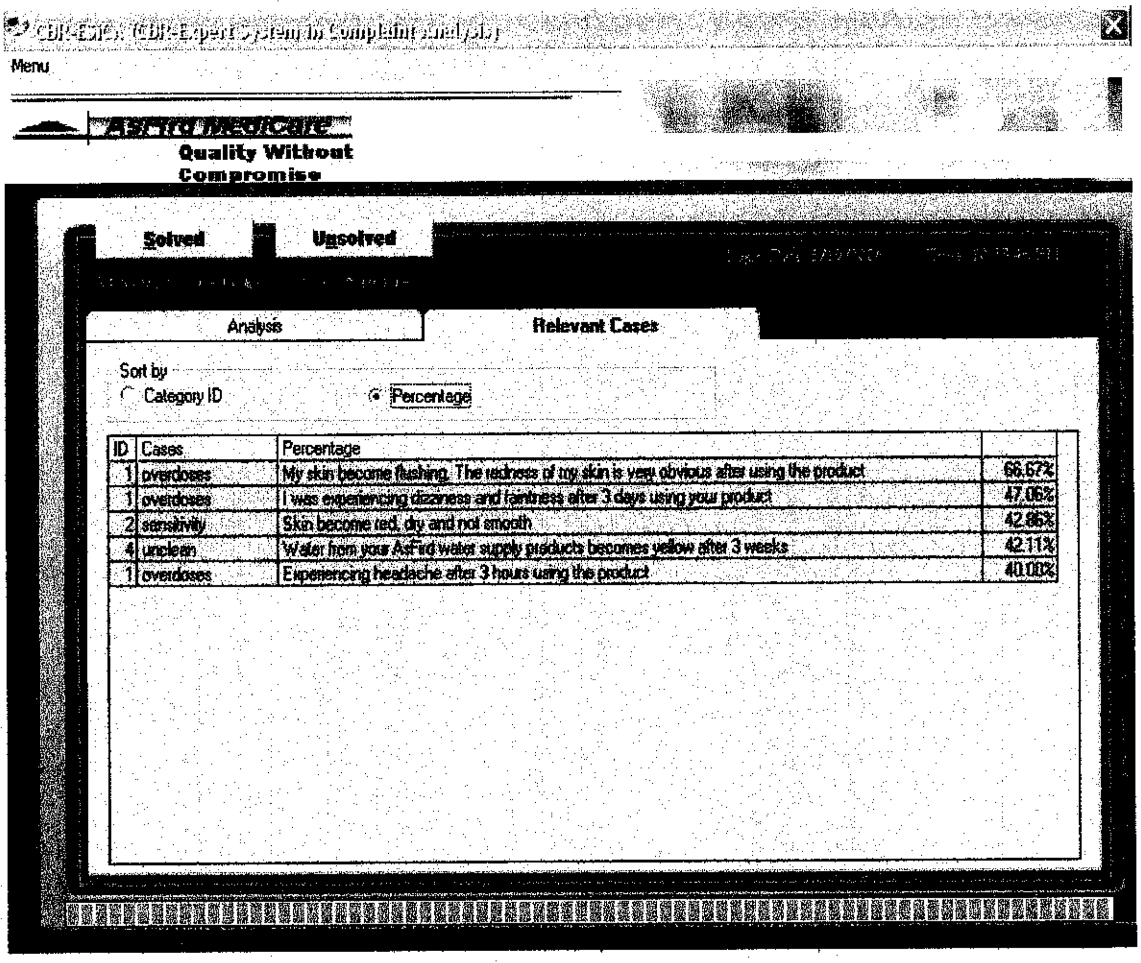
Recommended Category:

overdoses

Figure 1.2: "Analysis" tab – display chart that depicts number of cases per category.

*NOTE: Category that has most relevant cases will be shown as "Recommended" category.

4. Click on “Relevant Cases” tab to display Category ID, Relevant cases and percentage of similarities. User can sort result “by Category ID” or “by Percentage” (Refer Figure 1.3).



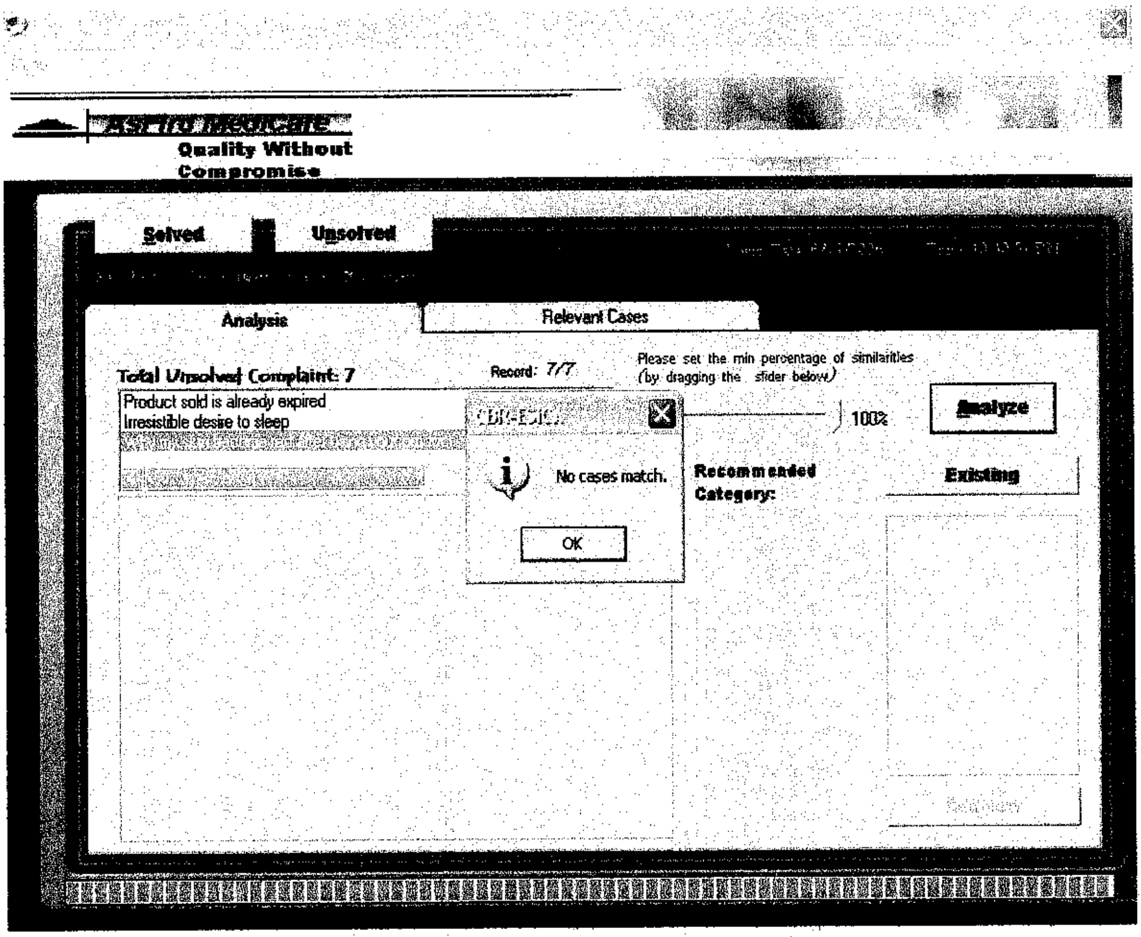
Sort by
Category ID [✓ Percentage]

ID	Cases	Percentage	
1	overdoses	My skin become flushing. The redness of my skin is very obvious after using the product	66.67%
1	overdoses	I was experiencing dizzness and faintness after 3 days using your product	47.06%
2	sensitivity	Skin become red, dry and not smooth	42.86%
4	unclean	Water from your AsFird water supply products becomes yellow after 3 weeks	42.11%
1	overdoses	Experiencing headache after 3 hours using the product	40.00%

Figure 1.3: “Relevant Cases” tab – list out all the relevant cases

*NOTE: only cases that have percentage of similarities greater or equal to benchmark value will be displayed.

5. If the complaint does not match to any cases in the database, the user may choose whether to categorize the case under existing category or to add new category (Refer Figure 1.4, Figure 1.5, and Figure 1.6).



Total Unsolved Complaint: 7

Record: 7/7

Please set the min percentage of similarities (by dragging the slider below)

100%

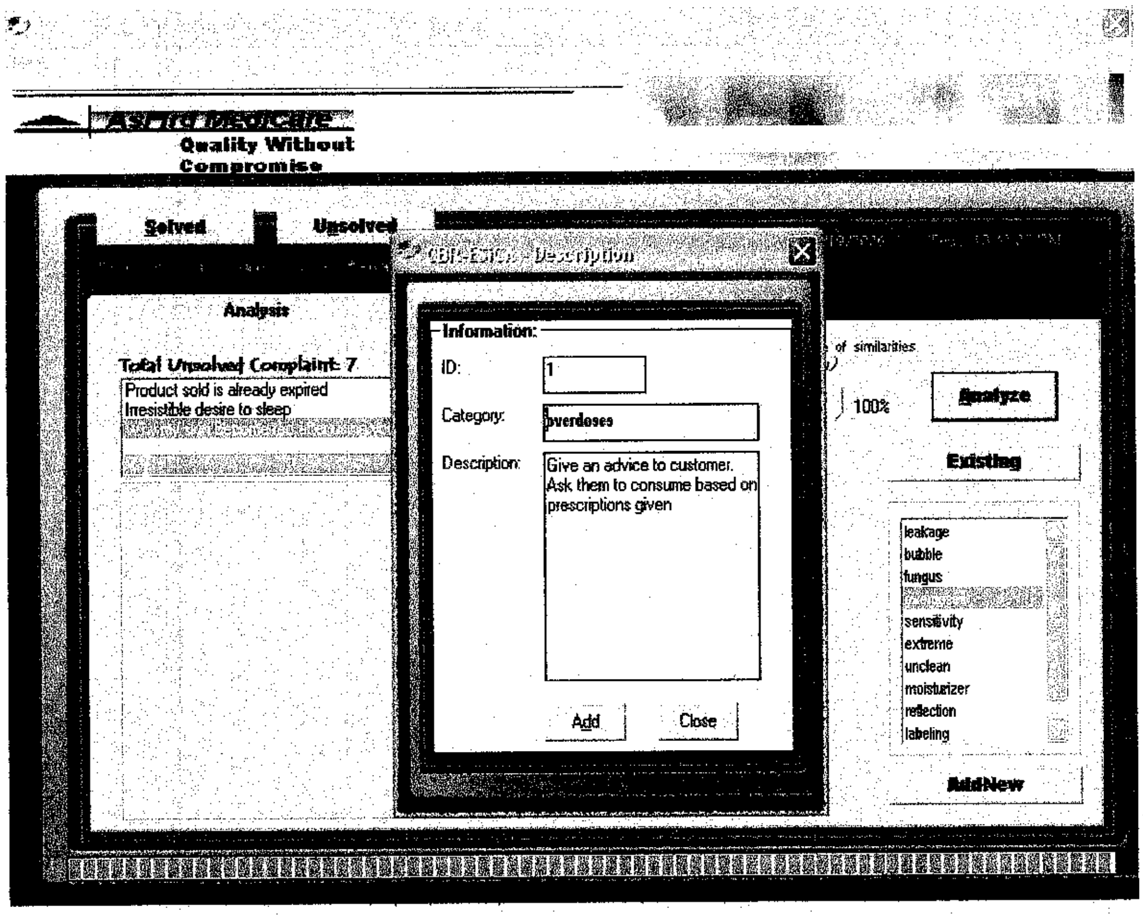
Product sold is already expired
Irresistible desire to sleep

No cases match.

Recommended Category:

Figure 1.4: Condition where no match cases

6. Click on “Existing” button to store new complaints under existing complaint category (Figure 1.5).
- System will list out all the existing categories that not yet listed in Recommended list.
- Form Description will be displayed when user click on one of Existing lists.



Total Unsolved Complaint: 7
Product sold is already expired
Irresistible desire to sleep

Information:
ID: 1
Category: overdoses
Description: Give an advice to customer. Ask them to consume based on prescriptions given

leakage
bubble
fungus
sensitivity
extreme
unclean
moisturizer
reflection
labeling

Figure 1.5: Description Form

7. Click on “Add New” button to add new category into the database (Figure 1.6)

- User needs to enter all required information into the form before click “Save” button.

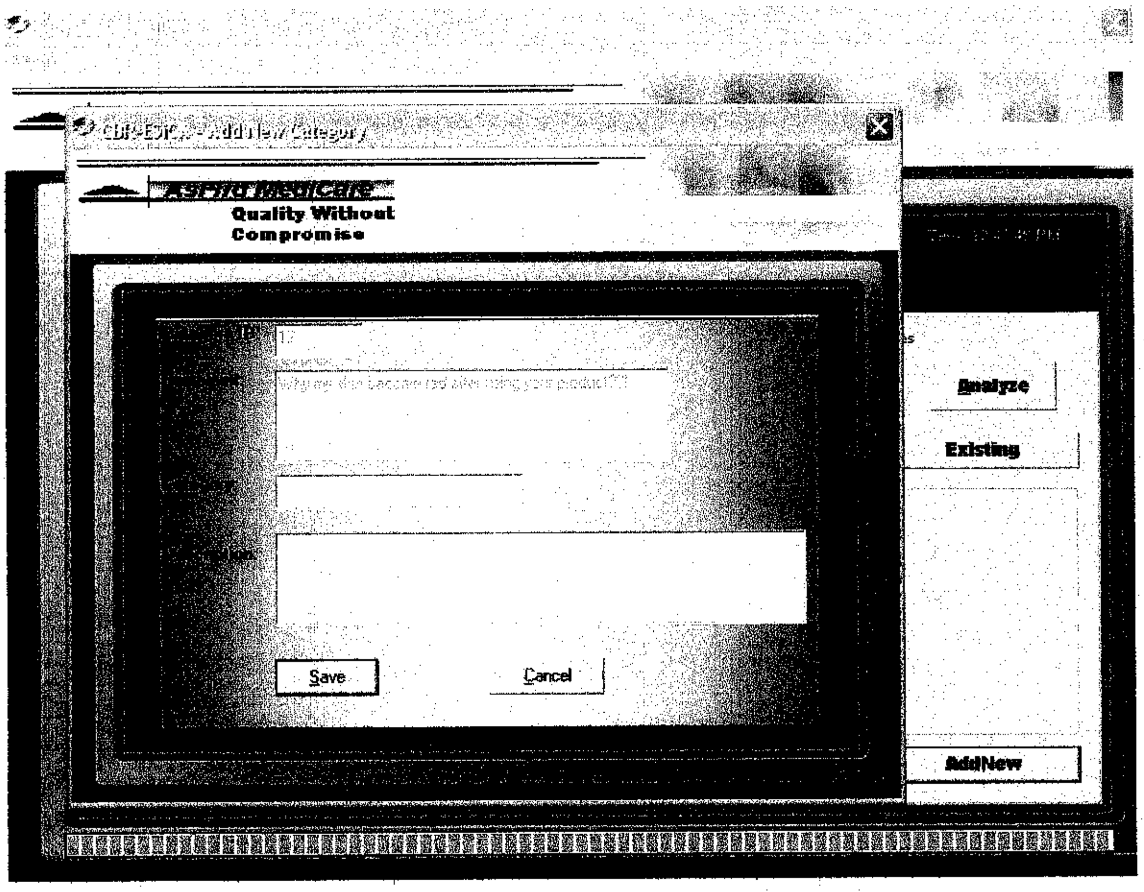
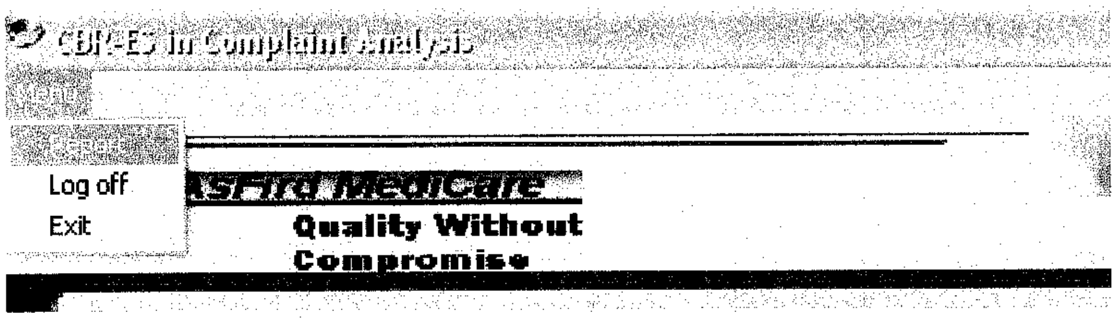
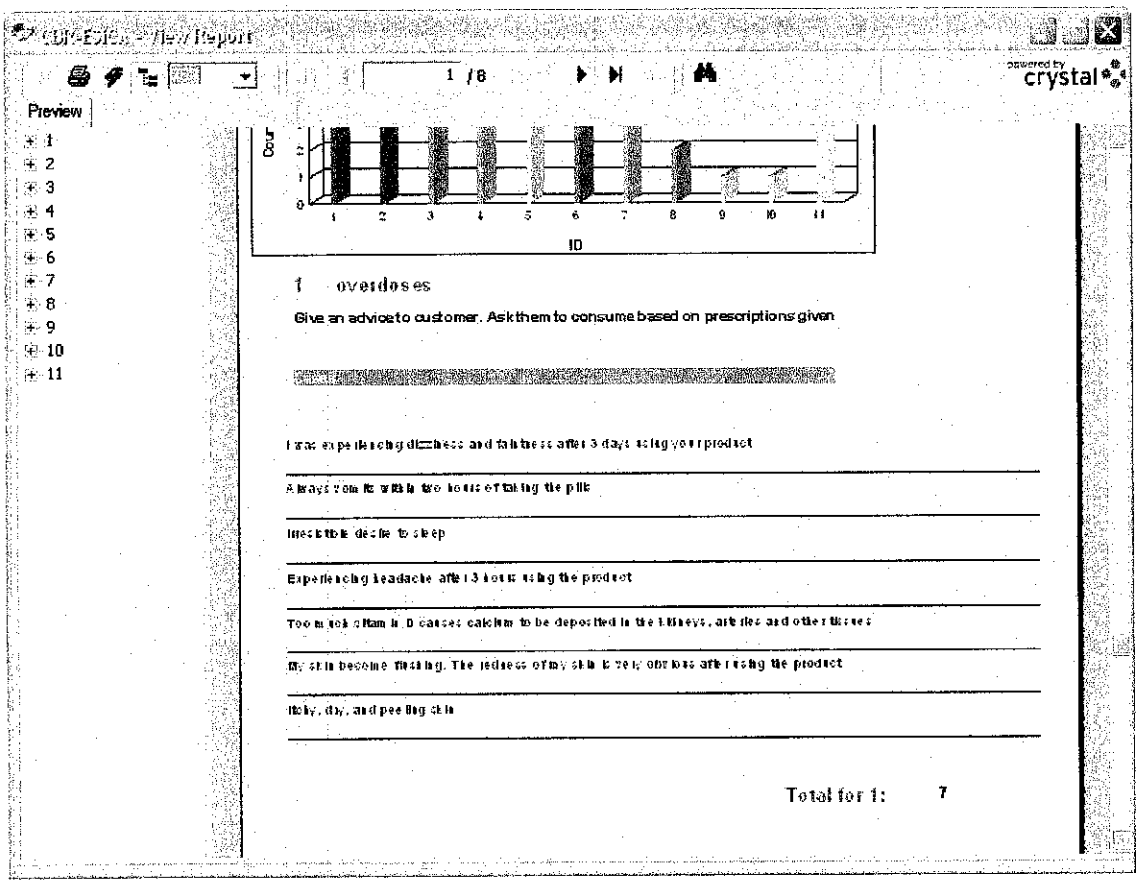


Figure 1.6: Add New form

- View Report

Customer Care Manager is able to display all complaints under each category in the database (Figure 1.7). From the report, Customer Care Manager may identify which category contributes most of the complaints.





1 overdoses

Give an advice to customer. Ask them to consume based on prescriptions given

I was experiencing dizziness and faintness after 3 days using your product

Always vomits within two hours of taking the pills

Irresistible desire to sleep

Experiencing headache after 3 hours using the product

Too much vitamin D causes calcium to be deposited in the kidneys, arteries and other tissues

My skin become flushing. The redness of my skin is very obvious after using the product

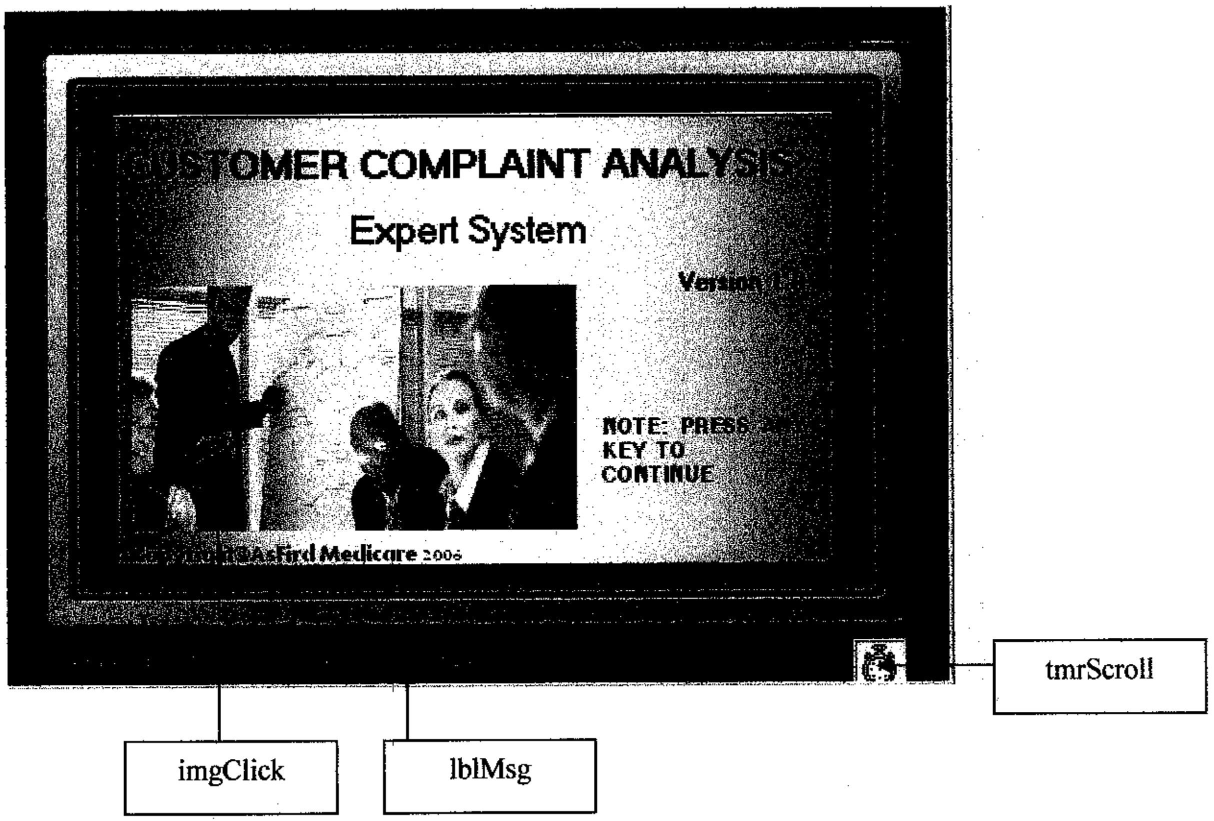
Itchy, dry, and peeling skin

Total for 1: 7

Figure 1.7: Example of report

GUI (Graphical User Interface) and Codes

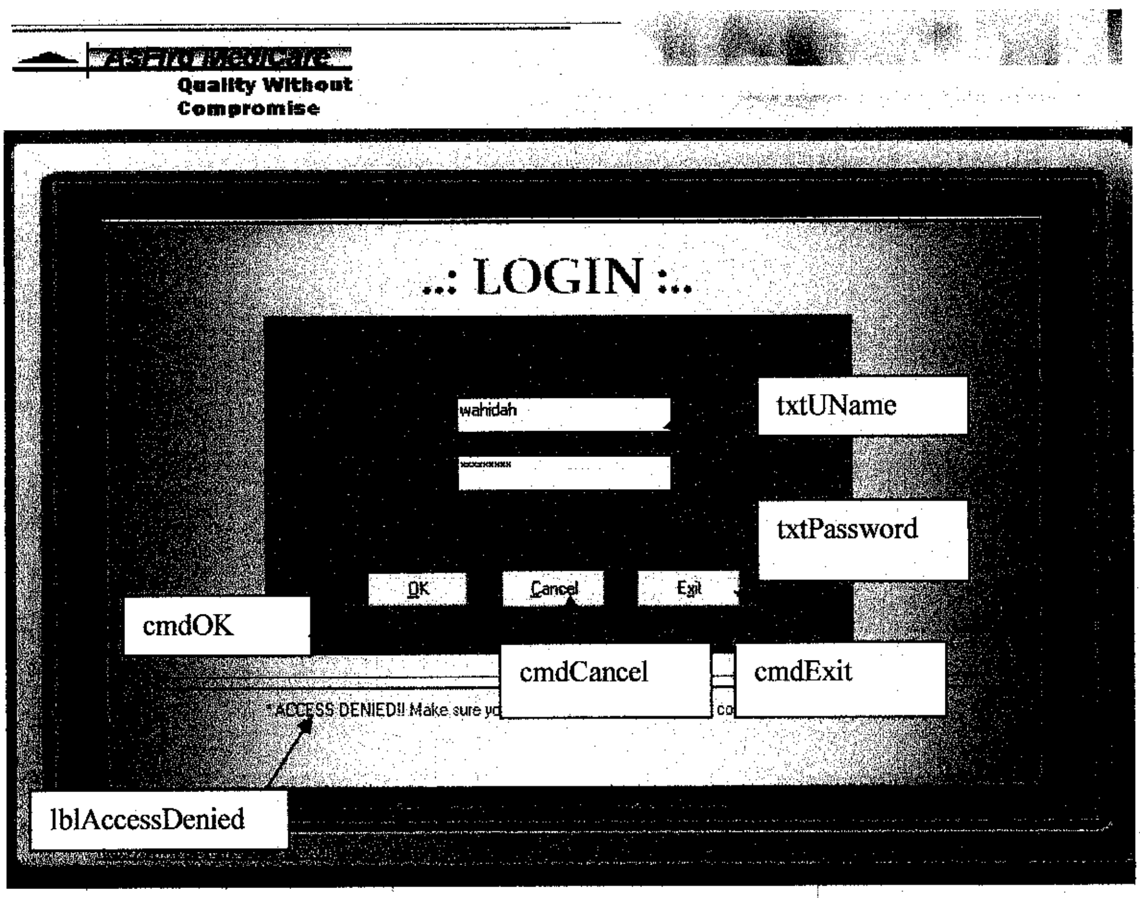
frmIntro:


CUSTOMER COMPLAINT ANALYSIS
Expert System
Version 1.0
NOTE: PRESS ANY KEY TO CONTINUE
Copyright@Asfird Medicare 2006
tmrScroll
imgClick
lblMsg

```
'*****************************************************
'*-------------------------------------------------------*
'*- Developer: Wahidah Padeli                           *
'*- ID: 3957                                            *
'*- Course: Business Information System                 *
'*- Title: CBR-ES in Complaint Analysis                 *
'*- Start date: July 2005                               *
'*- End date: Jun 2006                                  *
'*-------------------------------------------------------*
'*****************************************************

Option Explicit
Public sconnection As String
```

```
Private Sub Form_Load()

'to display scrolling text
lblMsg.Left = picHold.Width
tmrScroll.Interval = 10
tmrScroll.Enabled = True

End Sub
------------------------------------------------------------------------------------------------------------------------
Private Sub Form_KeyPress(KeyAscii As Integer)

'unload frmIntro when user press any key
Unload Me
frmLogin.Show

End Sub
------------------------------------------------------------------------------------------------------------------------
Private Sub Image1_Click()

'unload frmIntro when user click on the image
Unload Me
frmLogin.Show

End Sub
------------------------------------------------------------------------------------------------------------------------
Private Sub tmrScroll_Timer()

'set scrolling text at the bottom of frmIntro
If lblMsg.Left > -lblMsg.Width Then
    lblMsg.Left = lblMsg.Left - 10
  Else
    lblMsg.Left = picHold.Width
  End If

End Sub
------------------------------------------------------------------------------------------------------------------------
```

frmLogin:



"ACCESS DENIED!! Make sure yo

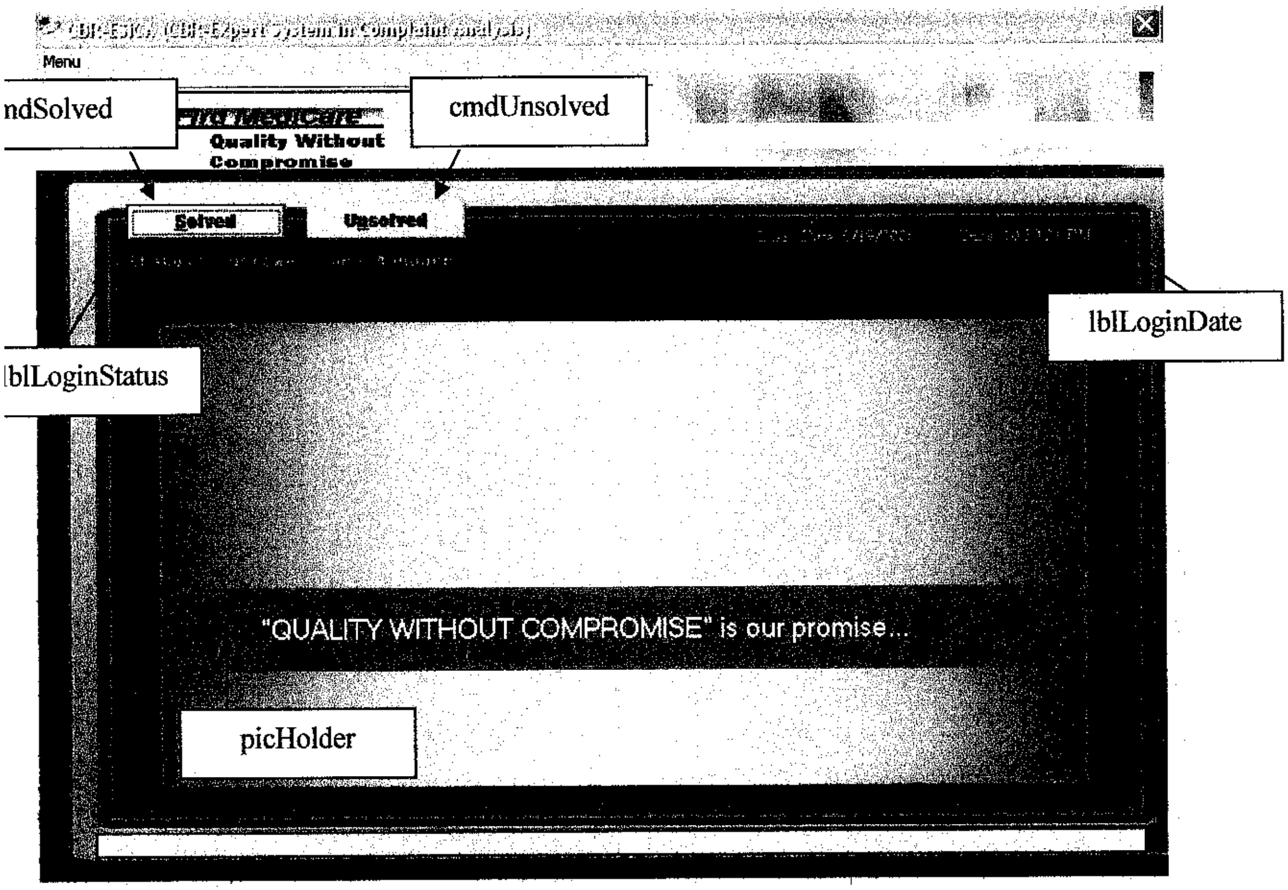
```
Private Sub cmdCancel_Click()

'clear previous data entered by user
txtUName.Text = ""
txtPassword.Text = ""
txtUName.SetFocus

End Sub
```

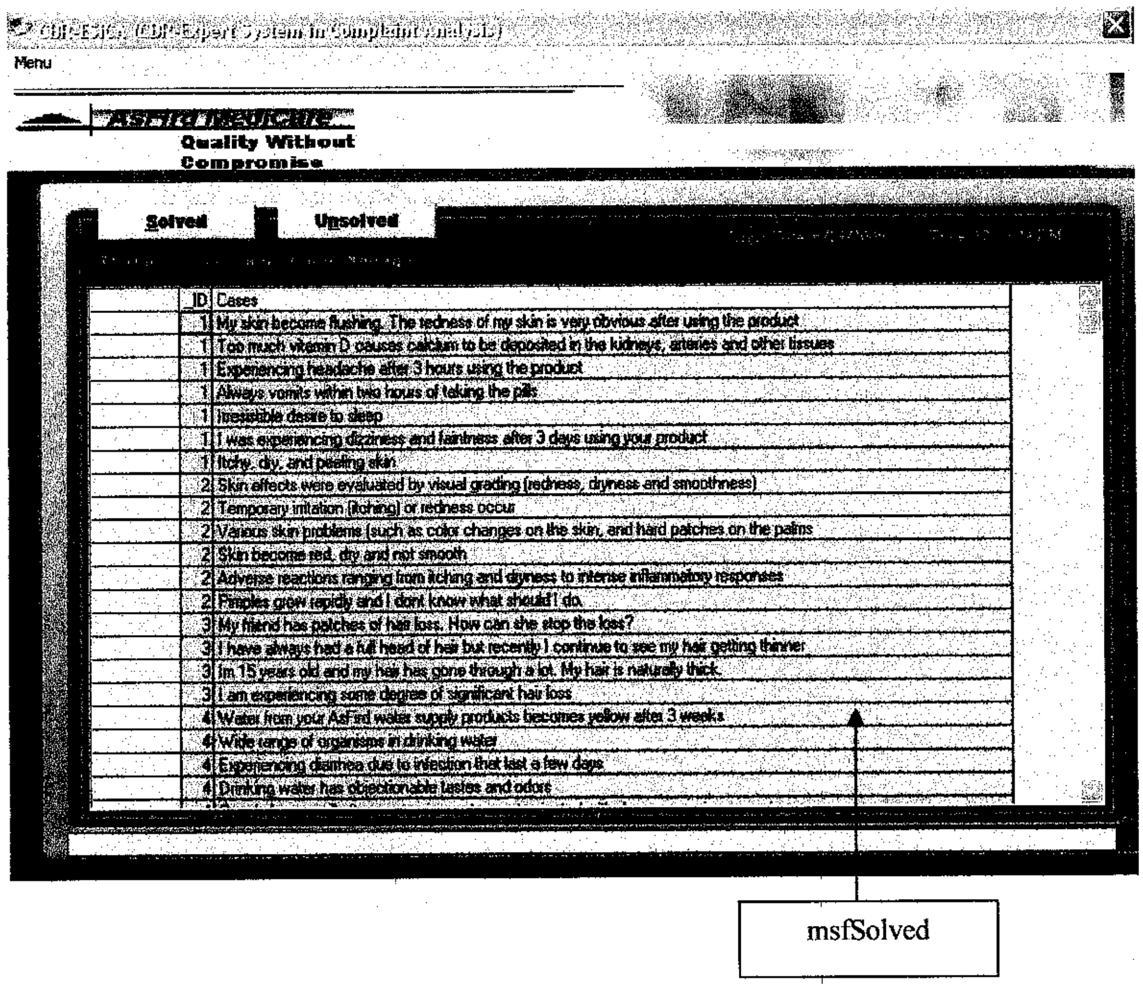
```
Private Sub cmdExit_Click()

'get user confirmation to exit from the system
response = MsgBox("Are you sure to exit from the system?", vbExclamation + vbYesNo, "CBR-ESiCA")

If response = vbYes Then
   End
Else
End If

End Sub
```

```
Private Sub cmdOK_Click()

frmDescription.Visible = False

UserName = UCase(txtUName.Text)
```

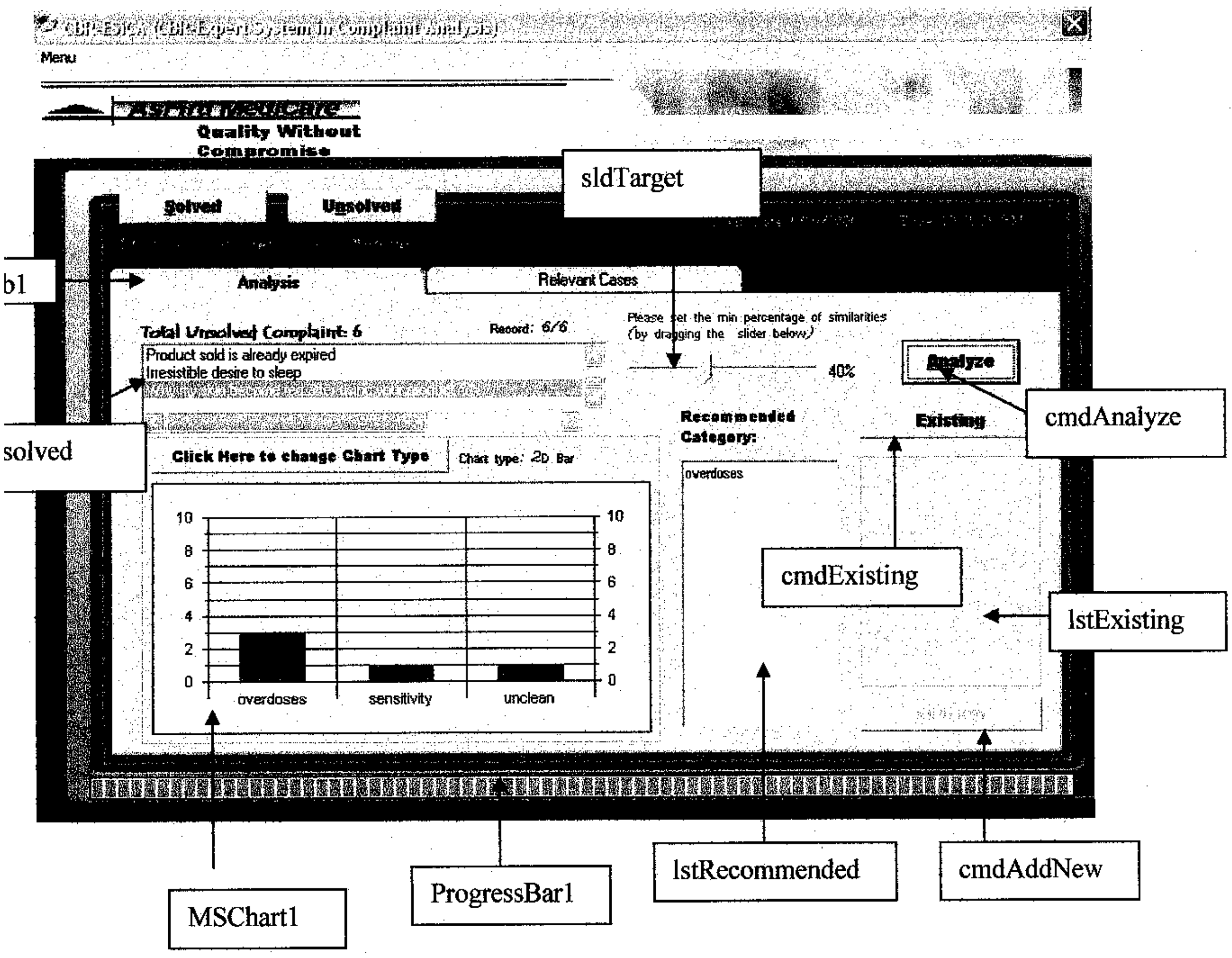
```
'check validity username and password
If UserName = "WAWA" And txtPassword.Text = "180903" Then

   Unload Me
   frmDescription.Visible = False
   frmtxtCompare.Show
   frmtxtCompare.submnuReport.Enabled = False
   frmtxtCompare.lblLoginStatus = "Status: Customer Care Officer"

ElseIf UserName = "WAHIDAH" And txtPassword.Text = "220584" Then

   Unload Me
   frmDescription.Visible = False
   frmtxtCompare.Show
   frmtxtCompare.lblLoginStatus = "Status: Customer Care Manager"

Else

   'if the login fail - display "ACCESS DENIED"
   MsgBox "ACCESS DENIED! Please re-login.", vbCritical + vbOKOnly, "ERROR!"
   lblAccessDenied.Visible = True
   txtUName.SetFocus

End If

End Sub
-------------------------------------------------------------------------------------------------------------
Private Sub Form_Load()

frmDescription.Visible = False

End Sub
-------------------------------------------------------------------------------------------------------------
```

frmtxtCompare:


CBR-ESICA (CBR-Expert System in Complaint analysis)
Menu
ndSolved
cmdUnsolved
Quality Without Compromise
Solved
Unsolved
lblLoginDate
lblLoginStatus
"QUALITY WITHOUT COMPROMISE" is our promise...
picHolder


CDR-ESICA (CDR-Expert System in Complaint Analysis)
Menu
AsFird Medicare
Quality Without Compromise
Solved
Unsolved
msfSolved

_ID	Cases
1	My skin become flushing. The redness of my skin is very obvious after using the product
1	Too much vitamin D causes calcium to be deposited in the kidneys, arteries and other tissues
1	Experiencing headache after 3 hours using the product
1	Always vomits within two hours of taking the pills
1	Irresistible desire to sleep
1	I was experiencing dizziness and faintness after 3 days using your product
1	Itchy, dry, and peeling skin
2	Skin effects were evaluated by visual grading (redness, dryness and smoothness)
2	Temporary irritation (itching) or redness occur
2	Various skin problems (such as color changes on the skin, and hard patches on the palms
2	Skin become red, dry and not smooth
2	Adverse reactions ranging from itching and dryness to intense inflammatory responses
2	Pimples grow rapidly and I dont know what should I do.
3	My friend has patches of hair loss. How can she stop the loss?
3	I have always had a full head of hair but recently I continue to see my hair getting thinner
3	Im 15 years old and my hair has gone through a lot. My hair is naturally thick.
3	I am experiencing some degree of significant hair loss
4	Water from your AsFird water supply products becomes yellow after 3 weeks
4	Wide range of organisms in drinking water
4	Experiencing diarrhea due to infection that last a few days
4	Drinking water has objectionable tastes and odors



CBR-ESICA (CBR-Expert System in Complaint Analysis)

Total Unsolved Complaint: 6

Record: 6/6

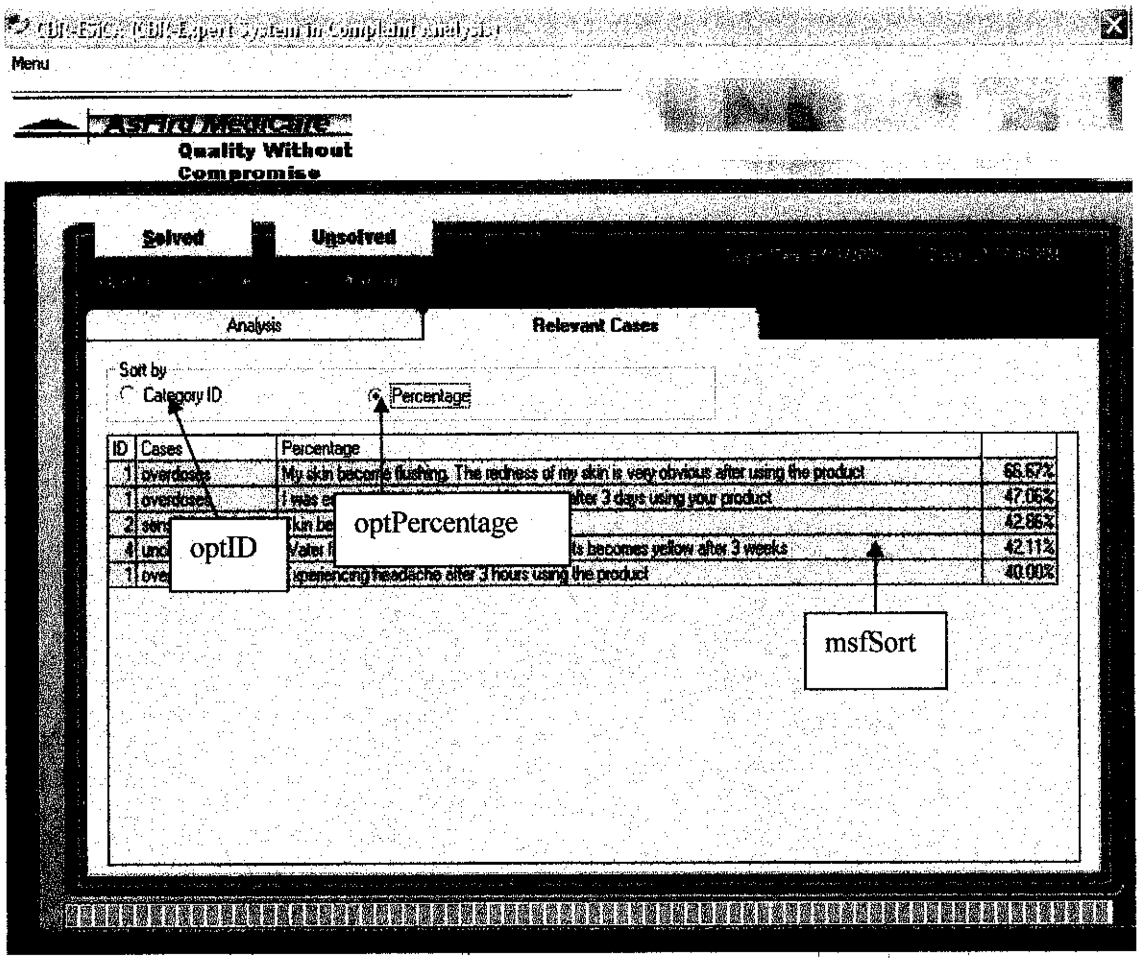
Product sold is already expired
Irresistible desire to sleep

Please set the min percentage of similarities (by dragging the slider below)

40%

Chart type: 2D Bar

overdoses



Sort by
Category ID [✓ Percentage]

ID	Cases	Percentage	
1	overdoses	My skin become flushing. The redness of my skin is very obvious after using the product	66.67%
1	overdoses	I was e… after 3 days using your product	47.06%
2	sens	kin be	42.86%
4	unc	Water f… ts becomes yellow after 3 weeks	42.11%
1	ove	xperiencing headache after 3 hours using the product	40.00%

```
'Global declaration

Public sconnection As String

Dim IDk(100) As String
Dim SQL As String
Dim response As String
Dim newCtr(50) As Integer
Dim iRow As Integer
Dim Xi() As Variant
Dim booOldRedraw As Boolean
Private values() As Variant
Private numpoints As Integer
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" _
    (ByVal hwnd As Long, ByVal wMsg As Long, _
    ByVal wParam As Long, lParam As Any) As Long
Private Const LB_FINDSTRINGEXACT = &H1A2
Const LB_SETHORIZONTALEXTENT = &H194

Public NoOfCases As Integer
```

```
Public Sub gSetHorizontalExtent(ByVal lstUnSolved As ListBox)

' Set the horizontal extent of the control (in pixel).
' If this value is greater than the current control's width
' an horizontal scrollbar appears.
SendMessage lstUnSolved.hwnd, LB_SETHORIZONTALEXTENT, 500, ByVal 0&

End Sub
```

```
Private Sub cmdAddNew_Click()

'call form to allow user to add new category
frmDescription.Hide
lstExisting.Visible = False
frmAddNew.Show

End Sub
------------------------------------------------------------------------------------------------------------------------
Private Sub cmdAnalyze_Click()

'private declaration
Dim Cat(100) As Integer
Dim ctr(50) As Integer
Dim iIndex As Long
Dim iMatch As Long
Dim iCopies As Long
Dim iHighest As Long
Dim aCommon() As Long
Dim sString As String
Dim bSkip As Boolean
Dim Value()

Dim tmpComplaint As String
Dim fgrid As Integer
Dim symbol(11) As String

'array for symbols
symbol(0) = "."
symbol(1) = ","
symbol(2) = "-"
symbol(3) = "!"
symbol(4) = "?"
symbol(5) = "_"
symbol(6) = "("
symbol(7) = ")"
symbol(8) = ":"
symbol(9) = ";"
symbol(10) = "'"
symbol(11) = """"

'clear the previous info
MSChart1.Refresh
lstRecommended.Clear
lstExisting.Clear
frmDescription.Visible = False
List7.Clear
lstAnalyzeResult.Clear


Dim sSQL As String

sSQL = "SELECT  * FROM tblComplaint WHERE Complaint Like'" & txtComplaint.Text & "%'"
adcComplaint.CommandType = adCmdText
adcComplaint.RecordSource = sSQL
adcComplaint.Refresh
adcComplaint.Recordset("Status") = 1
adcComplaint.Recordset.Update

If lstUnSolved.ListCount = 0 Then

    MsgBox "All complaints have been already solved.", vbInformation + vbOKOnly, "CBR-ESiCA"

    lstRecommended.Enabled = False
    lstRecommended.Visible = False
    lstExisting.Enabled = False
    cmdExisting.Enabled = False
    cmdAddNew.Enabled = False
    fraChart.Visible = False
    lstAnalyzeResult.Clear
    lstUnSolved.Clear
    lblTotalUnsolved.Caption = "Total Unsolved Complaint: 0"

    If msfSort.Rows = 2 Then
    Else
       For fgrid = msfSort.Rows To 2 Step -1
          msfSort.RemoveItem (fgrid)
       Next fgrid
    End If

Else

    'lstRecommended.Enabled = True
    lstExisting.Enabled = True
    cmdExisting.Enabled = True
    cmdAddNew.Enabled = False
    fraChart.Visible = True
    lblRecLabel.Enabled = True

    txtComplaint.Text = lstUnSolved.List(lstUnSolved.ListIndex)

    'get number of cases in db
    NoOfCases = adcCasesSplit.Recordset.RecordCount

    If msfSort.Rows = 2 Then
    Else
       For fgrid = msfSort.Rows To 2 Step -1
          msfSort.RemoveItem (fgrid)
       Next fgrid
    End If

    'reset control properties
    picHolder.Visible = False
    txtMsg.Visible = False
    cmdExisting.Enabled = True
    cmdAddNew.Enabled = False

    tmpComplaint = Trim(txtComplaint)

    For i = 0 To UBound(symbol)
       tmpComplaint = Trim(Replace(tmpComplaint, symbol(i), ""))
    Next i

    'split the complaint according to space
    trimmedcomplaint = Split(tmpComplaint, " ")

    'get total words in each complaint
    n = UBound(trimmedcomplaint) + 1
    ReDim simpan(n)

    'initialize no. of word in complaint as 0
    Count1 = 0

    'remove "noise words" from complaint
    For f = 0 To UBound(trimmedcomplaint)
    wd = trimmedcomplaint(f)
       If wd <> "" Then
          If Not Noise(LCase(wd)) Then
             Count1 = Count1 + 1
             simpan(Count1) = wd
          End If
       End If
    Next f

    'initialize no. of shared words as 0
    samewords = 0

    'variable to store shared words
    myStr = ""
```

```
num2 = 0
num = 0

For cmp = 0 To storenum

    'progress bar setup
    ProgressBar1.Value = num2 + 1
    DoEvents

    'split the cases according to space
    trimmedcases = Split(Caseslist(num2), " ")

    'get total words in each case
    m = UBound(trimmedcases) + 1
    ReDim simpan2(m)

    'initialize no. of word in each case as 0
    Count2 = 0

    'remove "noise words" from each case
    For fc = 0 To UBound(trimmedcases)
       wd = trimmedcases(fc)
       If Not Noise(LCase(wd)) Then
          Count2 = Count2 + 1
          simpan2(Count2) = wd
       End If
    Next fc


    'find shared words between sentences
    For f = 1 To Count1
       For fc = 1 To Count2
          Compare = InStr(1, simpan2(fc), simpan(f), vbTextCompare)
          If Compare <> 0 Then
             samewords = samewords + 1
             myStr = myStr & simpan2(fc) & ", "
             Exit For
          Else
          End If
       Next fc
    Next f

    'calculate percentage of similarities
    percentage = Format$((samewords * 100 / ((Count1 + Count2) / 2)), "fixed")

    If percentage > 100 Then
      percentage = 100
    End If

    'display category ID,relevant cases and percentage of similarities
    'for cases that have ***% and above of similarities

    If (sldTarget.Value <= percentage And percentage <= 100) Then

        SQL = "Select Cat_ID From tblCases Where Cases Like '" & Caseslist(num2) & "%'"
        adcCases.CommandType = adCmdText
        adcCases.RecordSource = SQL
        adcCases.Refresh

       'to set alignment
       tmpstr = Caseslist(num2) + Space(70)
       tmpstr = Left(tmpstr, 100)

       'list out all ID for relevant cases
       ID(num2) = adcCases.Recordset!Cat_ID

       SQL = "Select Category From tblCategory Where ID = " & ID(num)

        adcCategory.CommandType = adCmdText
        adcCategory.RecordSource = SQL
        adcCategory.Refresh

        Cattt(num2) = adcCategory.Recordset!Category


        'display ID, relevant cases and percentage of similarities
        lstAnalyzeResult.AddItem "(" & ID(num2) & ")" & Space(3) & Cattt(num) & _
                      Space(3) & tmpstr & percentage & "%"
        msfSort.AddItem ID(num2) & Chr(9) & Cattt(num2) & Chr(9) & tmpstr & Chr(9) & percentage & "%"

        frmtxtCompare.List7.AddItem adcCases.Recordset!Cat_ID

        If percentage = 100 Then
          cmdExisting.Enabled = False
        End If

      End If

      num2 = num2 + 1
      num = num + 1

      samewords = 0

    Next cmp


    If msfSort.Rows > 2 Then
      msfSort.RemoveItem (1)
    End If

    'sorting - default: by category ID
    optID_Click

    'max of progress bar
    ProgressBar1.Max = num2

    'no suggestion if no relevant case exist
    If lstAnalyzeResult.ListCount = 0 Then
      response = MsgBox("No cases match.", vbOKOnly + vbInformation, "CBR-ESiCA")

      If msfSort.Rows = 2 Then
      Else
        For fgrid = msfSort.Rows To 2 Step -1
          msfSort.RemoveItem (fgrid)
        Next fgrid
      End If

      lstRecommended.AddItem ""
      lstRecommended.Visible = False
      cmdExisting.Enabled = True
      cmdAddNew.Enabled = True
      cmdChangeType.Visible = False
      lblChartType.Visible = False
      lblRecLabel.Enabled = False
      MSChart1.Visible = False
      lstRecommended.Enabled = False

    Else
      lstRecommended.Visible = True
      lstRecommended.Enabled = True
      cmdChangeType.Visible = True

      ' Use manual scale to display y axis (value axis)
      With frmtxtCompare.MSChart1.Plot.Axis(VtChAxisIdY).ValueScale
        .Auto = False
        .Minimum = 0
        .Maximum = 10
        .MinorDivision = 2
        .MajorDivision = 5
        End With

        MSChart1.Visible = True
        For d = 0 To List7.ListCount - 1
           Cat(d) = List7.List(d)
           If d >= 0 Then
              ctr(Cat(d)) = ctr(Cat(d)) + 1
           End If
        Next d

        CTRCounter = 0
        For RC = 1 To UBound(ctr)
           If ctr(RC) > 0 Then
              CTRCounter = CTRCounter + 1
           End If
        Next RC

        MSChart1.RowCount = CTRCounter

           'This sets the number of columns per row.
         MSChart1.ColumnCount = 1

        'This indicates to show the label
        MSChart1.ShowLegend = False
        fraChart.Visible = True
        MSChart1.Visible = True
        lblChartType.Visible = True

        CTRCounter = 0

        For RC = 1 To UBound(ctr)
           If ctr(RC) > 0 Then
              CTRCounter = CTRCounter + 1
              MSChart1.Row = CTRCounter
              SQL = "select * from tblCategory where ID =" & RC
              adcCategory.CommandType = adCmdText
              adcCategory.RecordSource = SQL
              adcCategory.Refresh

              MSChart1.RowLabel = adcCategory.Recordset.Fields("Category")

              Call MSChart1.DataGrid.SetData(CTRCounter, 1, ctr(RC), nullflag)

           End If
         Next RC
     End If

     'find most repeated category
     For iIndex = 0 To List7.ListCount - 1
        iCopies = 0
        iMatch = -1
        bSkip = False

        'Skip this one if it's the same as the last Item Checked
        If iIndex Then
           bSkip = (List7.List(iIndex) = List7.List(iIndex - 1))
        End If

        'Skip this one if there's a previous instance of it in the List
        If Not bSkip Then
           bSkip = (SendMessage(List7.hwnd, LB_FINDSTRINGEXACT, -1, _
                 ByVal List7.List(iIndex)) < iIndex)
        End If

        'While there are other Instances in the List..
        While iMatch <> iIndex And Not bSkip
           'Increment the No of Copies Found of this Item
           iCopies = iCopies + 1
           'Find the next Copy..
           iMatch = SendMessage(List7.hwnd, LB_FINDSTRINGEXACT, _
        IIf(iMatch < 0, iIndex, iMatch), _
        ByVal List7.List(iIndex))
  Wend

  'If there were more than 1 Copies
  If iCopies > 1 And Not bSkip Then
    'If the No. of Copies is Greater or the Same as the Highest so far..
    If iCopies >= iHighest Then
      If iCopies > iHighest Then
        'new Highest Copies
        ReDim aCommon(0)
      Else
        'Another Item with the same highest amount of Copies
        ReDim Preserve aCommon(UBound(aCommon) + 1)
      End If

      'Store this Index
      aCommon(UBound(aCommon)) = iIndex

      'Remember the Highest No. of Copies
      iHighest = iCopies
    End If
  End If
Next

If iHighest Then
  'If Copies were Found..list out the suggested category
  For iIndex = 0 To UBound(aCommon)
    adc2tables.RecordSource = "Select (Category) From tblCategory where ID = " & frmtxtCompare.List7.List(aCommon(iIndex))
    adc2tables.Refresh
    lstRecommended.AddItem adc2tables.Recordset.Fields("Category")
    sString = sString & ", " & frmtxtCompare.List7.List(aCommon(iIndex))
  Next
  'frmtxtCompare.txtRepeated.Text = Mid$(sString, 3) & "- Category: " & frmtxtCompare.adc2tables.Recordset.Fields("Category")
Else

  For k = 0 To List7.ListCount - 1
    adc2tables.RecordSource = "Select (Category) From tblCategory where ID = " & frmtxtCompare.List7.List(k)
    adc2tables.Refresh
    frmtxtCompare.lstRecommended.AddItem adc2tables.Recordset.Fields("Category")
    'frmtxtCompare.txtRepeated.Text = "Each one is appropriate"
  Next k

End If

End If

End Sub
```

```
Private Function Noise(ByVal wd As String) As Boolean

'declare an array for "noise words"
Dim nix(27) As String

'list of "noise words"
nix(0) = "a"
nix(1) = "an"
nix(2) = "the"
nix(3) = "of"
nix(4) = "in"
nix(5) = "for"
nix(6) = "with"
nix(7) = "to"
nix(8) = "from"
nix(9) = "is"
nix(10) = "was"
nix(11) = "are"
nix(12) = "were"
nix(13) = "and"
nix(14) = "that"
nix(15) = "which"
nix(16) = "or"
nix(17) = "by"
nix(18) = "i"
nix(19) = "my"
nix(20) = "have"
nix(21) = "has"
nix(22) = "been"
nix(23) = "am"
nix(24) = "on"
nix(25) = "but"
nix(26) = "because"
nix(27) = "your"

'set default boolean value for Noise function
Noise = False

'find "noise words" in sentences
For f = 0 To 26
   If (StrComp(wd, nix(f)) = 0) Then
      Noise = True
      Exit For
   End If
Next f

End Function
-----------------------------------------------------------------------------------------------------------------
Private Sub cmdChangeType_Click()

If MSChart1.chartType = VtChChartType2dBar Then
   MSChart1.chartType = VtChChartType2dArea
   lblChartType.Visible = True
   lblChartType.Caption = "Chart type = 2D Area"

Else
   MSChart1.chartType = VtChChartType2dBar
   lblChartType.Visible = True
   lblChartType.Caption = "Chart type = 2D Bar"
End If

End Sub
-----------------------------------------------------------------------------------------------------------------
Private Sub cmdExisting_Click()

Dim NoOfCategory As Integer
Dim NoOfSuggestion As Integer
Dim suggested As Integer
Dim existing As Integer
Dim exist() As String
Dim suggest() As String
Dim escCtr(10) As String

NoOfCategory = adcExistingCat.Recordset.RecordCount
ReDim exist(NoOfCategory)

NoOfSuggestion = lstRecommended.ListCount - 1
ReDim suggest(NoOfSuggestion)

lstExisting.Visible = True
lstExisting.Clear

'list out all existing category
adcExistingCat.Recordset.MoveFirst
While Not adcExistingCat.Recordset.EOF
   exist(existing) = adcExistingCat.Recordset!Category
   lstExisting.AddItem exist(existing)
   adcExistingCat.Recordset.MoveNext
Wend

If lstAnalyzeResult.ListCount = 0 Then
```

```
Else

'remove category if it is already listed in lstRecommended
   For suggested = 0 To lstRecommended.ListCount - 1
      suggest(suggested) = lstRecommended.List(suggested)

      existingcount = lstExisting.ListCount - 1
      For existing = existingcount To 0 Step -1
         exist(existing) = lstExisting.List(existing)
         Compare = InStr(1, exist(existing), suggest(suggested), vbTextCompare)

         If Compare = 1 Then
            lstExisting.RemoveItem (existing)
         Else
         End If

      Next existing

   Next suggested
End If

End Sub
--------------------------------------------------------------------------------------------------------
Private Sub cmdSolved_Click()

msfSolved.Visible = True
SSTabMain.Visible = False
picHolder.Visible = False
txtMsg.Visible = False

Dim SQL As String

'SQL = "Select * From tblComplaint Where Status = 1"
SQL = "Select * From tblCases"
adcComplaint2.CommandType = adCmdText
adcComplaint2.RecordSource = SQL
adcComplaint2.Refresh
Set msfSolved.DataSource = adcComplaint2

End Sub
--------------------------------------------------------------------------------------------------------
Private Sub cmdUnsolved_Click()

lblTotalUnsolved.Caption = "Total Unsolved Complaint: 0"
lblRecStatus.Caption = "Record: 0/0"

msfSolved.Visible = False
SSTabMain.Visible = True
picHolder.Visible = False
txtMsg.Visible = False
cmdExisting.Enabled = False
cmdAddNew.Enabled = False
lblRecLabel.Enabled = False
fraChart.Visible = False

countno2 = 0

Dim SQL As String

SQL = "Select * From tblComplaint Where Status = 0"
adcComplaint.RecordSource = SQL
adcComplaint.CommandType = adCmdText
adcComplaint.Refresh

lstUnSolved.Clear

If adcComplaint.Recordset.BOF And adcComplaint.Recordset.EOF Then
   Exit Sub
End If
```

```
adcComplaint.Recordset.MoveFirst

Do Until adcComplaint.Recordset.EOF
   lstUnSolved.AddItem adcComplaint.Recordset!Complaint
   adcComplaint.Recordset.MoveNext
Loop

lblTotalUnsolved.Caption = "Total Unsolved Complaint: " & lstUnSolved.ListCount

lstUnSolved.ListIndex = 0

'call 1st case in db
adcCasesSplit.Recordset.Sort = "Cat_ID"
adcCasesSplit.Recordset.MoveFirst

num = 0

'do the process until last record
While Not (adcCasesSplit.Recordset.EOF)

   'assign cases into caseslist array
   Caseslist(num) = adcCasesSplit.Recordset.Fields!Cases

   num = num + 1

   'move to next case
   adcCasesSplit.Recordset.MoveNext

   store(countno2) = Count2

   countno2 = countno2 + 1

Wend

storenum = num

End Sub
--------------------------------------------------------------------------------------------------------------

Private Sub Form_Load()

frmDescription.Visible = False

'Create Horizontal scroll bar on the Activity Log list box
gSetHorizontalExtent lstUnSolved

sconnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path & "\dbTextCompare.mdb;" & "Persist Security
Info=False"

frmtxtCompare.adcCases.ConnectionString = sconnection
frmtxtCompare.adcCasesSplit.ConnectionString = sconnection
frmtxtCompare.adc2tables.ConnectionString = sconnection
frmtxtCompare.adcComplaint.ConnectionString = sconnection
frmtxtCompare.adcCategory.ConnectionString = sconnection
frmtxtCompare.adcExistingCat.ConnectionString = sconnection
frmtxtCompare.adcComplaint2.ConnectionString = sconnection

'sort ID in database and display automatic ID for new record
adcCases.Recordset.Sort = "Cat_ID"
adcCases.Recordset.MoveFirst

'display value of benchmark - default = 0%
sldTarget.Value = 1
lblBenchmark = sldTarget.Value & "%"

'set default value to sort by category ID
optID.Value = True

lstExisting.Visible = False
lstRecommended.Visible = False
```

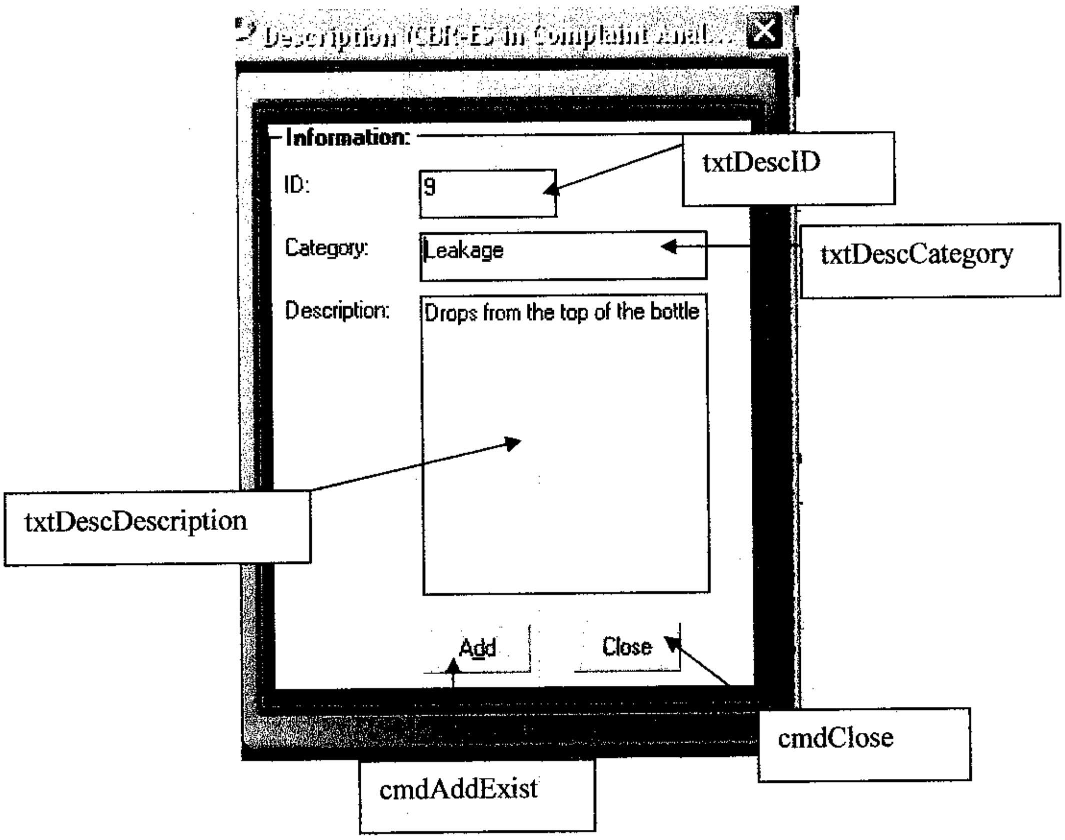
```
cmdExisting.Enabled = False
cmdAddNew.Enabled = False

'scrolling text
txtMsg.Left = picHolder.Width
tmrScroll.Interval = 10
tmrScroll.Enabled = True

'set hierarchical flexgrid properties
With msfSort
   .Cols = 4 'no of column
   .TextMatrix(0, 0) = "ID"   'header column index 0
   .TextMatrix(0, 1) = "Category"
   .TextMatrix(0, 1) = "Cases" 'header column index 1
   .TextMatrix(0, 2) = "Percentage" 'header column index 2
   .ColWidth(0) = 300 'width column index 0
   .ColWidth(1) = 1500 'width column index 1
   .ColWidth(2) = 7500 'width column index 2
   .ColWidth(3) = 800 'width column index 3
End With

With msfSolved
   .ColWidth(1) = 350 'width column index 1
   .ColWidth(2) = 8500
End With

End Sub
--------------------------------------------------------------------------------
Private Sub Form_Unload(Cancel As Integer)

frmLogin.Show

End Sub
--------------------------------------------------------------------------------
Private Sub lstExisting_Click()

frmDescription.Show
frmDescription.cmdAddExist.Enabled = True

'display information for the chosen category
SQL = "select * from tblCategory where Category Like '" & lstExisting.Text & "%'"
frmDescription.adcDescription.CommandType = adCmdText
frmDescription.adcDescription.RecordSource = SQL
frmDescription.adcDescription.Refresh
frmDescription.txtDescCategory = frmDescription.adcDescription.Recordset.Fields("Category")
frmDescription.txtDescDescription = frmDescription.adcDescription.Recordset.Fields("Description")
frmDescription.txtDescID = frmDescription.adcDescription.Recordset.Fields("ID")

End Sub
--------------------------------------------------------------------------------
Private Sub lstRecommended_Click()

frmDescription.Show
frmDescription.cmdAddExist.Enabled = False

'display information for the chosen category
SQL = "select * from tblCategory where [Category] ='" & frmtxtCompare.lstRecommended.Text & "'"
frmDescription.adcDescription.CommandType = adCmdText
frmDescription.adcDescription.RecordSource = SQL
frmDescription.adcDescription.Refresh
frmDescription.txtDescCategory = frmDescription.adcDescription.Recordset.Fields("Category")
frmDescription.txtDescDescription = frmDescription.adcDescription.Recordset.Fields("Description")
frmDescription.txtDescID = frmDescription.adcDescription.Recordset.Fields("ID")

End Sub

--------------------------------------------------------------------------------
```

```
Private Sub lstUnSolved_Click()

txtComplaint.Text = lstUnSolved.List(lstUnSolved.ListIndex)
lblRecStatus.Caption = "Record: " & lstUnSolved.ListIndex + 1 & "/" & lstUnSolved.ListCount

End Sub
------------------------------------------------------------------------------------------------------------------------------
Private Sub msfSolved_Click()

'sort column category ID - ascending
msfSolved.Col = 1
msfSolved.Sort = 3

End Sub
------------------------------------------------------------------------------------------------------------------------------
Public Sub optID_Click()

'sort column category ID - ascending
msfSort.Col = 0
msfSort.Sort = 3
msfSort.ColSel = msfSort.Cols - 1

End Sub
------------------------------------------------------------------------------------------------------------------------------
Private Sub optPercentage_Click()

'sort column percentage - descending
msfSort.Col = 3
msfSort.Sort = 2
msfSort.ColSel = msfSort.Cols - 1

End Sub
------------------------------------------------------------------------------------------------------------------------------
Private Sub sldTarget_Change()

'set value for benchmark set by user
lblBenchmark.Caption = sldTarget.Value & "%"

End Sub
------------------------------------------------------------------------------------------------------------------------------
Private Sub sldTarget_Click()

'set value for benchmark set by user
lblBenchmark.Caption = sldTarget.Value & "%"

End Sub
------------------------------------------------------------------------------------------------------------------------------
Private Sub submnuExit_Click()

frmDescription.Visible = False

'get user confirmation to continue analysis
response = MsgBox("Are you sure to exit from the system?", vbExclamation + vbYesNo, "CBR-ESiCA")
If response = vbYes Then
  End
Else
End If

End Sub
------------------------------------------------------------------------------------------------------------------------------
Private Sub submnuLogoff_Click()

'get user confirmation
response = MsgBox("Are you sure to log off?", vbExclamation + vbYesNo, "Log off confirmation..")
If response = vbYes Then
  Unload Me
  frmLogin.Show
Else
End If
End Sub
------------------------------------------------------------------------------------------------------------------------
Private Sub submnuReport_Click()

valbar = 0

frmProcessing.Show
frmReportViewer.Show
frmProcessing.Hide

End Sub
------------------------------------------------------------------------------------------------------------------------
Private Sub Timer1_Timer()

'set current login date and time
lblLoginDate.Caption = "Login Date: " & Date & Space(10) & "Time: " & Time

End Sub
------------------------------------------------------------------------------------------------------------------------
Private Sub tmrScroll_Timer()

'set scrolling text
If txtMsg.Left > -txtMsg.Width Then
      txtMsg.Left = txtMsg.Left - 10
   Else
      txtMsg.Left = picHolder.Width
   End If

End Sub
------------------------------------------------------------------------------------------------------------------------
'sort flexgrid by clicking the column directly
Private Sub msfSort_MouseUp(Button As Integer, Shift As _
   Integer, x As Single, y As Single)

' If this is not row 0, do nothing.
If msfSort.MouseRow <> 0 Then Exit Sub

' Sort by the clicked column.
SortByColumn msfSort.MouseCol

End Sub
------------------------------------------------------------------------------------------------------------------------
'Sort by the indicated column.
Private Sub SortByColumn(ByVal sort_column As Integer)

'Hide the FlexGrid.
msfSort.Visible = False
msfSort.Refresh

'Sort using the clicked column.
msfSort.Col = sort_column
msfSort.ColSel = sort_column
msfSort.Row = 0
msfSort.RowSel = 0

' If this is a new sort column, sort ascending.
' Otherwise switch which sort order we use.
If m_SortColumn <> sort_column Then
      m_SortOrder = flexSortGenericAscending
   ElseIf m_SortOrder = flexSortGenericAscending Then
      m_SortOrder = flexSortGenericDescending
   Else
      m_SortOrder = flexSortGenericAscending
End If

msfSort.Sort = m_SortOrder

' Restore the previous sort column's name.
If m_SortColumn >= 0 Then
   msfSort.TextMatrix(0, m_SortColumn) = _
```

```
    Mid$(msfSort.TextMatrix(0, m_SortColumn), 3)
End If

' Display the new sort column's name.
m_SortColumn = sort_column
If m_SortOrder = flexSortGenericAscending Then
   msfSort.TextMatrix(0, m_SortColumn) = "> " & _
      msfSort.TextMatrix(0, m_SortColumn)
Else
   msfSort.TextMatrix(0, m_SortColumn) = "< " & _
      msfSort.TextMatrix(0, m_SortColumn)
End If

' Display the FlexGrid.
msfSort.Visible = True

End Sub
```

---

frmDescription:



9

Leakage

Drops from the top of the bottle

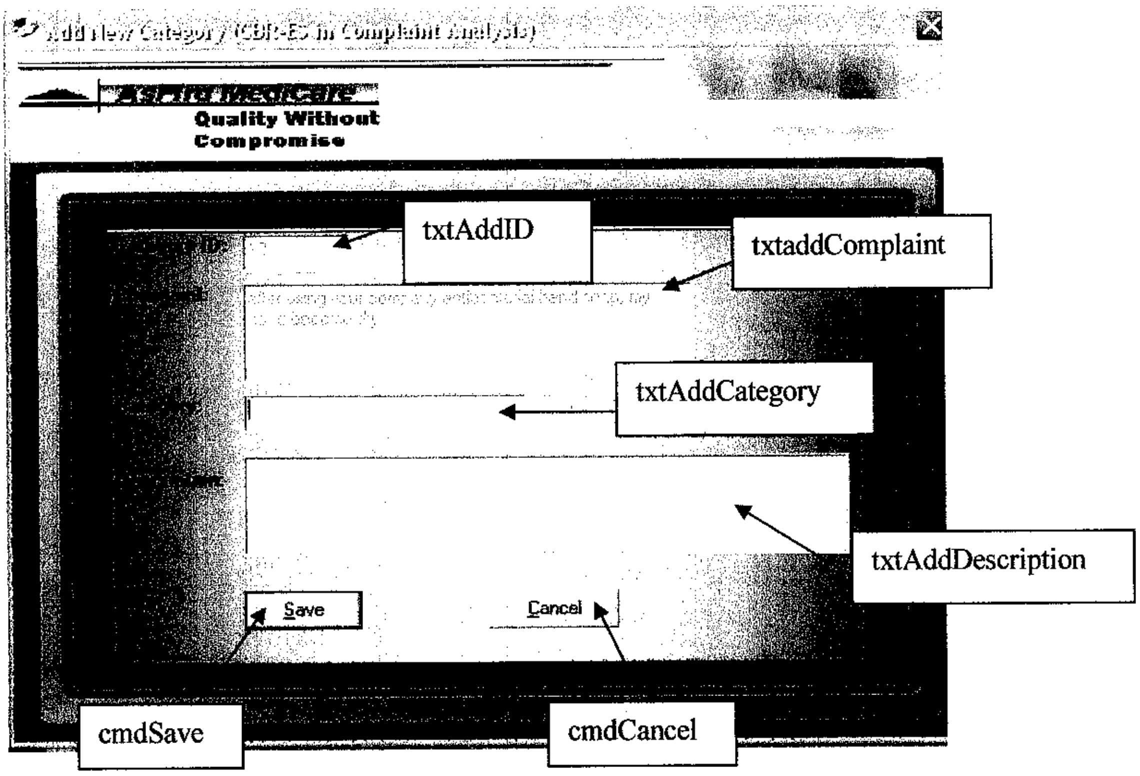
```
Private Sub cmdAddExist_Click()

txtAssignComplaint = frmtxtCompare.txtComplaint

'add new case to existing category
With frmDescription.adcAddCases.Recordset
   !Cat_ID = frmDescription.txtDescID.Text
   !Cases = txtAssignComplaint
   .Update
End With
```

```
'confirmation message that new category is successfully saved
MsgBox "New category have been successfully saved.", vbInformation + vbOKOnly, "CBR-ES in Complaint Analysis"
Me.Hide

End Sub
```

```
Private Sub cmdClose_Click()

'unload form when user click "close" button
Unload Me

End Sub
```

---

frmAddNew:



```
'Global declaration

Dim FirstRec As Variant
Public sconnection As String
```

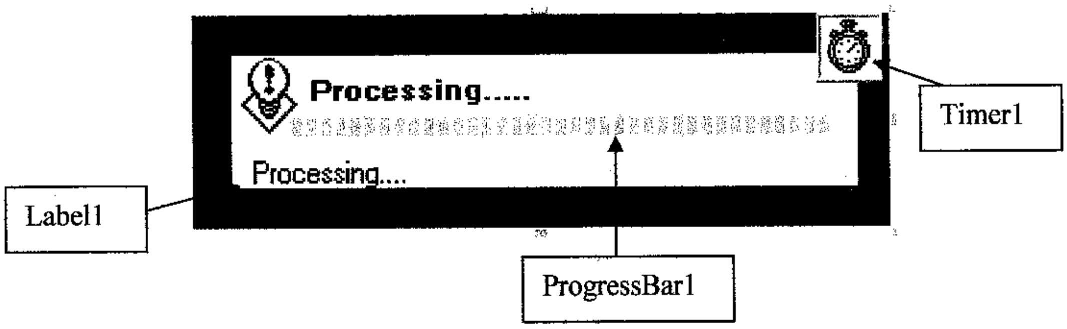
```
'To clear the previous data entered by user
Private Sub cmdCancel_Click()
frmAddNew.txtAddCategory.Text = ""
frmAddNew.txtAddComplaint.Text = ""
frmAddNew.txtAddDescription.Text = ""

Unload Me

End Sub
```

```
Private Sub cmdSave_Click()

'Check to ensure all required information are entered
If frmAddNew.txtAddID.Text = "" Or _
   frmAddNew.txtAddCategory.Text = "" Or _
   frmAddNew.txtAddComplaint.Text = "" Or _
   frmAddNew.txtAddDescription.Text = "" Then

   MsgBox "Please enter all required information.", vbExclamation + vbOKOnly, "Incomplete Information"

Else

   'to check whether the category already exisit or not
   frmAddNew.adcCheckCategory.Recordset.Find "Category like '" & LCase(txtAddCategory.Text) & "%'" _
   , , , FirstRec

   'no same category in database
   If adcCheckCategory.Recordset.EOF Then

      'add new category and description into tblCategory
      With adcAddNew_Category.Recordset
         .AddNew
         !Category = txtAddCategory
         !Description = txtAddDescription
      End With

      'add new cases into tblCases
      With adcAddNew_Cases.Recordset
         .AddNew
         !Cat_ID = txtAddID
      End With

      With adcAddNew_Cases.Recordset
         .AddNew
         !Cases = txtAddComplaint
      End With

      'confirmation message that new category is successfully saved
      MsgBox "New category has been successfully saved.", vbInformation + vbOKOnly, "CBR-ESiCA "
      Me.Hide

   Else

      'display message that category is already exits
      MsgBox "The category is already exist. Please use another category.", vbExclamation + vbOKOnly, "REDUNDANT
 DETECTED!"

      txtAddCategory.SetFocus

   End If
End If

End Sub
------------------------------------------------------------------------------------------------------------------------
Private Sub Form_Load()

sconnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path & "\dbTextCompare.mdb;" & "Persist Security
Info=False"

adcAddNew_Cases.ConnectionString = sconnection
adcAddNew_Category.ConnectionString = sconnection
adcCheckCategory.ConnectionString = sconnection

'sort ID in database and display automatic ID for new record
adcAddNew_Category.Recordset.Sort = "ID"
adcAddNew_Category.Recordset.MoveLast
NewID = adcAddNew_Category.Recordset.Fields("ID")
NewID = NewID + 1
adcAddNew_Cases.Recordset.AddNew
adcAddNew_Category.Recordset.AddNew
```

```
txtAddID.Text = NewID
txtAddComplaint.Text = frmtxtCompare.txtComplaint.Text

'set properties for recordsource - used to find redundant category
adcCheckCategory.Refresh
adcCheckCategory.Recordset.MoveFirst
FirstRec = adcCheckCategory.Recordset.Bookmark

End Sub
```

---

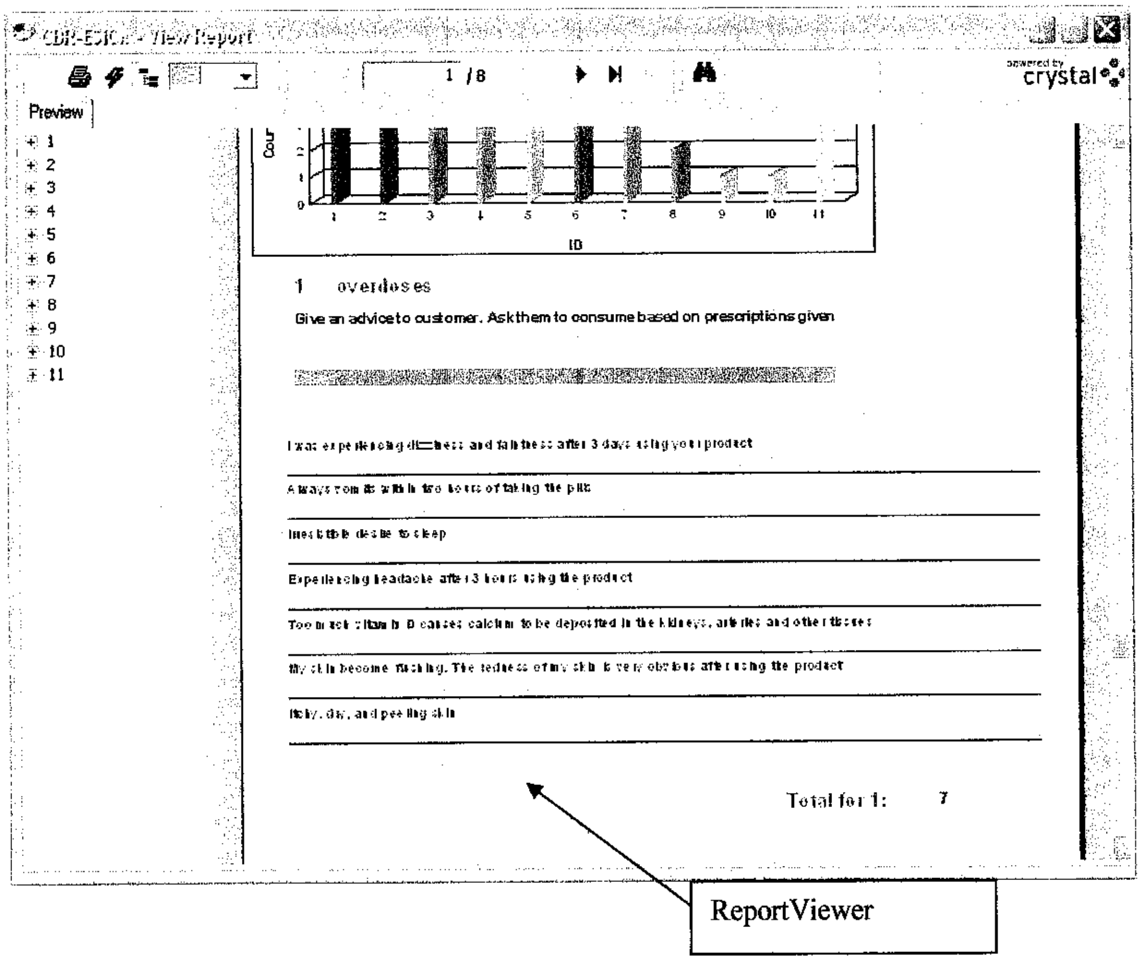
frmProcessing:



```
Private Sub Timer1_Timer()

'set status of processing

ProgressBar1.Value = ProgressBar1.Value + 5

    If ProgressBar1.Value <= 30 Then

       Label1.Caption = "Initialization of application ....."

    ElseIf ProgressBar1.Value <= 50 Then

       Label1.Caption = "Getting information ....."

    ElseIf ProgressBar1.Value <= 70 Then

       Label1.Caption = "Integration with database...."

    ElseIf ProgressBar1.Value <= 100 Then

       Label1.Caption = "Please wait ...."

    End If

    If ProgressBar1.Value = 100 Then

       frmReportViewer.Show

       Unload Me

    End If

End Sub
```

---

frmReportViewer:



1 overdoses

Give an advice to customer. Ask them to consume based on prescriptions given

I was experiencing dizziness and faintness after 3 days using your product

Always vomits within two hours of taking the pills

Irresistible desire to sleep

Experiencing headache after 3 hours using the product

Too much vitamin D causes calcium to be deposited in the kidneys, arteries and other tissues

My skin become flushing. The redness of my skin is very obvious after using the product

Itchy, dry, and peeling skin

Total for 1: 7

```
Private Sub Form_Resize()

'set report height and width
On Error Resume Next
ReportViewer.Height = DisplayPanel.Height - 180
ReportViewer.Width = DisplayPanel.Width - 135

End Sub
```

```
Private Sub Form_Load()

Dim Appn As CRAXDRT.Application
Dim cReport As CRAXDRT.Report
Dim ReportFile As String
ReportFile = App.Path & "\Report_CBR-ES.rpt"

'open the crystal report
Set Appn = CreateObject("CrystalRunTime.Application")
Set cReport = Appn.OpenReport(ReportFile)
ReportViewer.ReportSource = cReport
ReportViewer.ViewReport

End Sub
```
