MEDICAL DATA RECORDING AND SUPPORT SYSTEM ACQUISITION

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree

of

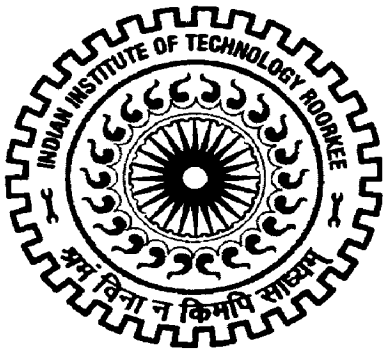
MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

By

ANIL KUMAR



ER & DCI NOIDA

IIT Roorkee – ER&DCI, Noida
C-56/1, "Anusandhan Bhawan"
Sector 62, Noida - 201 307

FEBRUARY, 2003

Enrollment No : 019006

621380285
ANI

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this dissertation titled **"MEDICAL DATA RECORDING AND SUPPORT SYSTEM ACQUITION"**, in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Information Technology,** submitted in **IIT, Roorkee – ER&DCI Campus, Noida,** is an authentic record of my own work carried out during the period from August 2002 to February, 2003 under the guidance of **Mr. Anupam Singh**, Project Manager(Software Design & Development), IMPAQ Systems, Noida.

The matter embodied in this dissertation has not been submitted by me for award of any other degree of diploma

Date: 25.02.03
Place: Noida

(Anil Kumar)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Co-Guide:

(Mr. Munish Kumar)
Project Engineer
ER&DCI- NOIDA

Date: 25-02-03
Place: Noida

Guide:

(Mr. Anupam Singh)
Project Manager
(Software Design & Development)
IMPAQ Systems, NOIDA
Date: 25-2-03
Place: Noida

ACKNOWLEDGEMENT

The work presented in this report would not have been completed without the guidance and support of many people. In first place, I would like to thank my guide, **Mr. Anupam Singh**, for his constant support, incredible enthusiasm and encouragement. I am highly indebted to respected **Dr. Prem Vratt**, Director, IIT Roorkee, **Sh. R.K.Verma**, Executive Director, ER&DCI, Noida, **Prof. A.K.Awasthi**, Dean PGS&R, IIT Roorkee, **Prof. R.P.Agrawal**, Course Coordinator, M.Tech.(IT), IIT Roorkee and **Mr. V.N.Shukla**, Course Coordinator, M.Tech.(IT), ER&DCI, Noida, for their kind efforts made for the betterment of my academic capability and for various resources provided to me for this thesis. I am highly indebted to respected **Dr. R.C.Joshi,** Head, Computer Centre, IIT Roorkee for his kind guidance.

My sincere thanks are due to **Dr. P.R.Gupta** for the encouragement and valuable suggestions she provided me with during the course of my work.

We are humbly thankful to **Mr. Munish Kumar,** project engineer, 'ER&DCI, NOIDA', and the entire staff of 'ER&DCI, NOIDA' for providing us with the necessary support to blossom.

Most of all I would like to thank my family. My wife **Preeti** and my children, **Abhinav & Shivangi** provided me a perfect environment for my studies and supported me throughout. Finally, I would like to extend my gratitude to all those persons who directly or indirectly helped me in the process and contributed towards this work..

(Anil Kumar)

Enrollment No. **019006**

CONTENTS

ABSTRACT

In the recent years, a lot of work has been done in the field of Information technology for Health care. A complete digital representation of human body has been made which is a huge database (known as **visible human database**) containing images. A powerful workstation, visualizing software and super-computer are used to create animation, walkthroughs and simulation of the human body. By using this digital representation, one can understand diseases or practice complex surgeries before doing them live. One can simulate the reactions of a human body to external stimuli like a vehicle crash.

Likewise various software were developed for image processing in medical systems, dentistry, homeopathy, simulated surgery etc. A new field is also emerging, known as Telemedicine which relies heavily on computing and telecommunication to deliver expert help to far away places.

Therefore, by looking at the need of such softwares, this dissertation discusses firstly, the need of software that can be used in the medical field for data recording and support system acquisition. Then, a research has been made to find the existing software available in various hospitals of India. A prototype has been developed by giving a user friendly common interface for all the activities. The major activities includes recording of all types of medical data i.e. text, image, video etc from any standard source in any required format. Finally a module has been made to take reports of any output in a desired presentation manner.

The development environment uses Visual Basic 6 enterprises edition to exploit the features of Microsoft Windows operating system.

INTRODUCTION

1.1 Objective

This dissertation project aims for developing a medical data recording system with the provisions of medical support device interface acquisition system.

1.2 Scope

It provides a fully automated environment for a doctor of any discipline where he/she can acquire medical data from any standard source in any required format. Reports of any output can be taken in a desired presentation manner and can be shared on a distributed network..

This software can be used in a new & fast emerging field of Health & Medicine i.e. Telemedicine. Telemedicine is delivering medical diagnosis and treatment over long distances, either in real time or store & forward modes. Here this software can be used to capture & store still or moving digital images, audio and text.

1.3 Problem Definition and Solution

In medical field, there are doctors of various disciplines. They require various kind of information about their patients. Moreover, every doctor has its own style of recording that information. Therefore a general purpose solution is required for accessing, organizing and working with medical data that can satisfy individual specific doctors need.

Patient record basically contains many kind of data such as text, images of various types & video clippings streaming from various sources. The numbers of field are not fixed. It may contain one or two pages of text data or a large amount of ECG, X-Ray, Operation report, Prescriptions or other type of reports. Moreover each type of information can be of variable size. It is inefficient and difficult to use conventional database systems which are used for fixed no. of fields & fixed length fields.

Hence, we are going to design our own data-structure for such type of applications. Every patient is considered as a separate entity and a schema is required to store the information about him. A doctor may use a existing schema or can create a new one within small period. At a later stage schema can be modified as and when required.

Similarly, for data acquisition, there are various equipments & devices from where doctors want to gather information. It can be in the form of text, image or video clipping. The proper research is to be conducted to find out all the standard interface methods and device drivers which are supposed to be used for creating interface for capturing such information.

Finally, for reports various formats are required to represent the gathered information in a desired manner. Therefore features are required to create formats, modify them and rearrange them according to the need. These reports can be burned on a compact disc for future reference or can be send on a network for further consultation with other doctor.

1.4 Organisation of Dissertation

The dissertation is further organized as follows:

Literature Survey for need of this software

A research work has been carried out to analyze the use of Information technology in medical field. Various applications of IT in the field of Health Care were also discussed. Then the present scenario in Indian context was told. Finally the existing softwares were analyzed.

System Design

It will discuss the various design issues concerned with this dissertation. Further all the modules will be discussed in detail.

Implementation

It will discuss the various user interfaces along with the implementation of various modules using various technologies.

Result and Discussions

Conclusion

LITRATURE SURVEY

2.1 Creating the Digital Human

The first visible human was created by the US National Library of Medicine; or rather the first visible human database was created there, with work starting in 1986. The visible human database is created in a rather macabre fashion. The dead bodies (cadavers) were thoroughly CT & MRI scanned and also X-rayed. They were then encased in blue Gelatin, and frozen. They were then physically sliced. The male cadaver was sliced into one millimeter thick slices while the female was sliced into slices a third of a millimeter thick. Each slice was photographed. The images thus created represent a huge database, which is not of much use as is. It is here that powerful workstations, visualizing software and supercomputers step in. Many offshoots of this project have successfully created animations, walkthroughs, and simulations of the human body.

By using this digital representation, one can understand diseases or practice complex surgeries before doing them live. One can simulate the reactions of a human body to external stimuli like a vehicle crash.

2.2 Applications of IT in HealthCare

2.2.1 Medical Imaging[1]

Imaging systems, particularly internal-imaging systems, play a key role in modern medicine. Once an image is acquired, it can be used to diagnose or decide the nature of intervention required. For example, an X-ray of a broken limb can help determine the extent and nature of bone injury (diagnosis) while an MRI of a tumor can help decide the details of the intervention (surgery) required, like where and how deep.

One can acquire such images of human internals using X-rays, radio frequencies or magnetic resonance (MR) depending on the organ being investigated and what the end use of the image is to be.

Once an image has been acquired using a scanner or camera, the rest of the work on it can be done using software, at a viewing station.

Standards in medical imaging

All software currently being written for medical-imaging systems has to conform to the DICOM (Digital Imaging in Communication in Medicine) standards to ensure that different systems from different vendors can successfully share information. So, one can, for example, acquire the image from a Siemens, and do the processing on a Philips viewing station.

2.2.2 Dentistry

Filling of carious teeth, capping or crowning of teeth comes under restorative dentistry requires several visits by the patient, who must endure extensive drilling, the rubbery impression material in his mouth, a fragile temporary restoration, and a two to four week wait for the laboratory to fabricate the crown.

CEREC (Chair side Economical Restoration of Esthetic Ceramics), a computerized dental-restorative system, takes as less as an hour. It allows the dentist to quickly restore damaged teeth with natural-colored ceramic (porcelain) fillings, saving patients time and inconvenience.

CEREC uses CAD/CAM (Computer Aided Design/Computer Aided Manufacturing) technology, incorporating an intra-oral digital camera, computer and milling machine in one instrument. The dentist uses this special camera to take an accurate picture of the damaged tooth. This optical impression is transferred and displayed on a color computer screen, where the dentist uses CAD technology to design the restoration. Then CAM takes over and automatically creates the restoration while the patient waits. Finally, the dentist bonds the new restoration to the surface of the old tooth. The whole process takes about one hour.

2.2.3 Homoeopathy

Homoeopathy treats a disease patently different from the way other systems of medicine, particularly allopathy (modern medicine), do. The effect-to-cause relationship used in homoeopathy may seem ideally suited to computerization, but the amount of software on the topic is limited and often beyond the financial reach of the average practitioner.

2.2.4 Simulated Surgery

Surgical simulation simulates the working environment of a surgical procedure using a computer and puts doctors in the perspective of the camera that is at the point of operation.

The first major shift from 'hand-eye connects' medical procedures was with the advent of tiny cameras and instruments that could be inserted into a patient's tract and monitored closely on a video monitor. Modern day endoscopies surgery relies on these tools to perform surgeries faster and with minimal risks. This is called MIS (Minimally Invasive Surgery). Doctors, however, found it a little difficult to adjust to them; hence, the surgical simulation. This concept is not restricted only to endoscopic procedures.

2.2.5 Telemedicine

Telemedicine is delivering medical diagnosis and treatment over long distances, either in real time (live) or store and forward modes. In real time, a patient consults the doctor over a teleconferencing system, while the patient's data is collected and transmitted to the consulting doctor. The data transmitted includes the patient's medical history. In store and forward, clinical data is collected, stored and later forwarded for interpretation. So, the patient and doctor are not required to be available at the same time. Computing systems used here have to capture and store still or moving digital images, audio and text.

Such techniques requires a secure (encrypted) telecommunications network (high bandwidth ISDN or satellite connection), network-conferencing systems, codecs(to compress very large, moving digital images), storage devices and database-management software all come into play to create a stable and recurring telemedicine system.

Most telemedicine units are PC based, capable of transferring digital images over an ISDN-based WAN or LAN. High-quality codecs and audio systems are important as there can be a lag between video and audio synchronization.

There are also room-based telemedicine systems with one or two large screens. They may also have white-boarding features and scanner, printer and VCR interfaces.

In India, telemedicine has great scope, particularly given that the experts are in cities, where patients from far away towns and villages will have difficulty in accessing them.

2.3 Scene in India

2.3.1 Virtual Super-specialists

Promoted by Ranbaxy, Fortis Healthcare envisages specialty mother hospitals (heart, or tho, cancer, etc) linked among themselves and to a chain of medical centers.

Using PACS (Picture Archiving and Communication System) doctors at the mother hospital will be able to diagnose and treat patients who come to the medical centers. The doctor can see in real-time parameters like ECG and Echo tests on special monitors from Agilent. He can then diagnose and treat the patient, without asking him to come to the mother hospital, which might be a few hundred km away.

As of now, the first hospital has been started at Mohali in Chandigarh (in 2001) and the first few medical centers will be up in Delhi and in Punjab by the end of this year. IT elements at Mohali include a 1 Gb/sec high-speed network, 750 data points and more that 20 compaq servers and 200 desktops running Microsoft software. It also features a single-window registration process thanks to integrated hospital-, lab-and inventory-management systems. This means that a patient can get his appointment, make payments for consultations, tests and medicines without being sent to different counters.

2.3.2 At Escorts Heart Institute and Research Centre[2] :

At Escorts Heart Institute and Research Centre, Delhi, the minute a patient is admitted, his details are entered into a database (Oracle 8i running on Win NT 4.0 with SP6). From here, M-pulse, a three-tier application developed by CellNext Solutions, takes over. It assigns a specialist from the available doctors and sends him an SMS about the details (name, bed and ward number) of the patient.

The condition of the patient is monitored 24×7 by nurses who update the database. Should the patient's condition worsen, M-pulse alerts the doctor through another

SMS. M-Pulse can also SMS other mobile numbers, say those of relatives of the patient, but only if the doctor enables this functionality.

Another feature is the availability of medical records at www.ehirc.com. Patients are given two unique id numbers at the time of registration. One is the patient specific id that remains the same for all subsequent visits, too. The other is unique to the incidence of stay at the hospital and changes each time the patient visits the hospital. The patient can use these ids to log into www.ehirc.com and retrieve his medical records.

2.3.3 At Indraprastha Apollo[3]

The Indraprastha Apollo hospital, Delhi, has automated most of its hospital-management processes. Their network consists of about 250 nodes on a 100 Mbps switched network. They use a Hospital Information system from Wipro Healthcare, which records information from the time a patient is registered at the hospital to when he's discharged. It also maintains medicine inventory, blood donation records, etc.

The radiology department uses RadWorks, an imaging software, to record all scans (MRI, CT scan, etc) so that they can be viewed from any node. In the oncology department, the hospital uses robotic arms for operations.

They offer telemedicine services to 16 remote areas in the country over a video-conferencing setup on 384 kbps ISDN links. The Apollo hospitals at Chennai and Hyderabad are also connected to this. They are now considering a shift from ISDN to satellite links.

A more recent implementation in the hospital has been an intranet, and there are plans to introduce a document-management system. www.apollolife.com is run by the online healthcare division.

2.3.4 At Apollo Heartline

Apollo has introduced a pocket-sized, mobile ECG transmitter, which transmits a patient's ECG to the Apollo Heart Command Centre. At the symptoms of a chest pain, the patient attaches the leads of the transmitter on his chest and calls up any Apollo Heart Command Centre in the country. The ECG signals are transmitted over the phone to the computer at the command Centre, which records the ECG in just 55

seconds. The specialist at the Centre compares the ECG with the patient's baseline ECG, which is stored in a database and gives immediate advice (in case of a heart attack a mobile Coronary Care Unit is arranged).

A second opinion from the patient's treating doctor is taken by sending him a fax of the ECG. This service is provided free of cost to all coronary bypass surgery and angioplasty patients who have undergone treatment in Apollo Hospitals for a period of three months.

2.3.5 At AIIMS[4]

The All India Institute of Medical Science has an 850-node network with a Gigabit Ethernet Fiber channel backbone. A 2 Mbps link to VSNL provides Internet connectivity.

AIIMS develops its own software for research, as well as uses third-party software. One of the software developed at AIIMS was based on the Bremmermann optimization technique. This is meant to reduce the energy of molecules and help predict the kind of shape a molecule will adopt. It was developed using Fortan-77. An example of third-party software is Sybyl by Tripos, a molecular analysis and design software that runs of SGI workstation.

It has signed a deal with Siemens Information Systems for a 22 month project to implement a complete hospital-management system. Plans to offer telemedicine service are also on the way. Satellite links to various hospitals in Lucknow and Chandigarh have already been established.

2.4 Existing Software

Medical Imaging with Syngo[5]

The medical imaging software from Siemens Medical systems is called **Syngo**. What's special about the software is its common user interface for all the medical imaging applications CT, MRI etc. This has been designed to minimize eye movement and mouse clicks, so that the software is easy to use and improves productivity. It has been built on Win NT/2000 Platform.

The software can integrate all processes of medical imaging, from data acquisition and processing to archiving. It starts from patient registration, and can also import

DICOM worklists. Image acquisition can be done specific to the scan type. For instance, in a CT scan, it provides a large number of protocols for each body section and the optimized parameters for each.

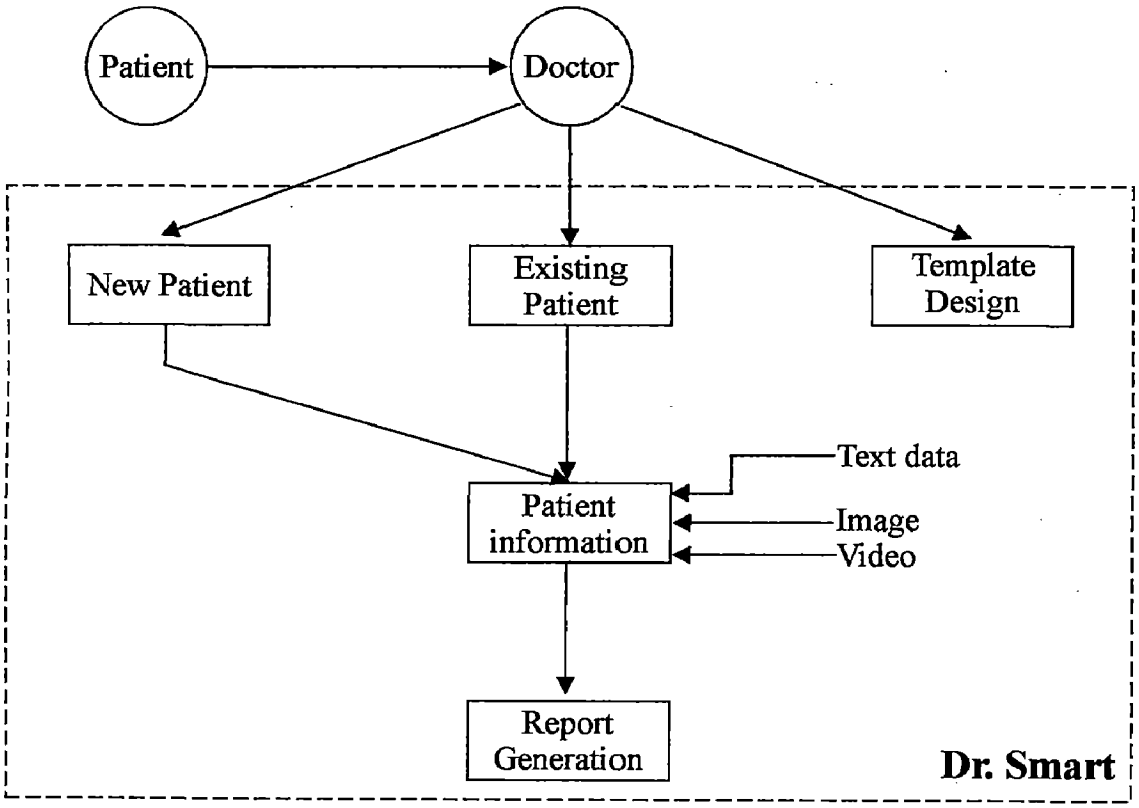
Syngo's functionality can be enhanced by adding different task cards. There are the 3D and advanced 3D imaging task cards, which can be used for multi-planner reconstruction, maximum intensity projection, and shaded surface display. It can be used to colorize grayscale Image scans by doing a simple drag-n-drop. Then there are task cards for computed tomography, which include Osteo CT, a noninvasive method for measuring bone mineral density, a HeartView CT for imaging of the heart and surrounding structures.

Such advanced processes are only possible due to the hardware. The latest CT-scan machine from Siemens has Intel P4 processors, and a graphics engine for image processing. Siemens also has a program (Evolve) that upgrades machine hardware every 3 years.

SYSTEM DESIGN

3.1 Design Structure

The whole system can be shown as follows



(Fig 3.1: Complete view)

3.1.1 Splash Screen:

It contains the introductory information about the software along with necessary copyright information.

3.1.2 Selection Screen:

After Splash screen, this is the first screen that will appear. It provides the three major selection operations. They are

Existing :	To select existing patient
New :	To enter information about new patient
Template:	To go to template design screen.

3.1.3 Existing Patient:

The list of all the existing patients in the system will appear.

The user can select one of the patient based on either

- Patient Identification Number (ID)
- Patient Name

There will be provisions for fast selection.

After selecting the patient, the user will switch to main working screen which contains all the information related to the patient.

3.1.4 New Patient:

This screen will allow user to create a storage structure for a new patient. The first step is to associate the patient with a template for storing information.

Firstly, enter the patient name. System will automatically assign Patient ID to that patient.

Then, assign the desired template by either way:

- Use default template.
- Use other template whose preview will be available.

There will be provisions for previewing the template & setting any template as default for storing information.

After associating the template with the patient, the user will switch to main working screen which contains all the information related to the patient.

3.1.5 Template Design:

Template is a basic structure designed to store information of a patient in a desired manner. An orthopedic doctor may require different format to store information than a general physician. Also a small clinic may require a different format than a large Nursing home or hospital. Every patient may require a different

format and different volume of data. Therefore, there is a provision for designing template for every type of patient or group of patients.

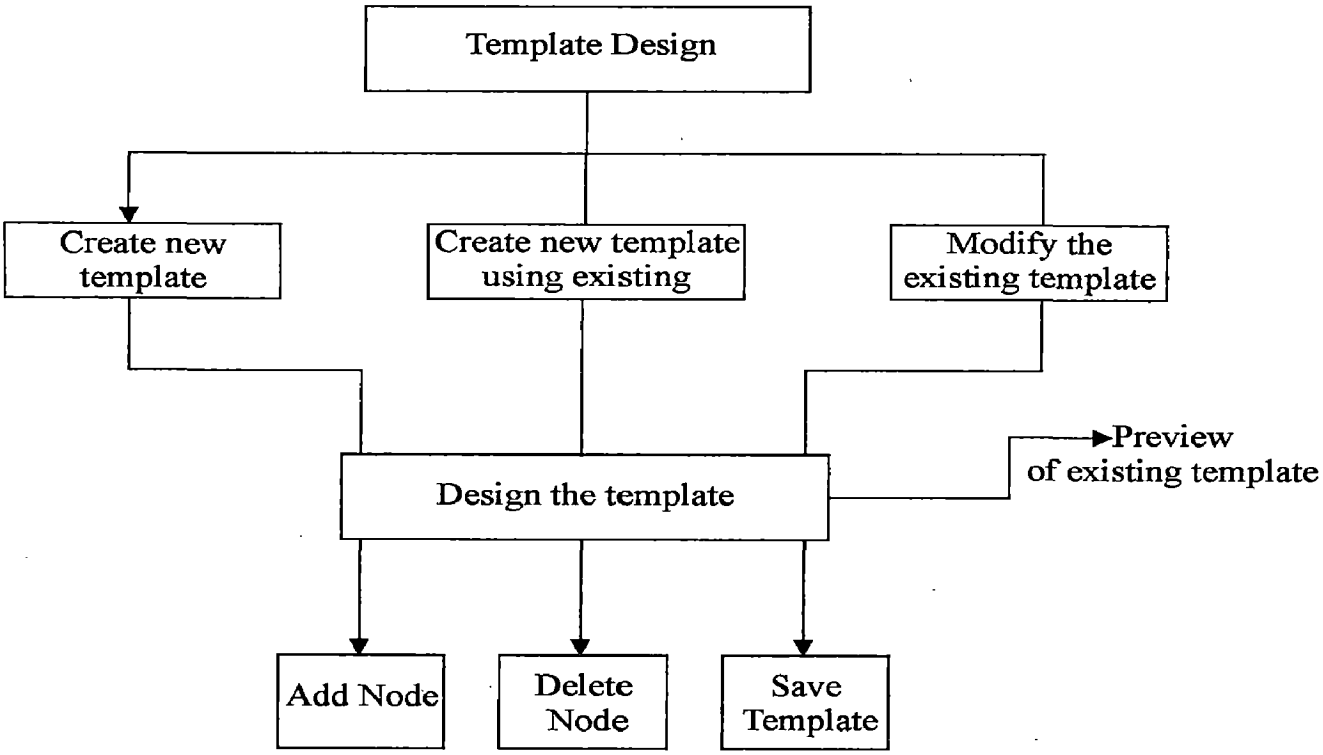
There are three ways of doing it:

(a) Creating a new template from scratch

(b) Creating a new template by using existing one

(c) Modify an existing template.

After selecting one of the options, one will get a template designing screen. Here we have provisions to Add or delete a node of information in a hierarchical structure. Finally, one can save the template.



(Fig 3.2: Template Design)

Adding Node: For adding a node, just follow three steps:

- Enter the name and description of the node.
- Select the type of node depending on the type of information required. It may be a text, image, text with image, video etc.
- Define the labels required for that particular node.

Delete Node: Select the node to be deleted and delete it by pressing a delete. All the information related to that particular node will be automatically removed from the structure.

Save template: Saving of template can be achieved by saving. If the template so far not provided with a name under which it is supposed to be saved then the prompt will be given to assign a new name for the template.

3.1.6 Patient Information

This interface is the main screen (or working screen, fig 3.3) for data acquisition from various sources such as keyboard, graphire tool, scanner, VCD, video camera etc.

The main goal of this module will be to efficiently arrange the information so that it can be easily accessed and utilized for diagnosis purposes. Also there must be provisions for storing prescriptions, recommendations and medical advices of various doctors.

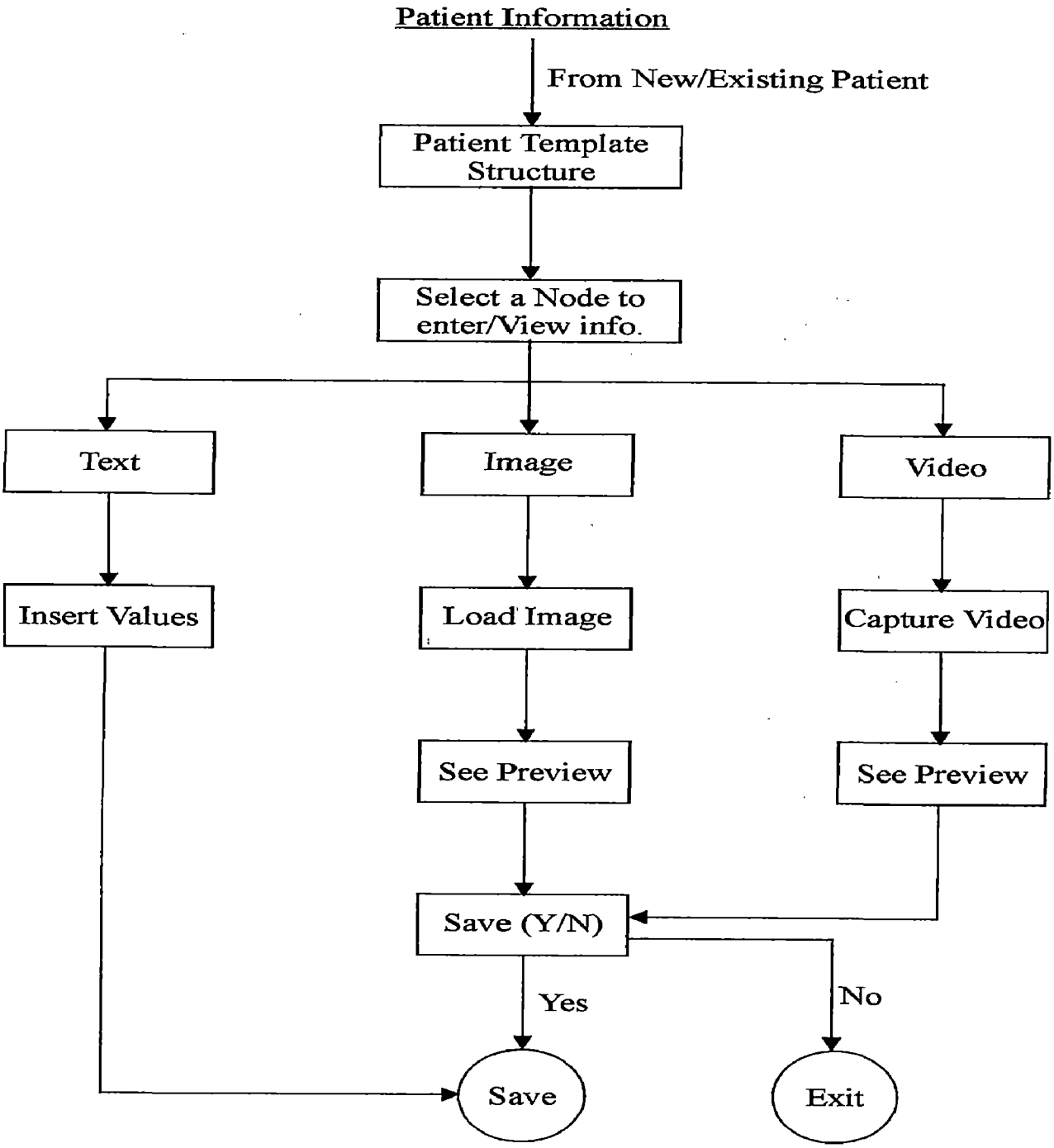
The Left hand side of the screen will display the structure of the patient depending on the template.

The Right hand side of the screen will display the information contained in the selected node of information.

Now, user can enter information depending on the type of node in the template.

Text Information:	From Keyboard & Graphire tool
Image:	From Scanner or video capturing
Video:	From video Camera/VCD

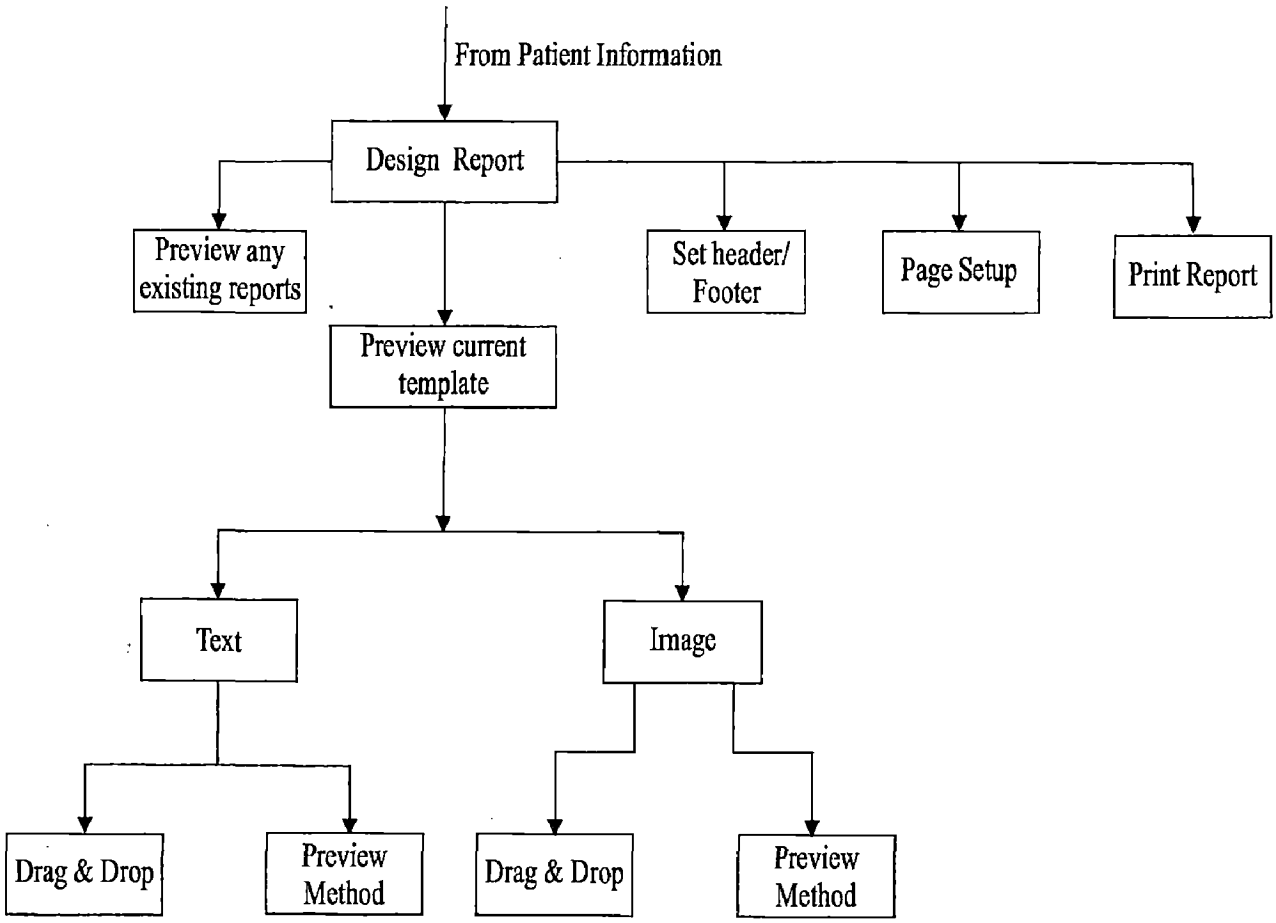
All these information will be stored in the individual patient file.



(Fig 3.3: Patient Information)

3.1.7 Report Generation :

In this screen, a report can be designed by the user. A very simple interface will provide drag and drop facility as well as preview facility to generate a report for any patient. Any printer attached to it will print the designed report after setting up a page setup interface. There should be provisions for setting header & footer for a page.



(Fig 3.4: Report Generation)

3.2 Design Issues

Following are the major design issues:

(a) Creation of User Interface

(b) Storage of Patient Information

(c) Collection of information from various sources

(d) Presentation of Information in a desired format

(e) Format Change and Compression of images (using Intel JPEG Library)

3.2.1 Creation of User Interface

The following major controls are used for creating User Interface:

- Picture Box Control
- MSFlexGrid Control
- SSTab
- ListView
- TreeView
- ImageList

3.2.1.1 Picture Box[6]

The primary use for the picture box control is to display a picture to the user. The actual picture that is displayed is determined by the Picture property. The Picture property contains the file name (and optional path) for the picture file that one wish to display.

To display or replace a picture at run time, use the LoadPicture function to set the Picture property. Supply the name (and optional path) for the picture and the LoadPicture function handles the details of loading and displaying it:

```
picMain.Picture = LoadPicture("splash.BMP")
```

The picture box control has an AutoSize property that, when set to True, causes the picture box to resize automatically to match the dimensions of its contents.

Picture Box as a Container

The picture box control can also be used as a container for other controls. Like the frame control, one can draw other controls on top of the picture box. The contained controls move with the picture box and their Top and Left properties will be relative to the picture box rather than the form.

Other Uses for the Picture Box

The picture box control has several methods that make it useful for other purposes. The picture box can be used as a blank canvas upon which one can paint, draw or print. A single control can be used to display text, graphics or even simple animation.

The Print method allows the user to output text to the picture box control just as one would to a printer. Several font properties are available to control the characteristics of text output by the Print method; the Cls method can be used to erase the output.

Circle, Line, Point and Pset methods may be used to draw graphics on the picture box. Properties such as DrawWidth, FillColor, and FillStyle allow the user to customize the appearance of the graphics.

Animation can be created using the PaintPicture method by moving images within the picture control and rapidly changing between several different images.

The picture box control is similar to the image control in that each can be used to display graphics in your application — each supports the same graphic formats. The picture box control, however, contains functionality which the image control does not, for example: the ability to act as a container for other controls and support for graphics methods.

Supported Graphic Formats

The picture box control can display picture files in any of the following formats: bitmap, cursor, icon, metafile, enhanced metafile, or as JPEG or GIF files.

3.2.1.2 MSFlexGrid Control[6]

The Microsoft FlexGrid (MSFlexGrid) control displays and operates on tabular data. It allows complete flexibility to sort, merge, and format tables containing strings and pictures. When bound to a Data control, MSFlexGrid displays read-only data.

The user can put text, a picture, or both, in any cell of an MSFlexGrid. The Row and Col properties specify the current cell in an MSFlexGrid. One can specify the current cell in code, or the user can change it at run time using the mouse or the arrow keys. The Text property references the contents of the current cell.

If the text in a cell is too long to display in the cell, and the WordWrap property is set to True, the text wraps to the next line within the same cell. To display the wrapped text, one may need to increase the cell's column width (ColWidth property) or row height (RowHeight property).

Use the Col and Row properties to determine the number of columns and rows in an MSFlexGrid.

When distributing an application, install the MSFlxGrd.ocx file in the user's Microsoft Windows System directory.

3.2.1.3 SSTab Control[6]

The SSTab control provides a group of tabs, each of which acts as a container for other controls. Only one tab is active in the control at a time, displaying the controls it contains to the user while hiding the controls in the other tabs.

An SSTab control is like the dividers in a notebook or the labels on a group of file folders. Using an SSTab control, user can define multiple pages for the same area of a window or dialog box in his application. Using the properties of this control, one can:

- Determine the number of tabs.
- Organize the tabs into more than one row.
- Set the text for each tab.
- Display a graphic on each tab.

- Determine the style of tabs used.
- Set the size of each tab.

To use this control, user must first decide how he want to organize the controls that he will place into various tabs. Set the Tabs and TabsPerRow properties to create the tabs and organize them into rows. Then select each tab at design time by clicking the tab. For each tab, draw the controls you want displayed when the user selects that tab. Set the Caption, Picture, TabHeight, and TabMaxWidth properties as needed to customize the top part of the tab.

At run time, users can navigate between tabs by either pressing CTRL+TAB or by using accelerator keys defined in the caption of each tab.

One can also customize the entire SSTab control using the Style, ShowFocusRect, TabOrientation, and WordWrap properties.

The SSTab control is found in the TABCTL32.OCX file. To use the SSTab control in an application, one must add the control's .OCX file to the project. When distributing an application, install the appropriate .OCX file in the user's Microsoft Windows System or System32 directory.

3.2.1.4 ListView Control[6]

The ListView control displays items using one of four different views. User can arrange items into columns with or without column headings as well as display accompanying icons and text.

With a ListView control, user can organize list entries, called ListItem objects, into one of four different views:

- Large (standard) Icons
- Small Icons
- List
- Report

The View property determines which view the control uses to display the items in the list. You can also control whether the labels associated with items in the list wrap to more than one line using the LabelWrap property. In addition, one can manage how items in the list are sorted and how selected items appear.

The ListView control contains ListItem and ColumnHeader objects. A ListItem object defines the various characteristics of items in the ListView control, such as:

- A brief description of the item.
- Icons that may appear with the item, supplied by an ImageList control.
- Additional pieces of text, called subitems, associated with a ListItem object that one can display in Report view.

User can choose to display column headings in the ListView control using the HideColumnHeaders property. They can be added at both design and run time. At design time, user can use the Column Headers tab of the ListView Control Properties dialog box. At run time, use the Add method to add a ColumnHeader object to the ColumnHeaders collection.

3.2.1.5 TreeView Control[6]

A TreeView control displays a hierarchical list of Node objects, each of which consists of a label and an optional bitmap. A TreeView is typically used to display the headings in a document, the entries in an index, the files and directories on a disk, or any other kind of information that might usefully be displayed as a hierarchy.

After creating a TreeView control, user can add, remove, arrange, and otherwise manipulate Node objects by setting properties and invoking methods. One can programmatically expand and collapse Node objects to display or hide all child nodes. Three events, the Collapse, Expand, and NodeClick event, also provide programming functionality.

User can navigate through a tree in code by retrieving a reference to Node objects using Root, Parent, Child, FirstSibling, Next, Previous, and LastSibling properties. Users can navigate through a tree using the keyboard as well. UP ARROW and DOWN ARROW keys cycle downward through all expanded Node objects. Node

objects are selected from left to right, and top to bottom. At the bottom of a tree, the selection jumps back to the top of the tree, scrolling the window if necessary.

Several styles are available which alter the appearance of the control. Node objects can appear in one of eight combinations of text, bitmaps, lines, and plus/minus signs.

The TreeView control uses the ImageList control, specified by the ImageList property, to store the bitmaps and icons that are displayed in Node objects. A TreeView control can use only one ImageList at a time. This means that every item in the TreeView control will have an equal-sized image next to it when the TreeView control's Style property is set to a style which displays images.

3.2.1.6 ImageList Control[6]

An ImageList control contains a collection of ListImage objects, each of which can be referred to by its index or key. The ImageList control is not meant to be used alone, but as a central repository to conveniently supply other controls with images.

The ImageList control functions as a storehouse for images, and as such, it needs a second control to display the stored images. The second control can be any control that can display an image's Picture object, or it can be one of the Windows Common Controls that were specifically designed to bind to the ImageList control. These include the ListView, ToolBar, TabStrip, Header, ImageCombo, and TreeView controls. In order to use an ImageList with one of these controls, you must bind a particular ImageList control with the second control through an appropriate property. For the ListView control, one must set the Icons and SmallIcons properties to ImageList controls. For the TreeView, TabStrip, ImageCombo, and Toolbar controls, you must set the ImageList property to an ImageList control.

At design time, user can add images using the Images tab of the ImageList Control Properties dialog box. At run time, he can add images using the Add method for the ListImages collection.

The ImageList, TreeView and ListView controls are part of a group of ActiveX controls that are found in the MSCOMCTL.OCX file. To use the these control in the application, user must add the MSCOMCTL.OCX file to the project. When

distributing an application, install the MSCOMCTL.OCX file in the user's Microsoft Windows System or System32 directory.

3.2.2 Storage of Patient Information:

Now a day, the conventional way of storing large amount of information is using a Relational Data Base Management System. But, we have to store information about entities which have very complex structure. The complexity lies in the type of data to be stored and the volume of it. The patient information can be a few lines of text or it can be a large volume containing different types of data i.e. text, images, videos etc. In this way the short come is that same type of database structure is been imposed for all entities which leads to make lot of unfilled fields to store in database. For traditional approach the database structure can not be designed flexible for data storage. So for that reason a flexible type of data storage structure is needed to fulfill all type of requirement.

Therefore, we will store this information in a structure suitable for such application. It will be a hierarchical structure which consists of the following type of three file structure:

- Template Index File
- Template File
- Patient Data File

Note: All character length are in Hex

3.2.2.1 Template Index File

File:- TIndex

Extention :- **.DTI**

Path :- App.Path\System\TIndex.DRI (DrSmart Index File)

File Format:-

Date Modified (8), Date Created (8), Name Length (2), Template Name (n), Discription (n),Template File Name(8)

3.2.2.2 Template File

Name:- 8 character long. Based on hex naming e.g. 000000AF

Extention:- **.DRT**

Path:- App.Path\Templates*.DRT

Available at:- Registary DrSmart, Index, TemplateID, ...

File Format:-

For Text, Image, Movie Type Node
Node Type (1), Push (2), Length Of Node Text (2), Node Text (n),
No Of labels (2),Node Discription(n)
Length Of Label(2),Label, Label Discription
Length Of Label(2),Label, Label Discription
Length Of Label(2),Label, Label Discription

For Text Image, TextMovie Type Node
Node Type (1), Push (2), Length Of Node Text (2), Node Text (n),
No Of labels (2) ,Node Discription(n)
Length Of Label(2),Label, Label Discription
Length Of Label(2),Label, Label Discription
Length Of Label(2),Label, Label Discription

No Of Labels Pre Image/Movie
Length Of Label(2),Label, Label Discription
Length Of Label(2),Label, Label Discription
Length Of Label(2),Label, Label Discription

3.2.2.3 Patient Data File

Name:- 5 character long (First one is character and rest 4 are numbers e.g."A00A2.drp")

Extention:- ***.DRP**

Path:- App.Path\Data\PatientID\Data.DRP

Available at:- Registary DrSmart, Index, TemplateID, ...

File Format:

Structure of Text Node

NodeType(1)==Push(2)==LengthOfNodeText(3)==NodeText(n)=...
NoOfLabels(2)==
LengthOfLabel(2)==Label======LabelText
LengthOfLabel(2)==Label======LabelText

Structure of Image / Movie Node

NodeType(1)==Push(2)==LengthOfNodeText(3)===NodeText=...
NoOfLabelsPerImage(2)==TotalNoOfImagePresent(4)
LengthOflabel(3)===LabelName======LengthOflabel(3)==...
LabelName===LengthOFImageName(4)==ImageName===...
LengthOfData(3)===Data===LengthOfData(3)===Data====== ...

Structure of TextImage / TextMovie Node

NodeType(1)==Push(2)==LengthOfNodeText(2)===NodeText=...
NoOfLabels(2)===
LengthOfLabel(2)===Label======LabelText
LengthOfLabel(2)===Label======LabelText

Structure of TextImage/TextMovie Node (Only Image/Movie Part)

NoOfLabelsPerImage(2)==TotalNoOfImagePresent (4)
LengthOflabel(3)===LabelName======LengthOflabel(3)==...
LabelName===LengthOFImageName(4)===ImageName===...
LengthOfData(3)===Data====LengthOfData(3)===Data====== ...

3.2.3 Collection of information from various sources

For collecting the information under windows operating system we are using **Windows Platform SDK** to communicate with operating system for desired performance. Use of platform SDK will provide interconnection between capture hardware, operating system and our software. In this way it will be possible to tune the hardware for desired format performance like it is suppose to take data from S-Video or composite video.

The Platform SDK is organized as following groups of related technologies.

- Windows Programming
- Windows Base Services

- User Interface Services
- COM and ActiveX Object Services
- Internet, Intranets, and Extranet Services
- Networking and Distributed Services
- Database and Messaging Services
- Graphics and Multimedia Services
- Setup and Systems Management Services

Amongst these, we uses Windows Programming technologies and graphics services

The Windows Programming contains general information that will help to create, build, and manipulate Win32-based applications. The followings are used in this dissertation:

3.2.3.1 The Win32 API[7]

The Win32 API allows applications to exploit the power of the 32 bit Windows family of operating systems. The functions, structures, messages, macros, and interfaces form a consistent and uniform API for all of Microsoft's 32-bit platforms. Using the Win32 API, one can develop applications that run successfully on all platforms while still being able to take advantage of unique features and capabilities of any given platform.

The Win32 API can be grouped into these functional categories:

- Window Management
- Window Controls
- Shell Features
- Graphics Device Interface

- System Services
- International Features
- Network Services

The following functional category is used in this dissertation:

Graphics Device Interface[9]

The graphics device interface (GDI) provides functions and related structures that an application can use to generate graphical output for displays, printers, and other devices. Using GDI functions, one can draw lines, curves, closed figures, paths, text, and bitmap images. The color and style of the items one draw depends on the drawing objects — that is, pens, brushes, and fonts — that one create. One can use pens to draw lines and curves, brushes to fill the interiors of closed figures, and fonts to write text.

Applications direct output to a given device by creating a device context (DC) for the device. The device context is a GDI-managed structure containing information about the device, such as its operating modes and current selections. An application creates a DC by using device context functions. GDI returns a device context handle, which is used in subsequent calls to identify the device. For example, using the handle, an application can retrieve information about the capabilities of the device, such as its technology type (display, printer, or other device) and the dimensions and resolution of the display surface.

Applications can direct output to a physical device, such as a display or printer, or to a "logical" device, such as a memory device or metafile. Logical devices give applications the means to store output in a form that is easy to send subsequently to a physical device. Once an application records output in a metafile, it can play that metafile any number of times, sending the output to any number of physical devices.

Applications use attribute functions to set the operating modes and current selections for the device. The operating modes include the text and background colors, the mixing mode (also called the binary raster operation) that specifies how colors in a pen or brush combine with colors already on the display surface, and the mapping

mode that specifies how GDI maps the coordinates used by the application to the coordinate system of the device. The current selections identify which drawing objects are used when drawing output.

3.2.3.2 DirectDraw[9]

DirectDraw is a DirectX Programmer's Reference component of the Platform Software Development Kit (SDK) that allows one to directly manipulate display memory, the hardware blitter (A hardware component, built into the display adapter, that performs efficient blit operations) , hardware overlay support, and flipping surface (Any piece of memory that can be flipped) support. DirectDraw provides this functionality while maintaining compatibility with existing Microsoft Windows-based applications and device drivers.

DirectDraw is a software interface that provides direct access to display devices while maintaining compatibility with the Windows graphics device interface (GDI). It is not a high-level application programming interface (API) for graphics. DirectDraw provides a device-independent way for games and Windows subsystem software, such as 3-D graphics packages and digital video codecs, to gain access to the features of specific display devices.

DirectDraw works with a wide variety of display hardware, ranging from simple SVGA monitors to advanced hardware implementations that provide clipping, stretching, and non-RGB color format support. The interface is designed so that user's applications can enumerate the capabilities of the underlying hardware and then use any supported hardware-accelerated features. Features that are not implemented in hardware are emulated by DirectX.

DirectDraw provides device-dependent access to display memory in a device-independent way. Essentially, DirectDraw manages display memory. User's application need only recognize some basic device dependencies that are standard across hardware implementations, such as RGB and YUV color formats and the pitch between raster lines. He need not call specific procedures to use the blitter or manipulate palette registers. Using DirectDraw, one can manipulate display memory with ease, taking full advantage of the blitting and color decompression capabilities of

different types of display hardware without becoming dependent on a particular piece of hardware.

3.2.2.3 Video Capture[6]

User can easily incorporate video capture capabilities into his application by using the AVICap window class. AVICap provides applications with a simple, message-based interface to access video and waveform-audio acquisition hardware and to control the process of streaming video capture to disk.

AVICap supports streaming video capture and single-frame capture in real-time. In addition, AVICap provides control of video sources that are Media Control Interface (MCI) devices so the user can control (through an application) the start and stop positions of a video source, and augment the capture operation to include step frame capture.

The windows one create by using the AVICap window class can perform the following tasks:

- Capture audio and video streams to an audo-video interleaved (AVI) file.
- Connect and disconnect video and audio input devices dynamically.
- View a live incoming video signal by using the overlay or preview methods.
- Specify a file to use when capturing and copy the contents of the capture file to another file.
- Set the capture rate.
- Display dialog boxes that control the video source and format.
- Create, save, and load palettes.
- Copy images and palettes to the clipboard.
- Capture and save a single image as a device-independent bitmap (DIB).

Video Capture can be used to perform the following tasks:

- Creating a capture window
- Connecting to a capture driver
- Enumerating installed capture drivers
- Obtaining the capabilities of a capture driver
- Obtaining the status of a capture window
- Displaying dialog boxes to set video characteristics
- Obtaining and setting the video format
- Previewing video
- Enabling video overlay
- Naming the capture file
- Formatting audio capture
- Changing a video capture setting
- Capturing data
- Adding an information chunk
- Adding callback functions to an application
- Creating a status callback function
- Creating an error callback function
- Creating a frame callback function

3.2.4 Presentation of Information in a desired format

Now, the information stored in a new data storage structure is to be represented in a desired format. Therefore, an interface is required to design a format of a report to generate a hardcopy of the information stored in the structure.

Interface should have a facility to easily pickup the desired node of information. The best way of doing is to simply drag and drop.

There must be provisions for editing the report according to the needs of a user.

Finally user should be able to take a print out on any type of printer attached to the system.

3.2.5 Format change and Compression of images

When a image is captured, it creates a file in a BMP format which always require large amount of memory. Therefore it is necessary to change it into another format where it can be compressed to a smaller size. Here, we will convert these images into another format i.e. JPGE (Joint Photographic Experts Group is a compressed bitmap format which supports 8- and 24-bit color.). For this we may use Intel JPGE Library.

Intel JPGE Library (IJL)[10]

The IJL is a software library for application developers that provide high performance JPEG encoding and decoding of full color, and grayscale, continuous-tone still images.

The IJL was designed for use on Intel processors-based systems and has been tuned for high performance and efficient memory usage.

Additionally, the IJL was developed to take advantage of MMX technology if present. The IJL provides an easy-to-use programming interface without sacrificing low-level JPEG control to advanced developers. The IJL also includes a substantial amount of functionality that is not included in the ISO JPEG standard. This added functionality is typically necessary when working with JPEG images, and includes pre-processing and post-processing options like sampling and color space conversions.

Steps for Creating an IJL Application

1. Write a program with the IJL function calls. Use the IJL functions just as if they were defined in a program.

2. Include the IJL header file, **IJL.H**, in each source module that calls an IJL function.

3. Add the IJL import library **IJL15.LIB**, or static library **IJL15L.LIB** to a project's list of link libraries.

4. Compile and link an application as one would normally do to create a Win32 application.

The current JPEG standard (ISO DIS 10918-1) has 44 possible JPEG image compression techniques, many of which are application-specific and not used by the majority of the JPEG decoders. Similarly, the IJL supports only a subset of the possible compression techniques.

SYSTEM IMPLEMENTATION

The implementation of a system has been divided into five phases.

1. Creation of front end for data recording
2. Template Structure designing
3. Data acquisition from various sources.
4. Output in various formats.
5. Integration of various modules.

4.1 Creation of front end

The main front end screens for data recording are:

a) Splash Screen

b) Selection Screen

c) Patient Information Screen

4.1.1 Splash screen (frmSplash) :

It is implemented using Picture Box control as container which contains a image of company & various label are used to display other information. A timer is set to display this screen till the information is loaded on to the system.

4.1.2 Selection screen(frm Option Select) :

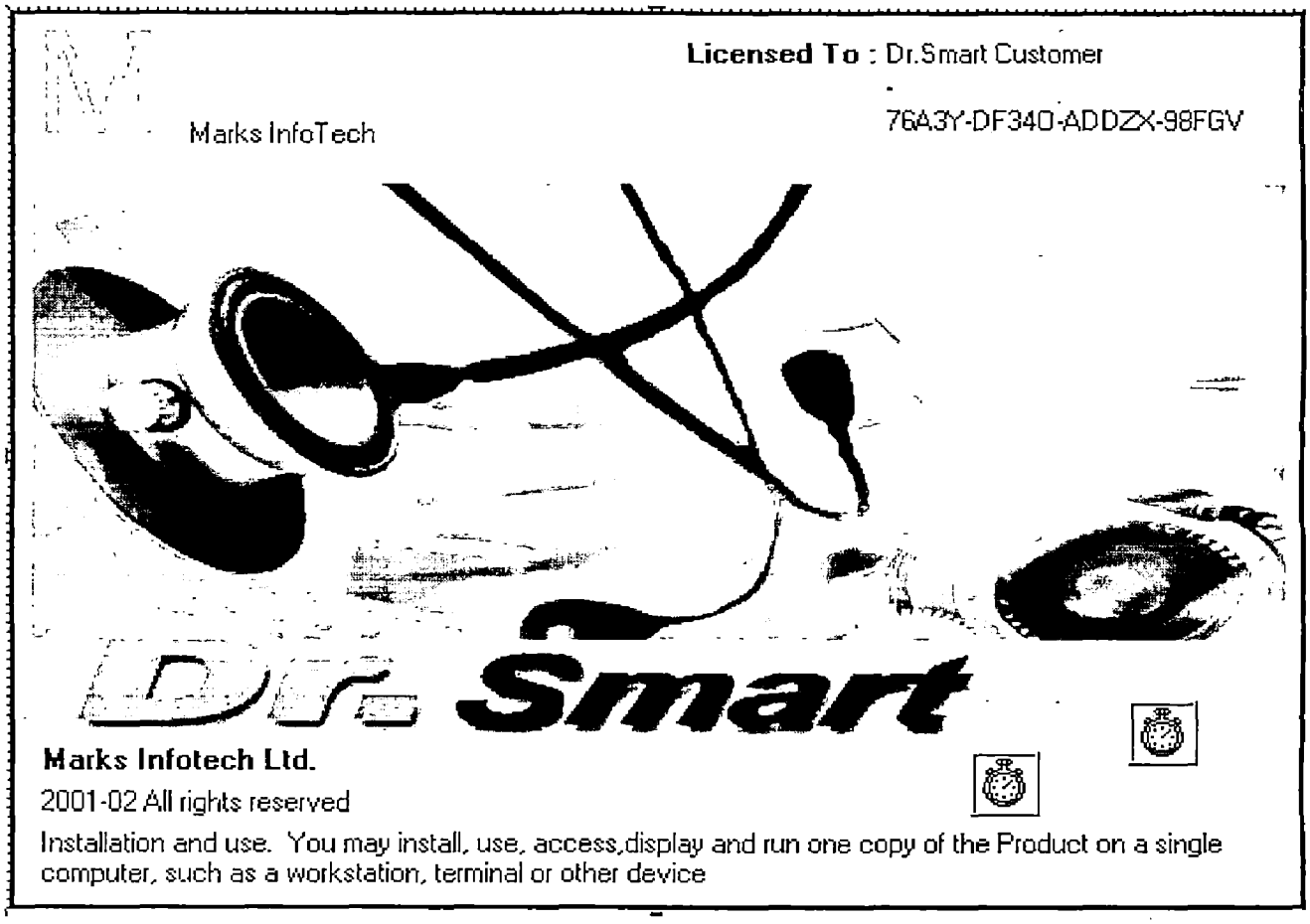
It is implemented using SSTab control where three tabs are used as a container for other controls. These tabs provide interface for three options:

Existing Patient

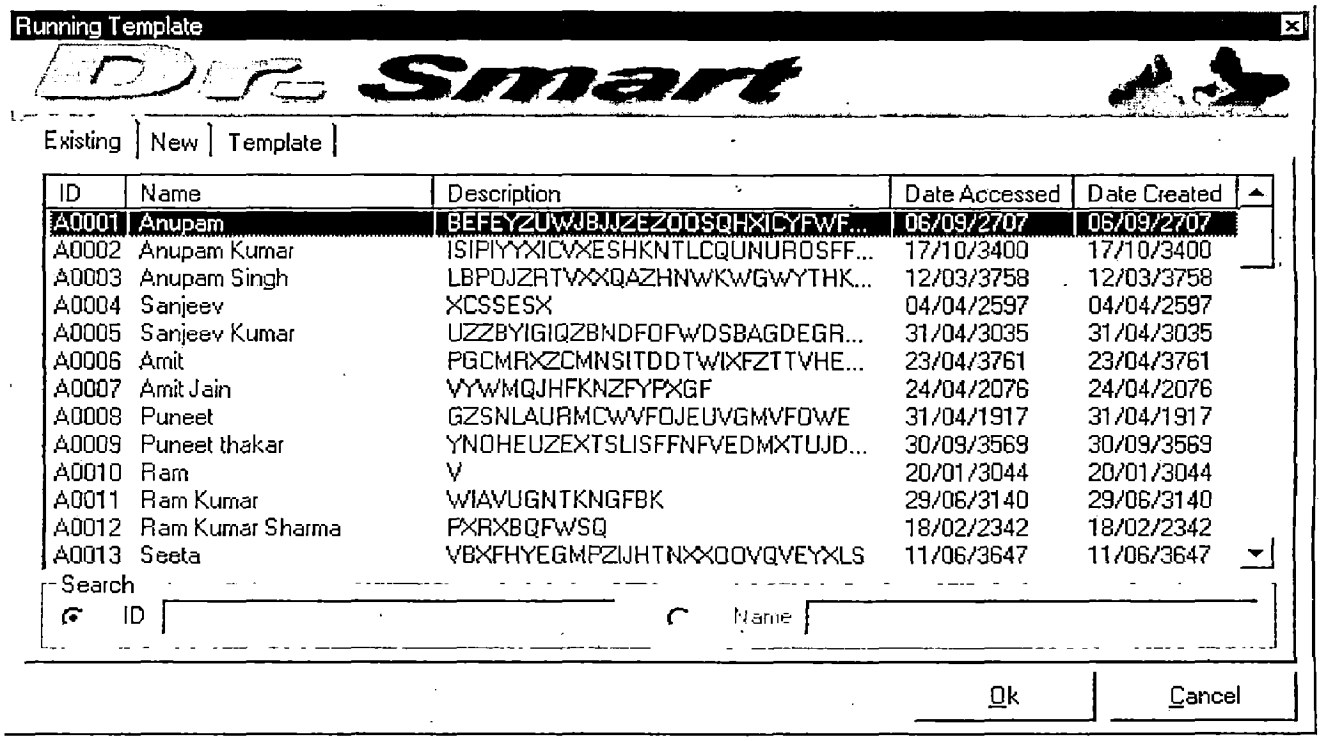
New Patient

Template

Only one option will remain active in the control at a time, displaying the controls it contains to the user.

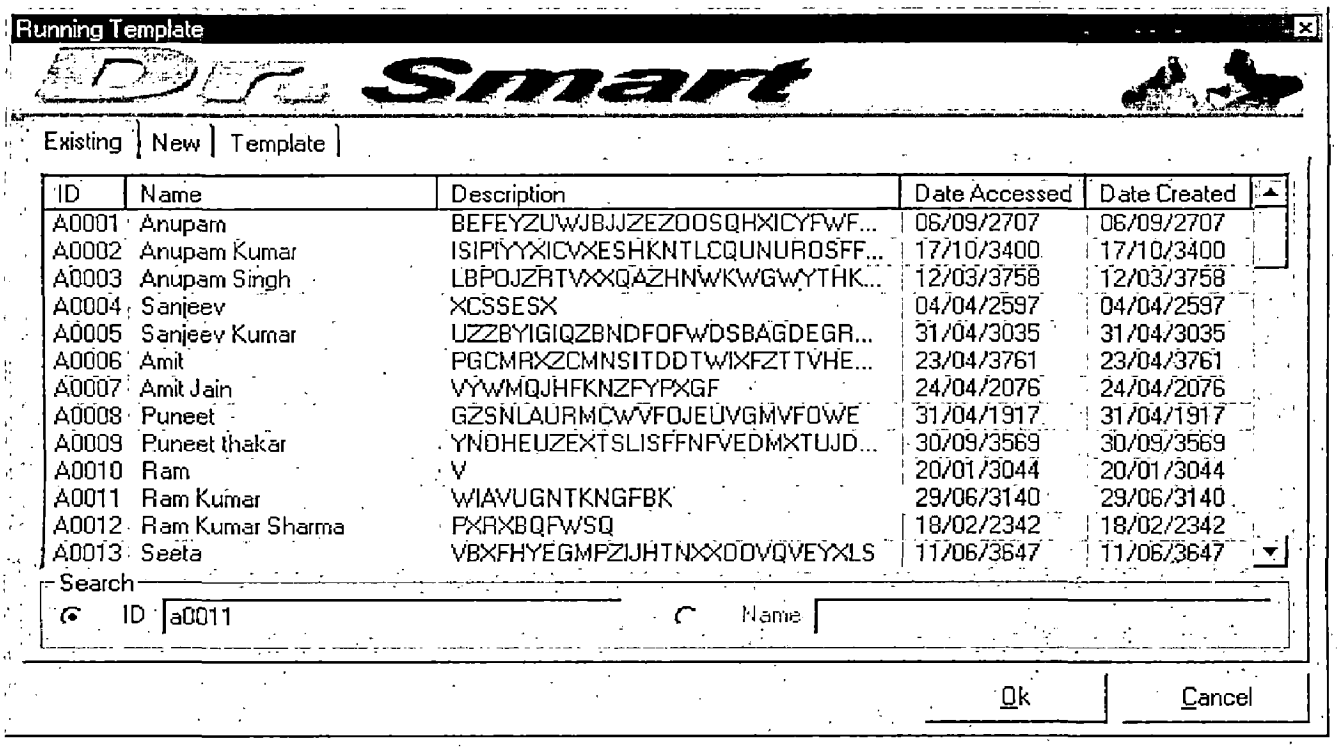


(fig 4.1 Splash Screen)



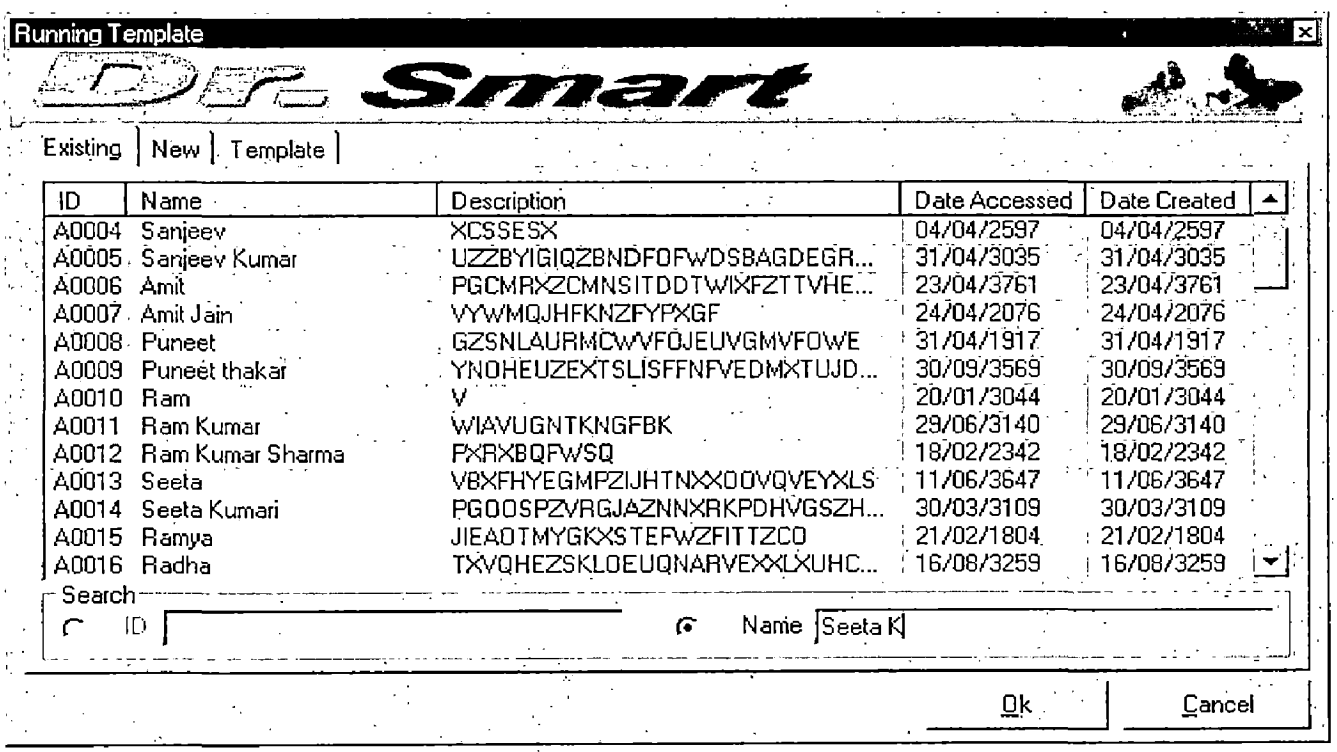
(fig 4.2 Selection Screen)

Existing tab will use List view control to display the list of existing patients. The various column headings are PatientID, Patient Name, Description, Datecreated, Dateaccessed.



(fig 4.3 Selection Screen : Existing Tab)

Using option button, user can select patient either by entering PatientID as shown above



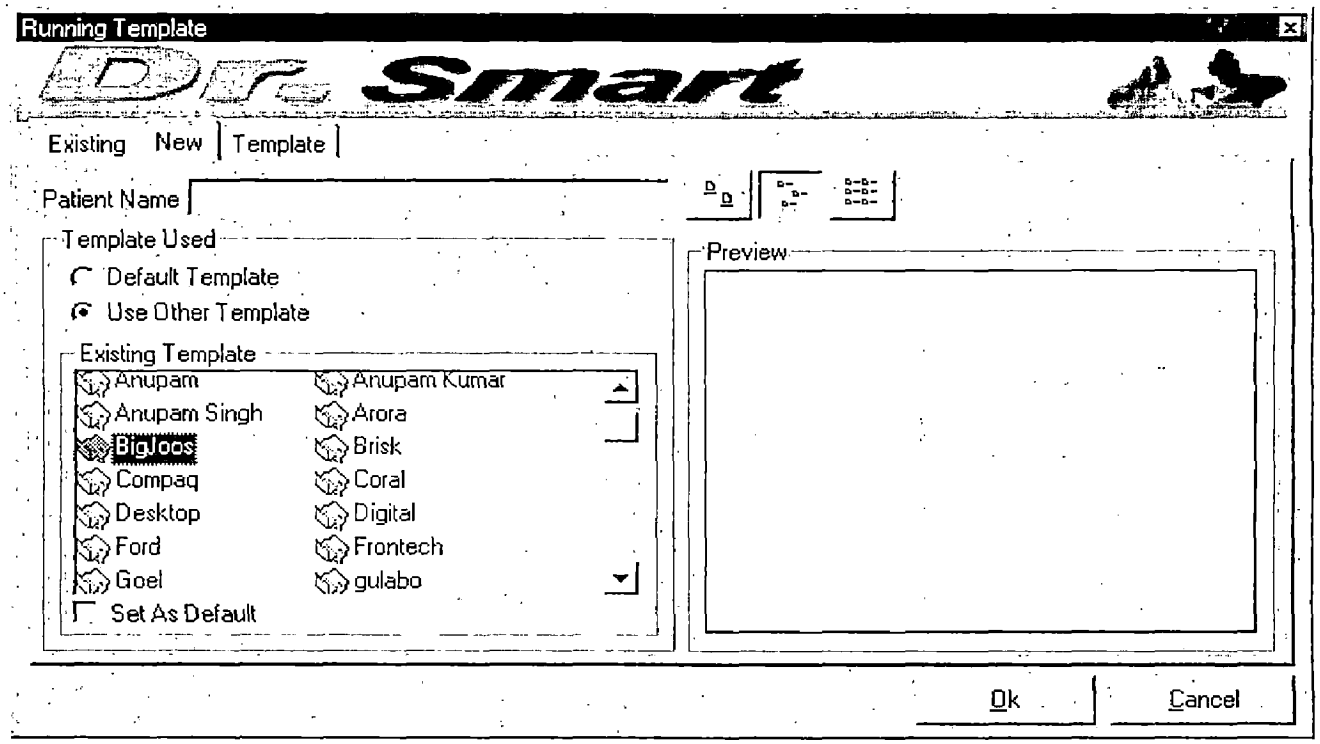
(Fig.4.4 :Selection by name)

Otherwise ,user can select patient by entering Patient Name.

The fast selection method is provided by using a patient Index File described in 3.2.2.1.

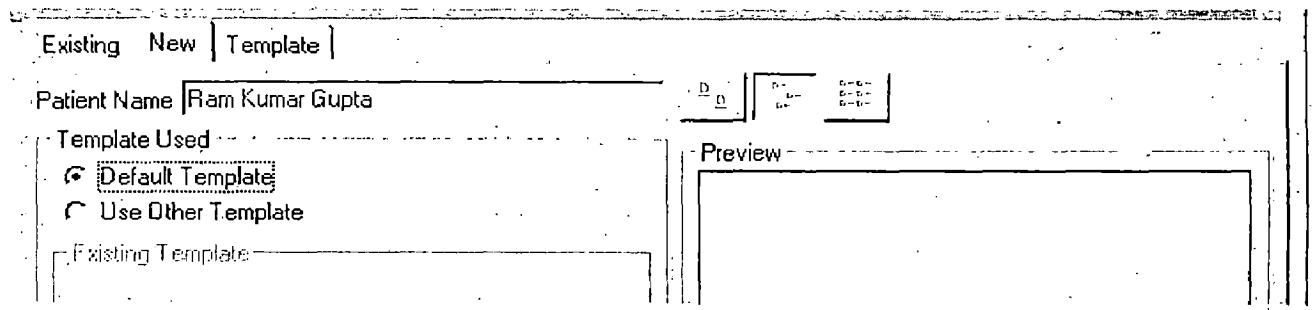
New tab will use textbox to enter the name of the new patient. Tab will display the option for the user to use the default template or select some other existing template.

List of existing template will be shown using Listview & ImageList control. For the ListView control, we had set the Icons and SmallIcons properties to ImageList control.



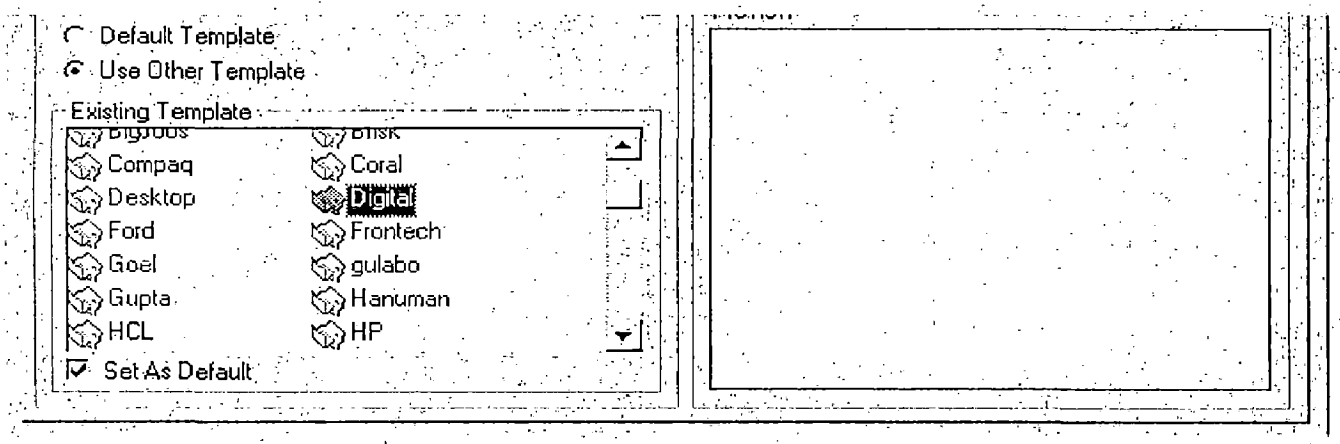
(fig 4.5 New tab)

The first option is to select the default template which was earlier set by the user.



(fig 4.6 New tab :using default)

The user is also allowed to use any existing template. The Listview control will display the list of existing template. User has to simply select one of them.



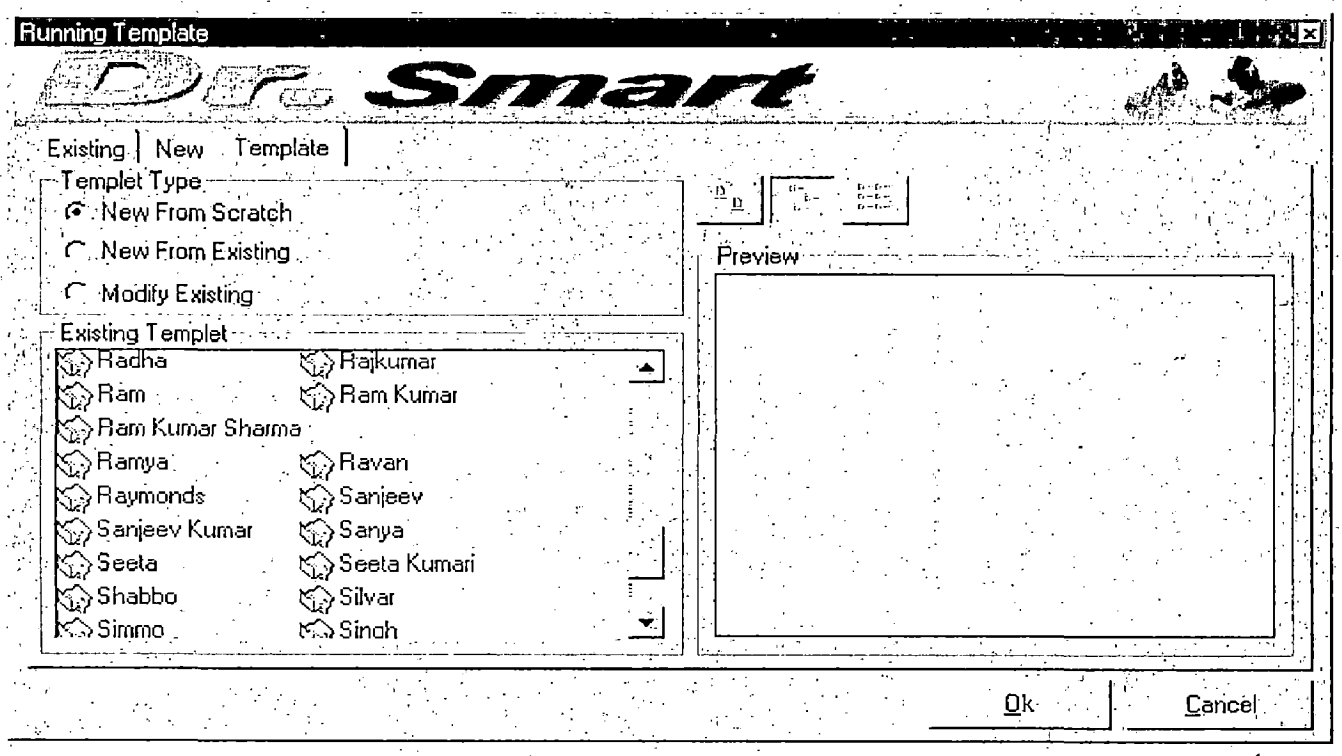
(fig 4.7 New tab: using other template)

Template tab will allow user to design a new template. There are three options for user. They are:

(a) Creating a new template from scratch

(b) Creating a new template by using existing one

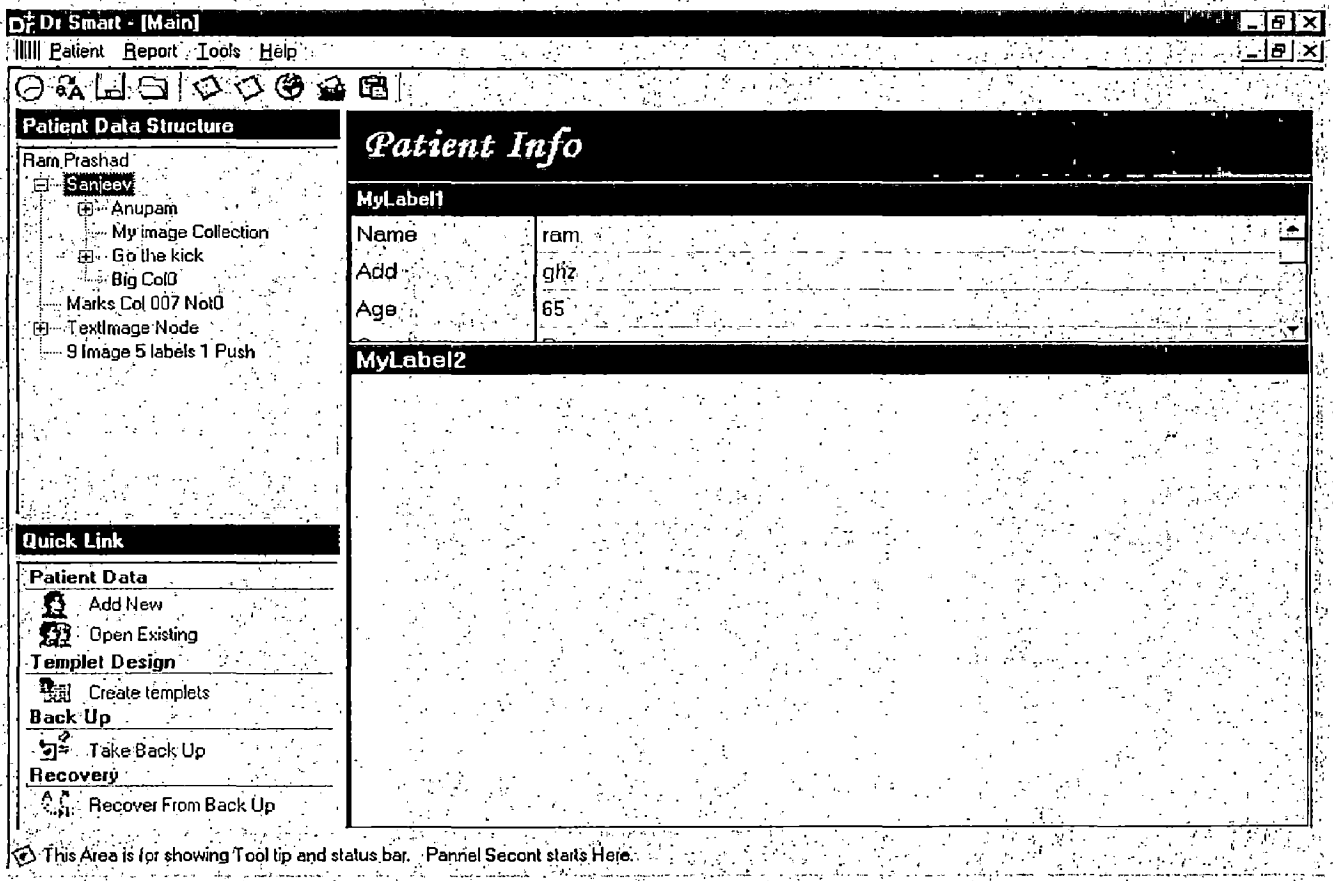
(c) Modify an existing template.



(fig 4.8 Template tab)

4.1.3 Patient Information screen(frmMain) :

This is the main screen which will be used to enter information about a patient in a manner decided by a template selected for that particular patient.



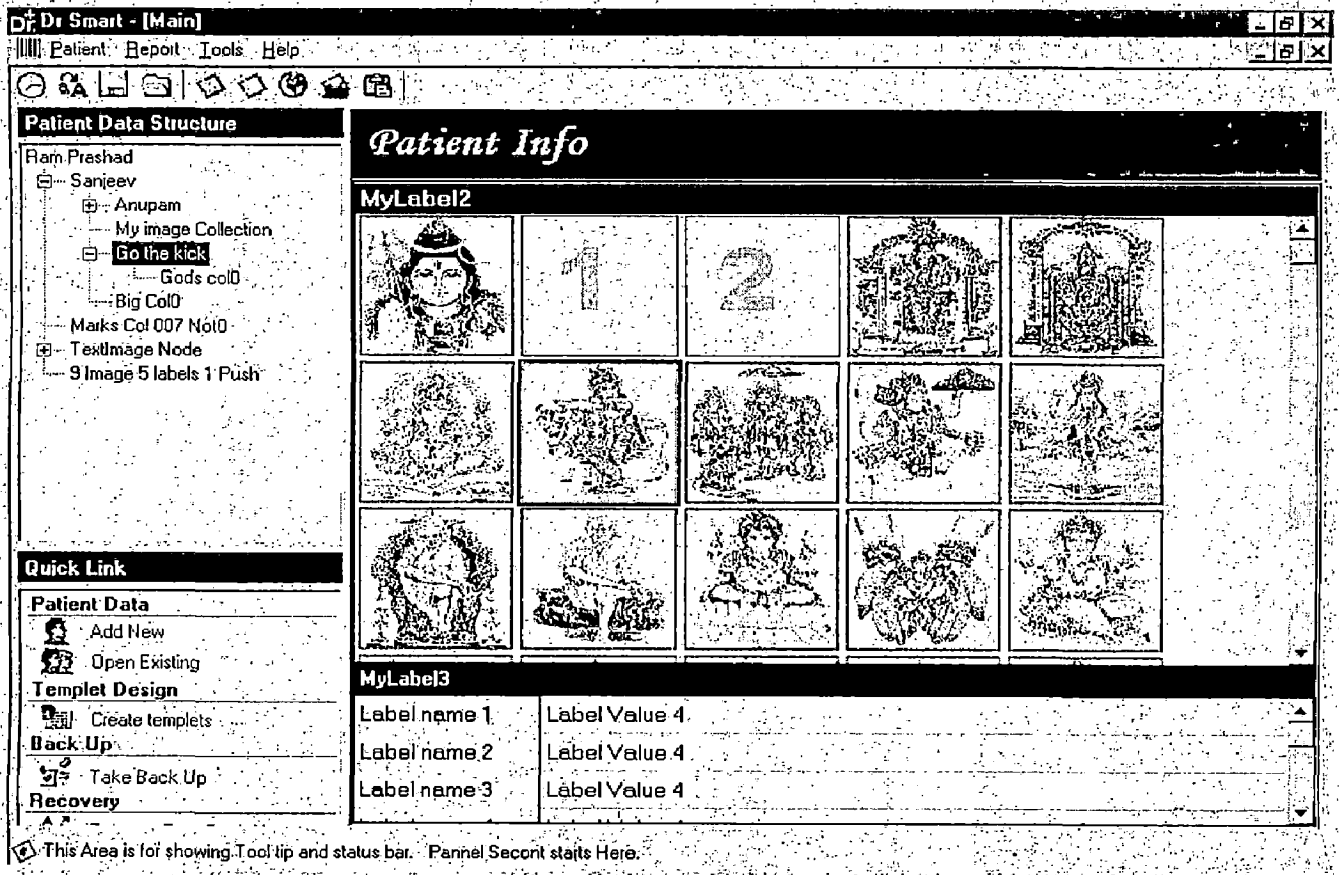
(fig 4.9 Patient Information : Text Node)

On Left hand side, a tree view control is used to display the template structure associated with the patient.

On Right hand side, the screen will display the information contained in the selected node of information.

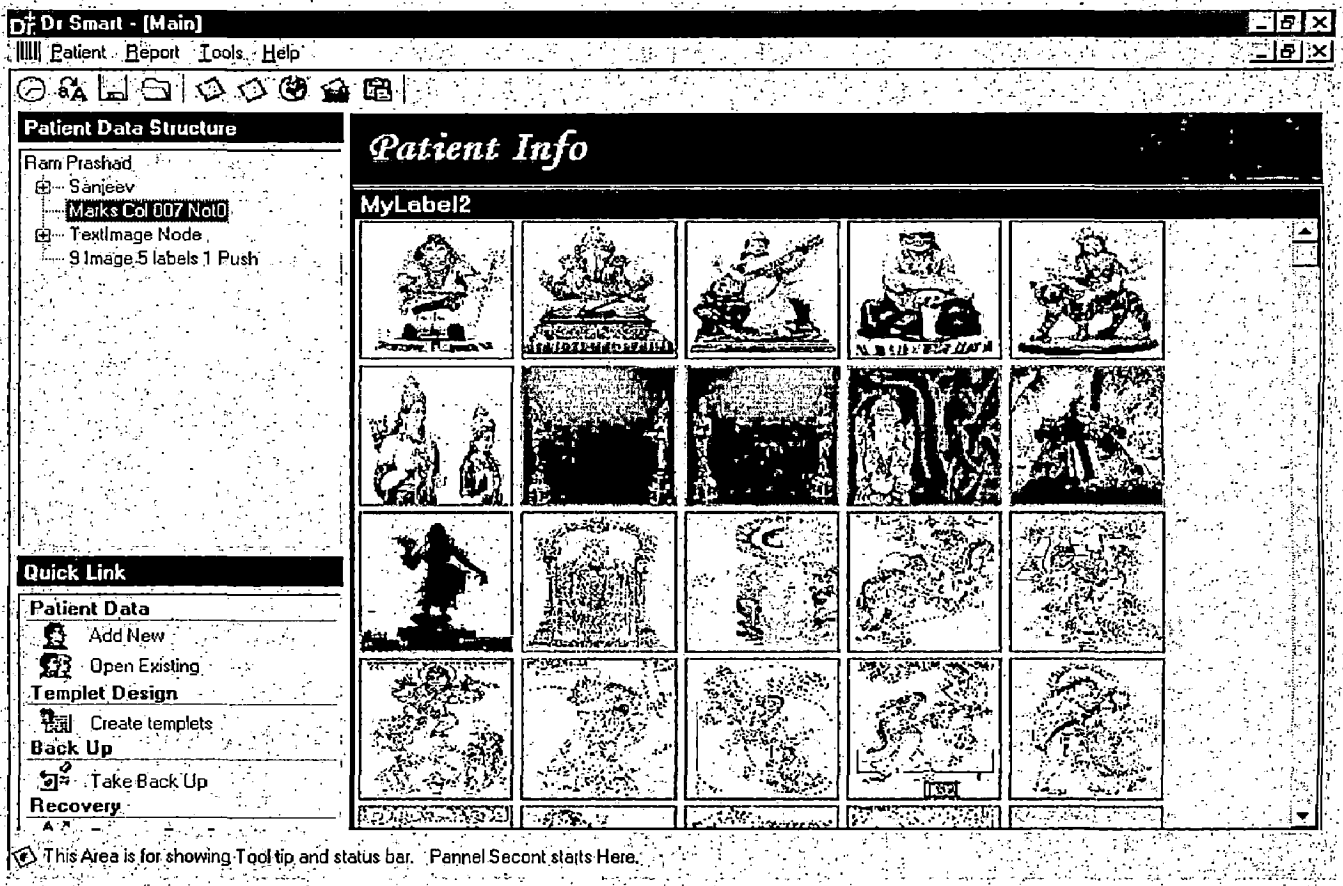
To display the blank node or filled node, custom controls are used depending on the type of information node.

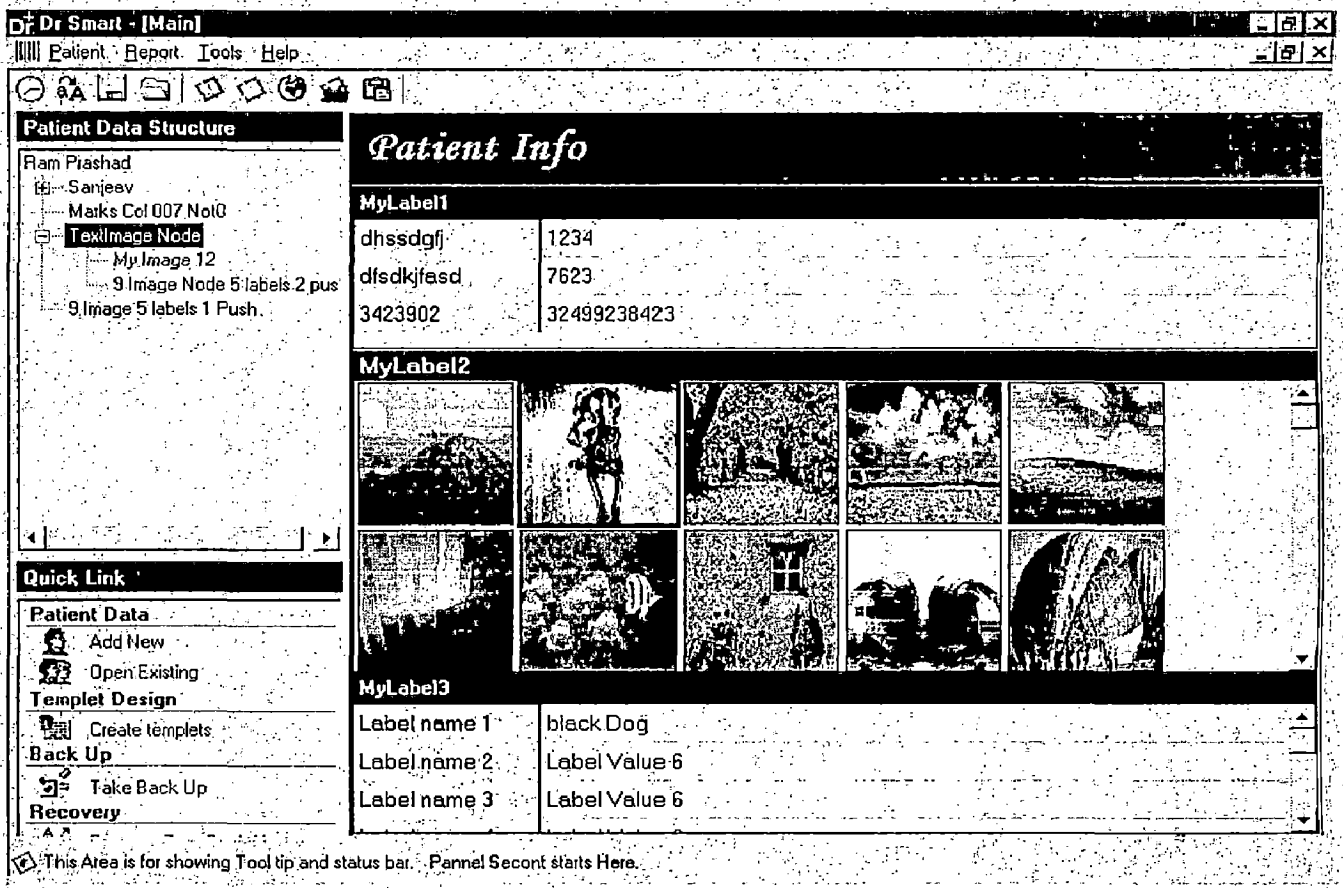
For **Text node**, a custom control called Text Grid is used which is designed using MSFlexGrid control, labels & Textboxes as shown in fig. 4.9



(fig 4.10 Patient Information : Image with text Node)

For **Image node,** a custom control called ImageGrid is used which is designed using another custom control, TextGrid , MSFlexGrid control and picture box. Imagelist control is also used to develop this control.





(fig 4.11 & 4.12 Patient Information : Image only & Image with Text Node)

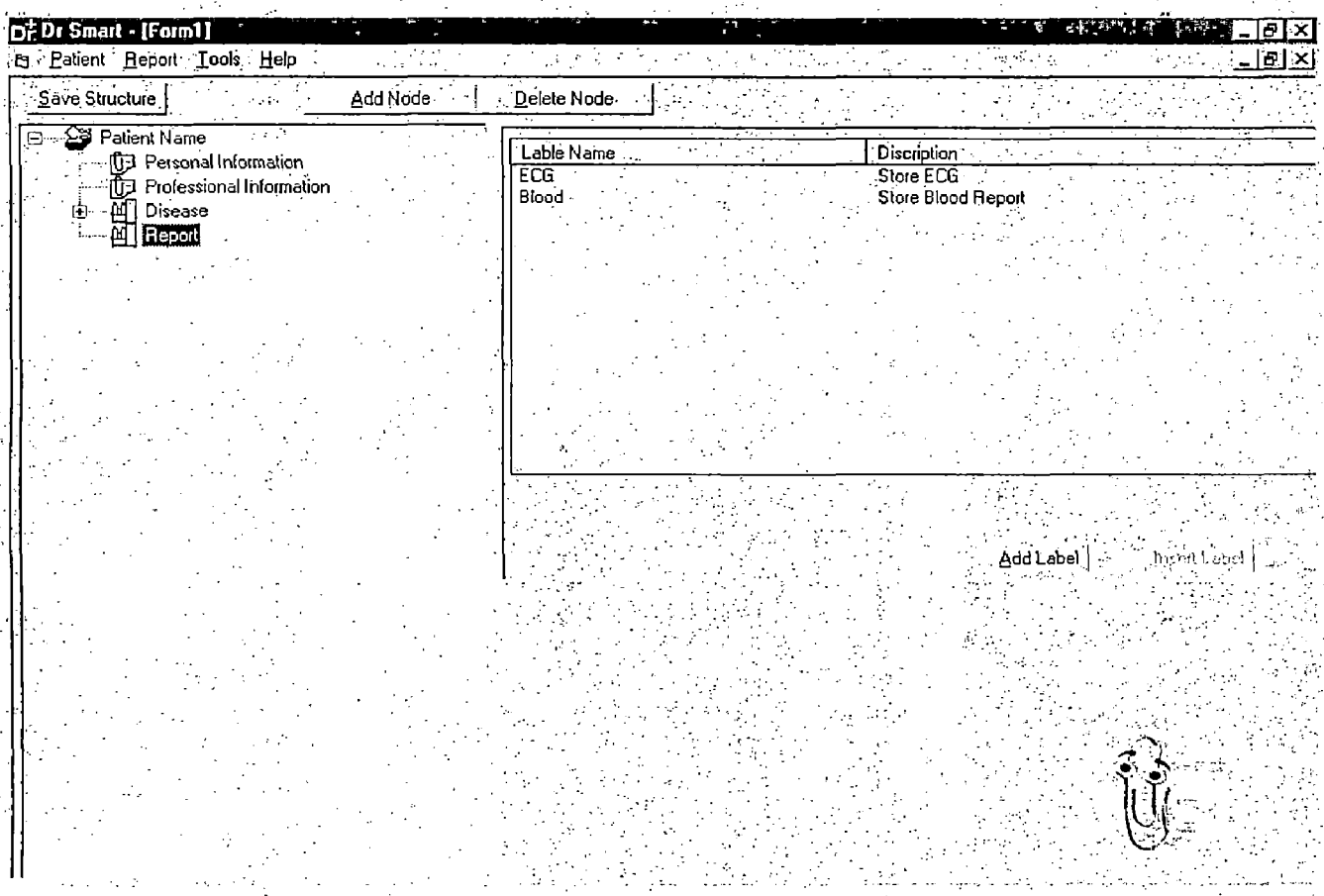
4.2 Template designing

Template design (frmTemplate) :

It is mainly implemented using Treeview, Listview, toolbar and picturebox control.

The major control is treeview. A TreeView control displays a hierarchical list of Node objects, each of which consists of a label and a bitmap. Each node represents information about a patient. The type of node decides the type of information it contained.

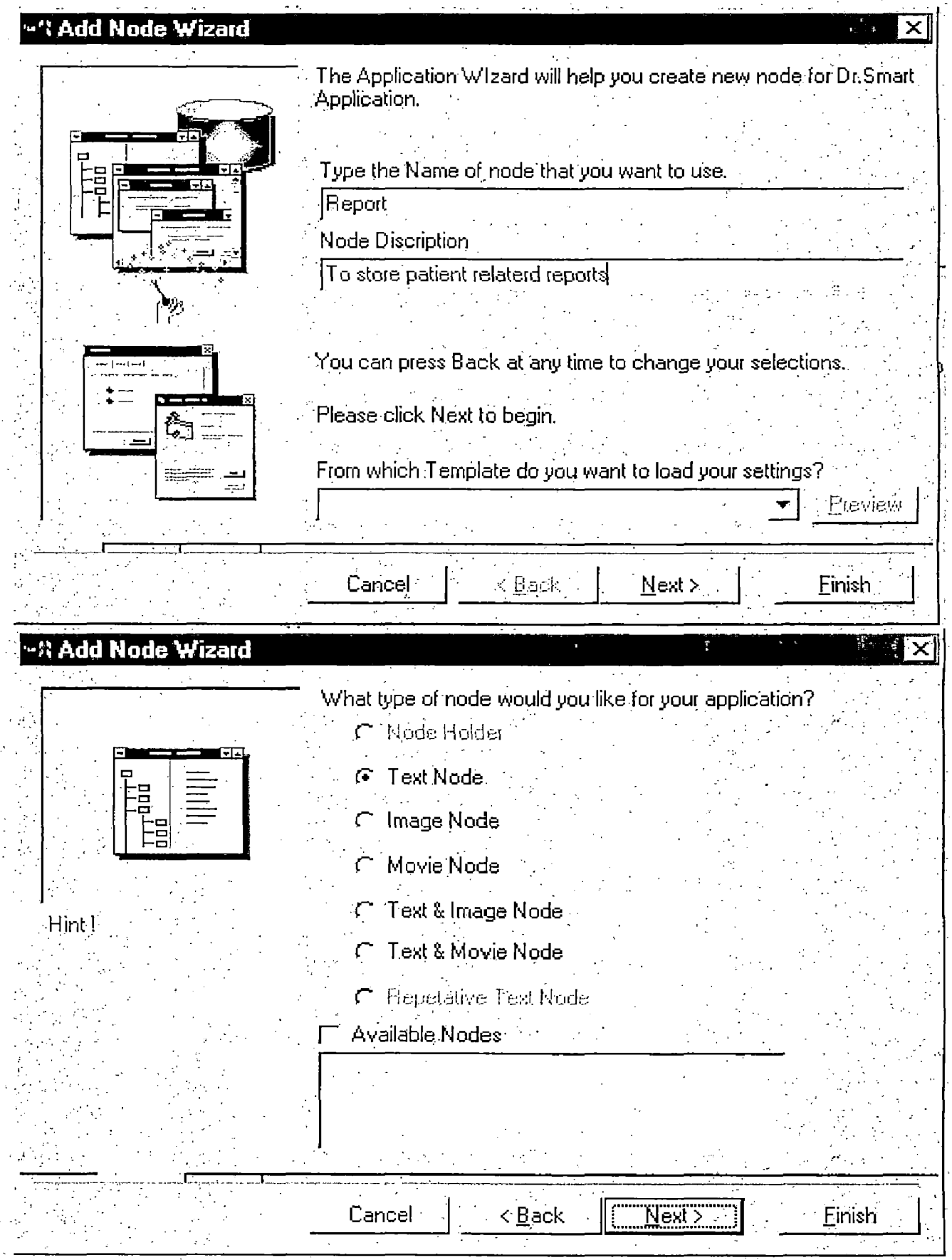
Using a TreeView control, user can add, remove, arrange, and otherwise manipulate Node objects by setting properties and invoking methods. Three events, the Collapse, Expand, and NodeClick event, provide programming functionality.



(fig 4.13 Template Design)

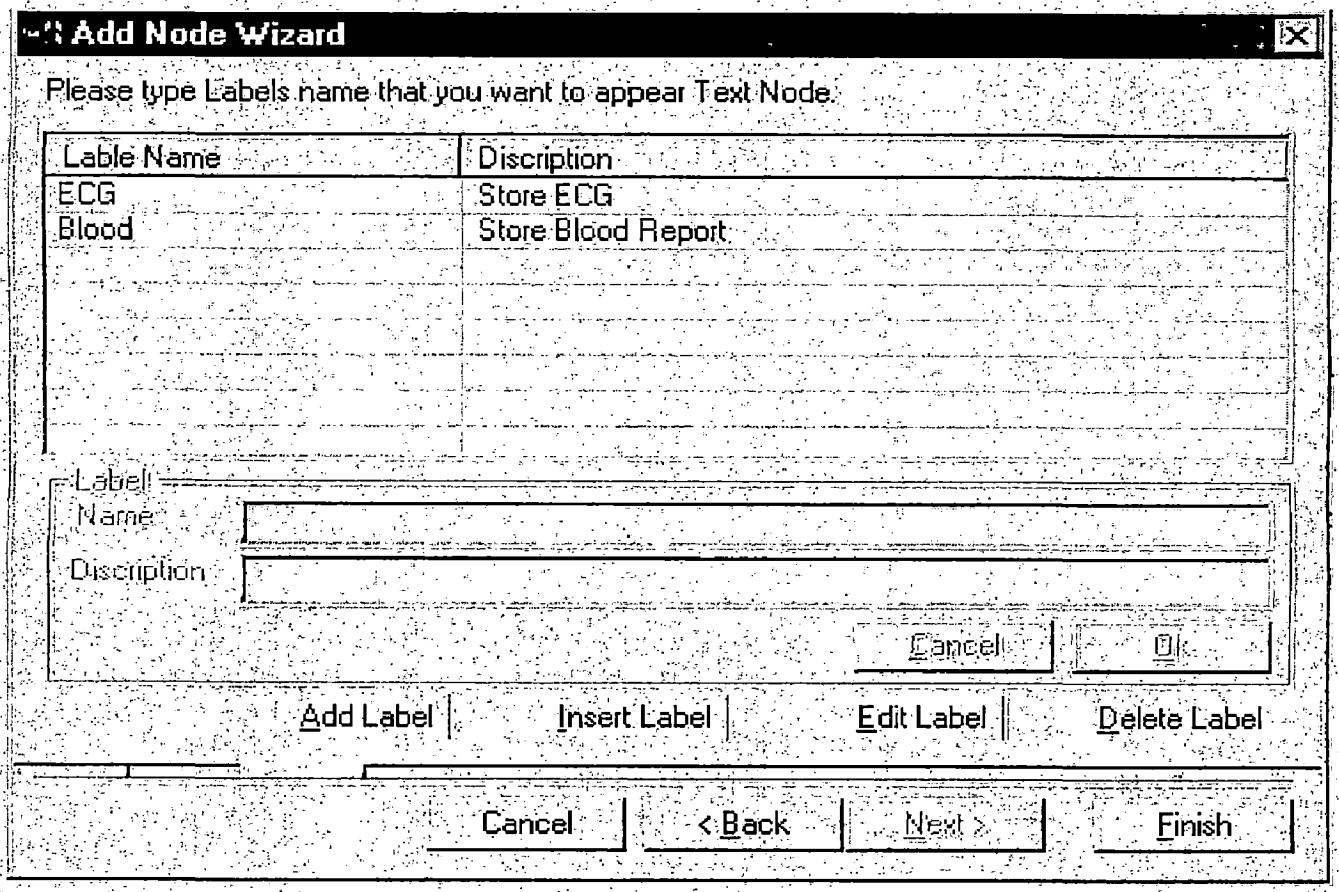
Tool Bar is designed for three basic operations:

- Add Node
- Delete Node
- Save Structure



(fig 4.14 & fig 4.15 Add Node Wizard (Step 1 & 2))

In **AddNode,** again we created a wizard using SSTab where each tab is shown in a sequence. One can easily interact with this wizard where common controls are used to add an information node & type of information it contained.



(fig 4.16Add Node Wizard (Step 3)

In **DeleteNode**, user has to simply select a node to be deleted, and press delete button on the toolbar. A method will be called to delete that particular node.

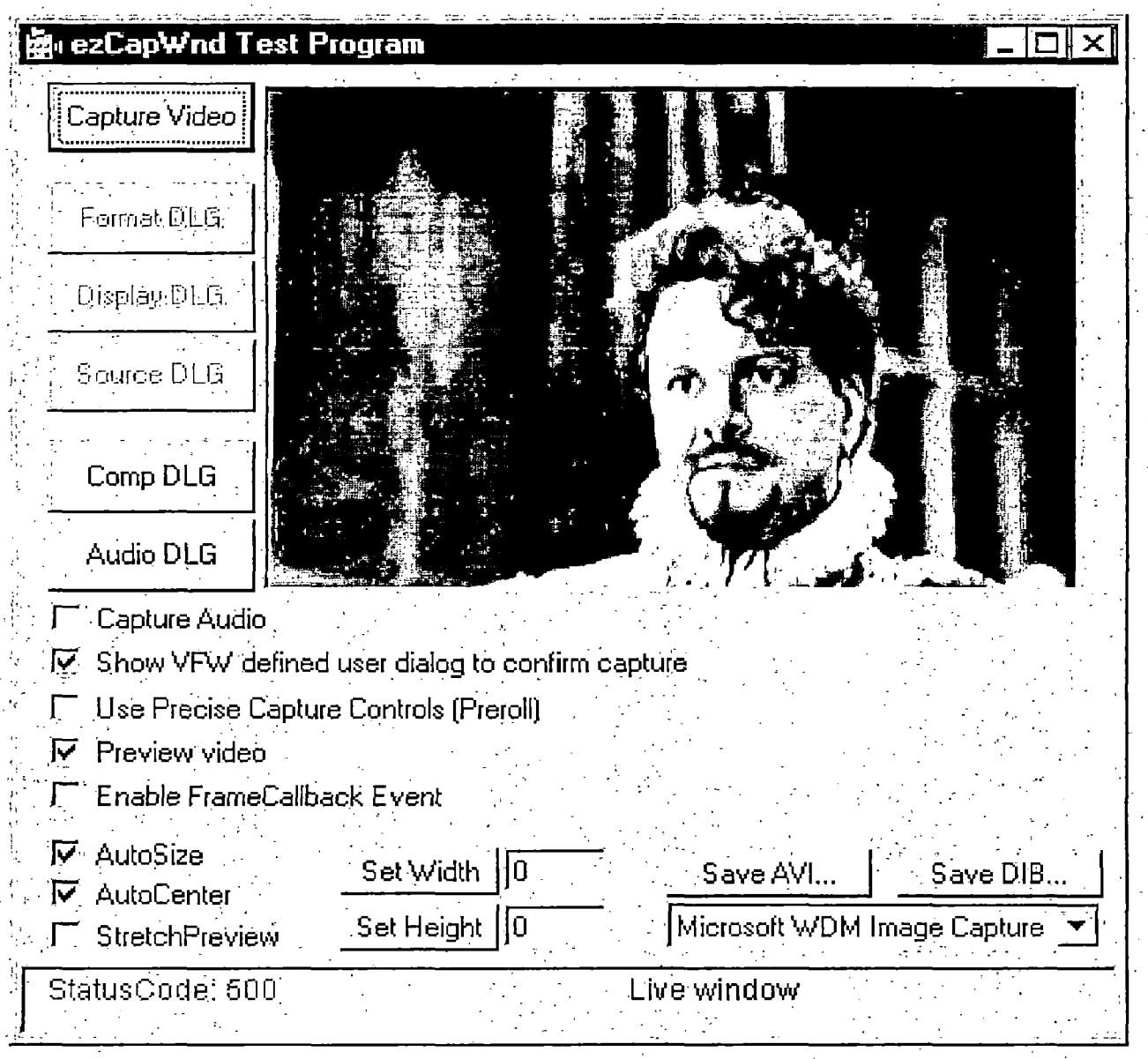
In **SaveStructure,** the hierarchical structure will be saved in a file as per the format designed for storing template as shown in 3.2.2.2.

4.3 Data acquisition from various sources

Data acquisition from various sources has been achieved using the support of e technical concepts of Windows Development. The major concepts used are DirectX (DirectDraw, DirectInput, DirectShow), Platform SDK (WDM functions & constructs for video capture).

We usually capture Data in the form of still image or Video Format. For capturing purpose, we need a capture card which must be installed in the system. For capturing we pass a message to the Windows Operating System to create a Capture Window. After creating capture window we find out how many and which Capture Driver we have (usually provided by the Capture Card).Then using the Win32 API Messaging, we send message to the card driver to make a communication link between the capture card and the capture window. The Device from which we'll have to capture the data should be connected to the capture card and must be working properly. Now the selection can be provided to select the input type that is going to work as a source like: S-Video, Composite Video or TV Tuner. After the input type has been selected then the user will be capable to see whatever the device is giving to the capture window. Now with the help of this infrastructure we can either capture a video or a still image.

To start the video capture; the user'll have to click a command button (Fig. 4.17) which will start the capture process and store the frames into the memory buffer. As soon as the user click on the Stop capture button; the capturing get stopped. The user can have a glance at whatever he has captured using the preview button available. If the user is satisfied with the captured video he can save the video by giving a name. Almost the same procedure has to apply to capture the still images.

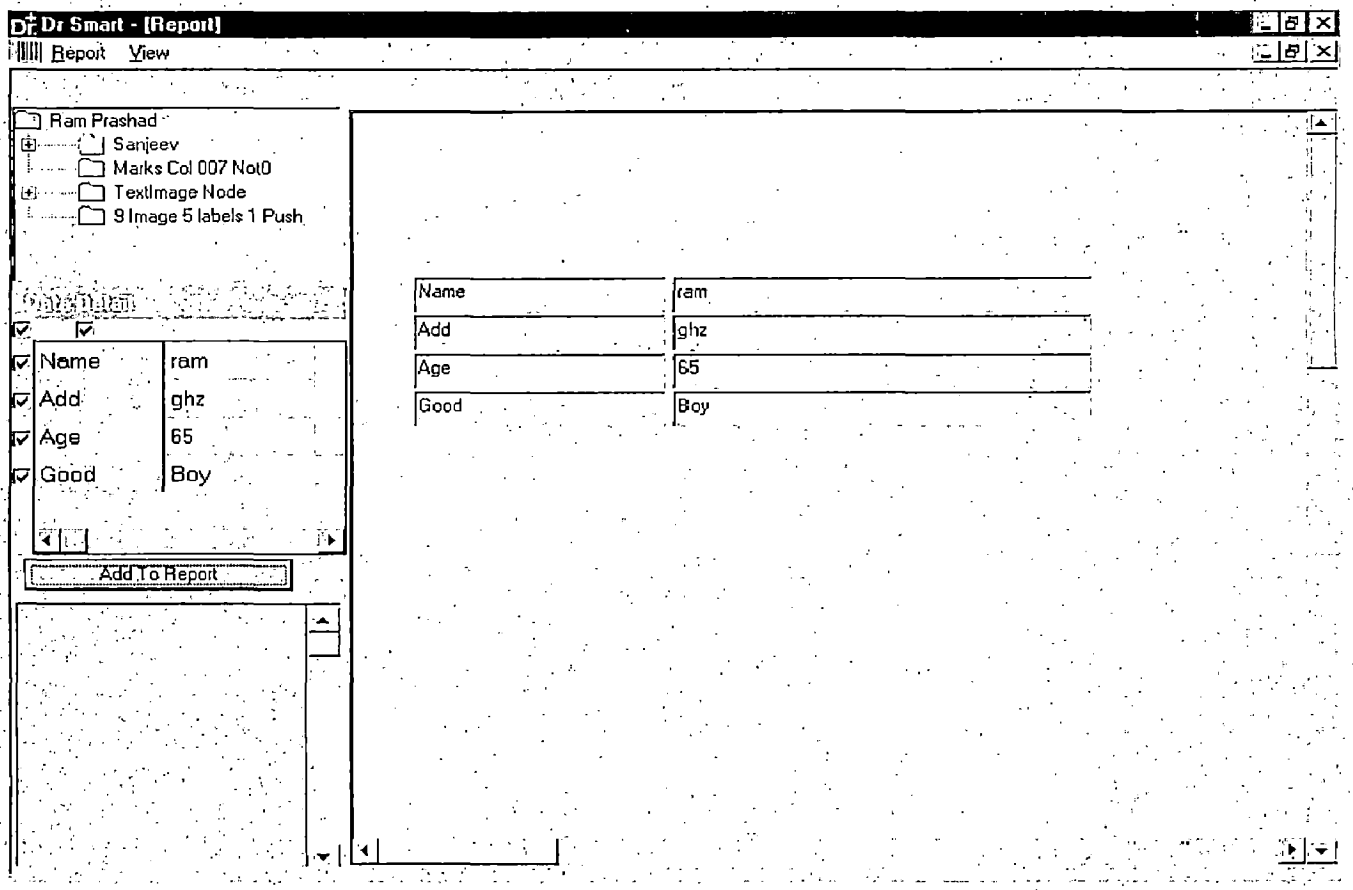


(fig 4.17 Video Capture)

4.4 Output in desired format

The report generation module is an output module of the software. The module takes input from the main patient screen. Patient information structure is displayed in the left pane of the screen. Depending on the type of the node chosen, we get the preview of the contents of that node in lower preview regions available. If the node chosen have text data then only textual data is displayed in the preview region. To display the preview of text data a customized control known as text_grid_ver2 has been used. If we have to display the text data in our report then the data should be chosen from this control. As soon as the user click on the add to report button the selected text appears on the printing palette (Which is the right portion of the screen and contents of only

this control appear on the printed report).The same can be achieved using the node drag and drop operation where the complete node information will be transferred on the printing palette.



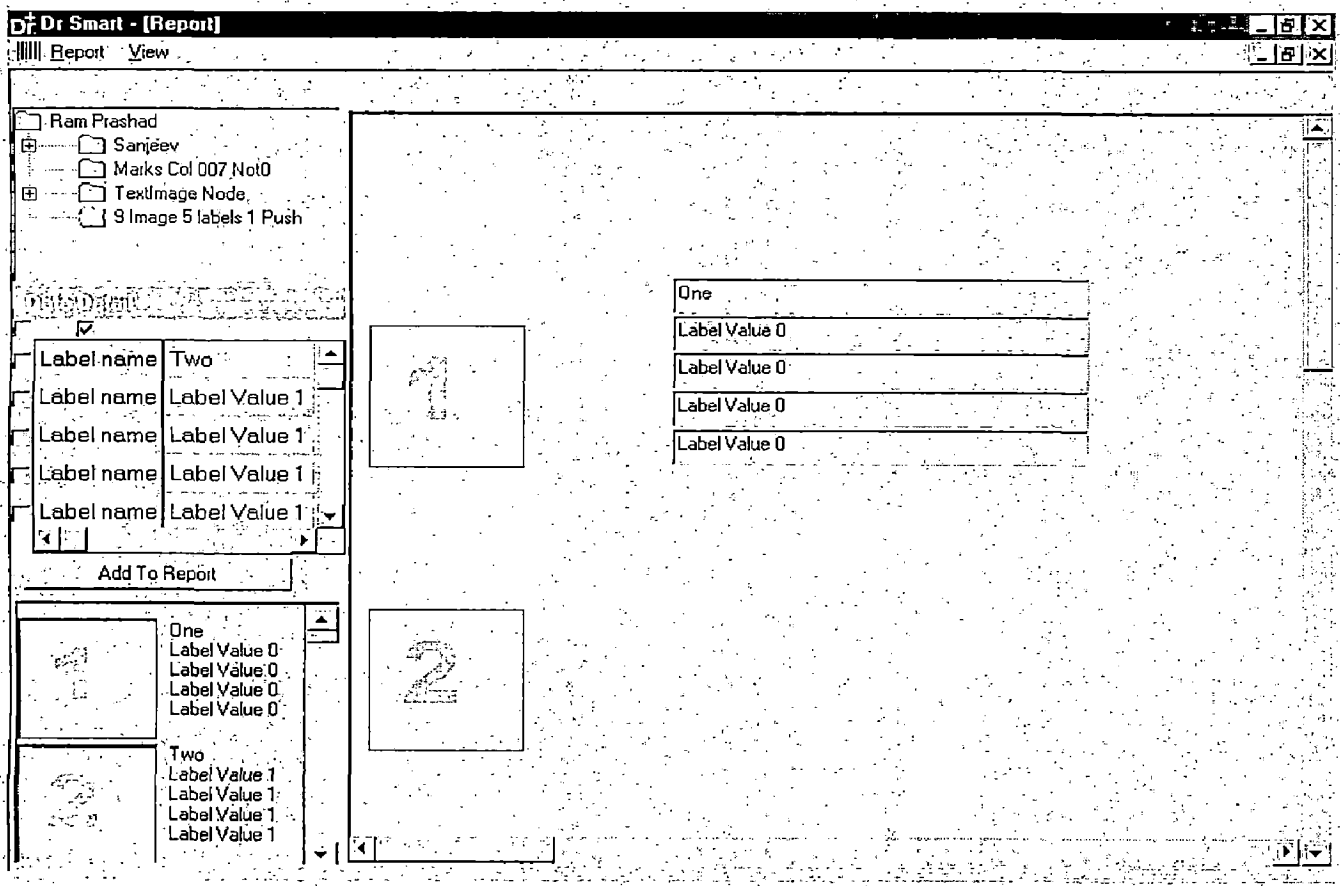
(fig 4.18 Report Generation)

If the user clicks on the image node, the images attached with node get displayed in the lower preview region. If the user selects any of the image available the text data attached with the image get displayed in the same manner as in the case of the text node. The image has to be dragged on to the printing palette. And the text related with the image appears on the printing palette in the same manner as in the case of text node.

The Video & its related information are not available for printing purpose.

The user have a facility for page setup where he can setup the margins of the page.

Whenever the user wants to print the content, a print dialog box appears which give the user the various options which are available in standard print dialog box. As soon as the user clicks on the OK button of the print dialog box the contents of the printing palette get printed.



(fig 4.19 Report Generation)

RESULTS AND DISCUSSION

This is a desktop version of the system for recording medical data and support system acquisition. The basic aim of storing all kind of medical data of a patient is achieved by designing a new structure which contains a unique file structure for storing patient information. A file is created for each individual patient which contains the structure as well as data (information). The flexibility that the system provides for storing any kind of information in a desired format is the key feature of this thesis.

There are provisions for capturing textual data from keyboard and using Graphire tool. The images can be linked to the patient information file after video capturing from a video source.

This version is also generating output of the patient information by giving provisions for setting the report format. Therefore this system can be customized to any format of a report of any patient. An exemplary report of a patient is printed overleaf.

For reducing the size of the patient file, necessary compression is done using Intel JPEG library.

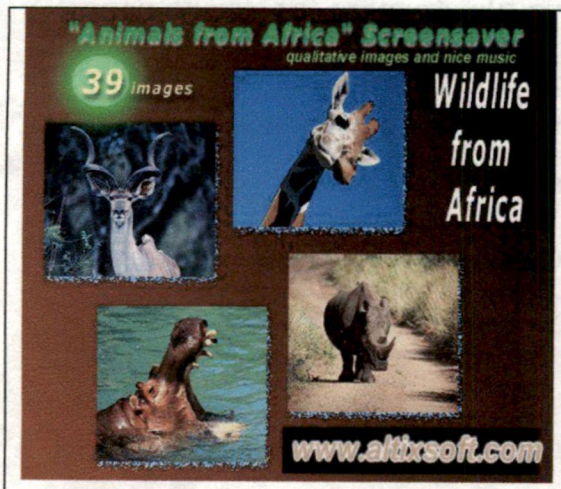
This desktop version can be enhanced to a server version for large hospitals and nursing homes where a centrally located server can be used to record all medical data of all the patients and clients machines can be used to acquire information from various sources such as doctors, X-Rays, Cat-Scan, MRI, pathology etc.

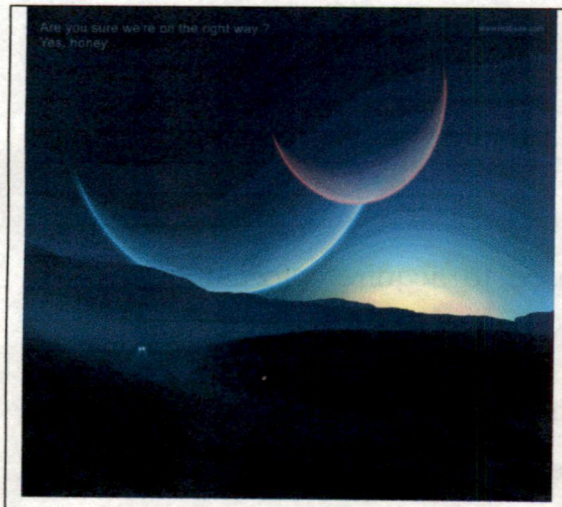
This can also be converted to a network version where it can communicate with any other computer and can be used in the field of telemedicine.

Limitations:

- It is a desktop version.
- Features like backup & recovery, security etc. are yet to be implemented.
- Testing with real data is still in progress.
- System is dependent on Windows operating system, but it can be made system independent by implementing windows .net environment.

Name: Abhinav **Age: 15**

	**Label value1**
	Label value2
	Label value3
	Label value4
	Label value5

	**Label value1**
	Label value2
	Label value3
	Label value4
	Label value5

CONCLUSION

As per the **NASSCOM Strategic Review, 2003**[11], one of the key trend witnessed during this year was that Indian companies had increased their focus on expanding their presence in several new areas such as HealthCare, retailing and utilities.

The health care market accounts for around US$ 12 billions of global IT services industry. Investments in IT by the HealthCare sector are likely to be focused mainly on customer (patient) management systems and maintaining electronic medical records. For instance, in the Us, regulatory provisions require HealthCare service providers to maintain an increasing proportion of their medical records in electronic form, to enable patients to switch service provider easily.

Several Indian companies have recently increased their focus on the HealthCare segment, especially healthcare service providers, health insurance, life sciences (bio-technology and bio-informatics) and medical equipment. TCS and Siemens Information system(SISL) have developed healthcare applications such as HMS and clinical record systems. Wipro acquired GE Medical systems Information Technology Pvt. Ltd and created a new division Wipro Health care and life sciences.

Recently, in Chandigarh, Project Telemedicine interconnecting three premier medical institutes, i.e. PGI- Chandigarh, AIIMS- Delhi and SGPG-Lucknow using ISDN for tele-diagnosis, tele-consultation and tele-education, each of which in turn will be connected with one medical college, has been intiated.

Therefore, in this context, the system developed by us is appropriate and timely started. Seeing the scope in this sector, it can be enhanced to any limits. So, it is just a beginning on my part.

REFERENCES

[1] Philips Medical systems, Bangalore, ShivKumar K.R.

[2] www.ehirc.com , Benoy George Thomas, Escorts

[3] Apollo, R.Srinivasan and Neelesh Katiyar

[4] www.aiims.ac.in , Dr. R.S.Tyagi, Dy. Dir and Head Computer Facility and Dr. G.S..Rao, Dept of Biophysics.

[5] Siemens Medical Solutions Division, Mr.Heinrich von Wulfer and Dr. GB Advani

[6] MSDN Library, Jan 2001

[7] Dan Appleman , "Visual Basic Programmer s Guide to the WIN32 API ", Techmedia, 1998.

[8] Steven Roman, "Win32 API Programming with Visual Basic" , O'REILLY, Shroff Publishers and Distributers Pvt. Ltd., March 2000.

[9] Damon Chandler Mechael Fotch, "Windows Graphics API ", Dreamtech Press.

[10] Intel JPGE Library (IJL) from intel.com

[11] Strategic Review,2003, The IT Industry in India, NASSCOM