

Contents

1. INTRODUCTION

1.1 Pharmacy Database Management System

2. PROJECT PLAN

2.1 Feasibility study

2.1.1 Operation

2.1.2 Technical

2.1.3 Cost/Benefit Analysis

2.2 System Analysis

2.3 System Requirement

2.4 Data-Flow-Diagram

2.4.1 Context DFD

2.4.2 First Level DFD

2.4.3 Second Level DFD

2.5 System Design

2.6 Coding details / Application

2.7 System Testing

3. WHY?

3.1 Visual Basic

3.2 Microsoft Access

CONCLUSION AND FUTURE SCOPE

BIBLIOGRAPHY AND REFERENCES

Chapter 1

INTRODUCTION

Pharmacy Database Management System

Introduction:

The main aim of the project is the management of the database of the pharmaceutical shop. This is done by creating a database of the available medicines in the shop. The database is then connected to the main program by using interconnection of the Visual Basic program and the database already created.

Application:

This program can be used in any pharmaceutical shops having a database to maintain. The software used can generate reports, as per the user's requirements. The software can print invoices, bills, receipts etc. It can also maintain the record of supplies sent in by the supplier

Chapter 2

PROJECT PLAN

2.1 Feasibility Study

A feasibility analysis involves a detailed assessment of the need, value and practicality of a proposed enterprise, such as systems development. The process of designing and implementing record keeping systems has significant accountability and resource implications for an organization. Feasibility analysis will help you make informed and transparent decisions at crucial points during the developmental process to determine whether it is operationally, economically and technically realistic to proceed with a particular course of action.

Most feasibility studies are distinguished for both users and analysts. First, the study often presupposes that when the feasibility document is being prepared, the analyst is in a position to evaluate solutions. Second, most studies tend to overlook the confusion inherent in system development – the constraints and the assumed attitudes.

2.1.1 Operational feasibility

People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. It is common knowledge that computer installations have something to do with turnover, transfers, retraining, and changes in employee job status. Therefore, it is understood that the introduction of a candidate system requires special effort to educate, sell and train the staff on new ways of conducting business.

2.1.2 Technical feasibility

Technical feasibility centers around the existing computer system (hardware, software, etc.) and to what extent it can support the proposed addition. For example, if the current computer is operating at 80 percent capacity – an arbitrary ceiling – then running another application could overload the system or require additional hardware.

This involves financial considerations to accommodate technical enhancements. If the budget is a serious constraint, then the project is judged not feasible.

2.1.3 Cost/ Benefit analysis

Economic analysis is the most frequently used method for evaluating the effectiveness of a candidate system. More commonly known as cost benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. Otherwise, further justification or alterations in the proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase in the system life cycle.

Costs:

- ☐ Cost of new computer approximately Rs. 22,000/-
- ☐ Cost of operating system approximately Rs. 5000/-

Benefits:

- ☐ Avoids tedious typing task
- ☐ Faster document retrieval
- ☐ Saving storage space
- ☐ Keeps data secure
- ☐ Easy to use, update and maintain

2.2 System Analysis

It is the most creative and challenging phase of the system life cycle. The analysis phase is used to design the logical model of the system whereas the design phase is used to design the physical model.

Many things are to be done in this phase .we began the designing process by identifying forms, reports and the other outputs the system will produce. Then the specify data on each were pinpointed. we sketched the forms or say, the displays, as expected to appear, on paper, so it serves as model for the project to began finally we design the form on computer display, using one of the automated system design tool, that is VISUAL BASIC 6.0.

After the forms were designed, the next step was to specify the data to be inputted, calculated and stored individual data items and calculation procedure were written in detail. File structure such as paper files were selected the procedures were written so as how to process the data and procedures the output during the programming phase. The documents were design ion the form of charts.

Output design means what should be the format for presenting the results. It should be in most convenient and attractive format for the user. The input design deals with what should be the input to the system and thus prepare the input format. File design deals with how the data has to be stored on physical devices. Process design includes the description of the procedure for carrying out operations on the given data.

2.3 System Requirements

The system services and goals are established by consultation with system user. They are then defined in details and serve as a system specification. System requirement are those on which the system runs.

Hardware Requirements:

- Computer with either Intel Pentium processor or AMD processor.
- 128MB DDR RAM
- 40GB hard disk drive

Software Requirements:

- Windows 98/2000/XP operating system.
- Microsoft Office package.
- Microsoft Visual Studio 6.0

2.4 Data Flow Diagrams

A data flow diagram is a graphical representation or technique depicting information flow and transform that are applied as data moved from input to output. The DFD are partitioned into levels that represent increasing information flow and functional details. The processes, data store, data flow, etc are described in Data Dictionary.

Data flow:



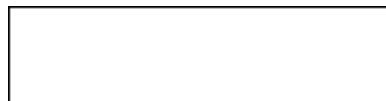
Data moves in a specific direction from an origin to destination

Process:



Procedure s people or devices that use or transform data

External entity:



This defines a source (originator) or destination of system data.

Data Store:



This indicates where data is stored in the system.

2.4.1 Context Diagram

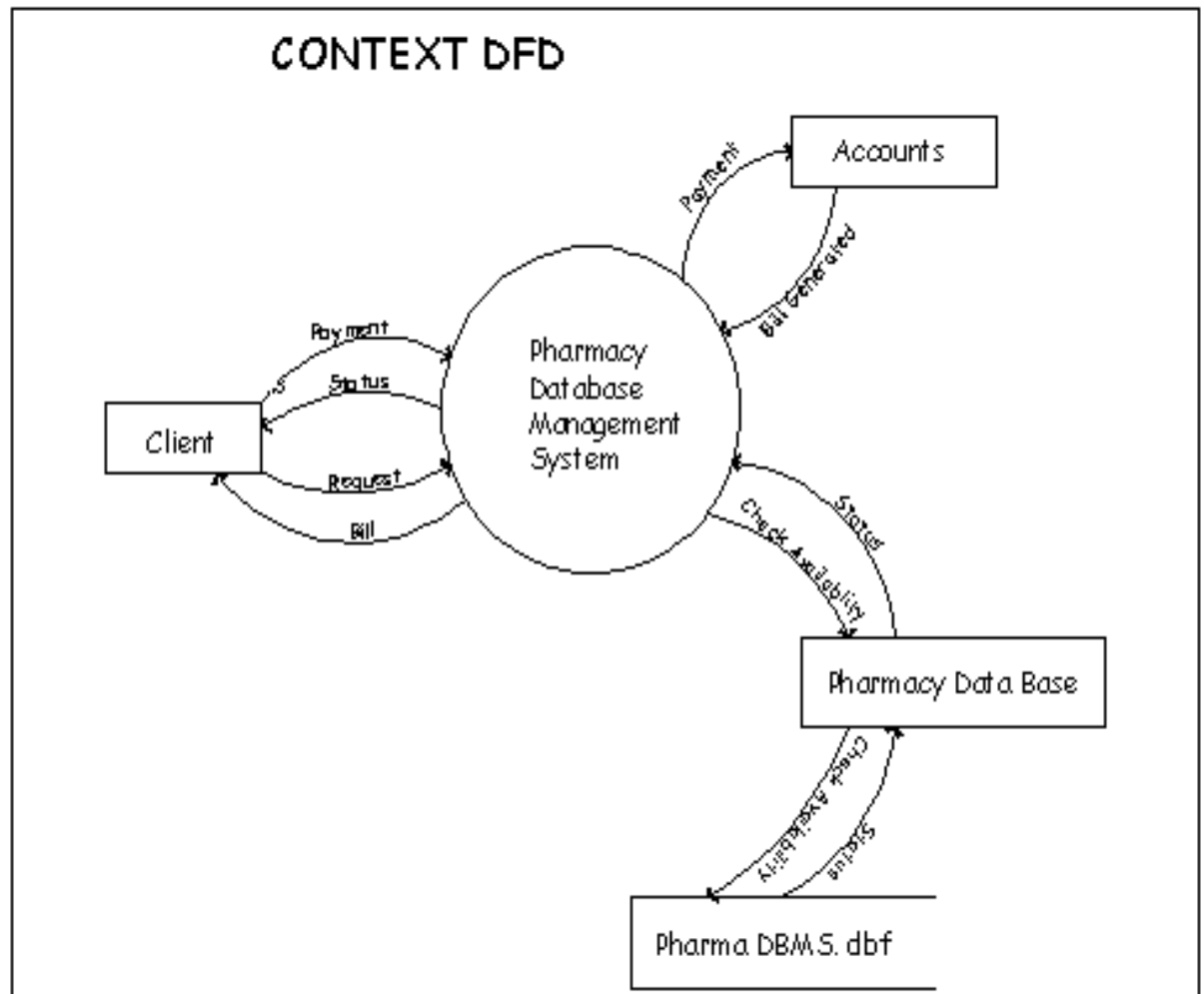


Fig. 0th Level DFD

In the 0th level of the DFD the client request the Pharmacy Database process for some product or the medicine than the process gives the check availability signal to the pharmacy Database for the requested product or the medicines availability. After checking the availability, the Database sends the status to the Pharmacy Database process. Then the Pharmacy Database process gives the status to the client and according to the status the client buys the product and pays the bill and the external entity Accounts than generates the bill for the purchased product.

2.4.2 First Level DFD

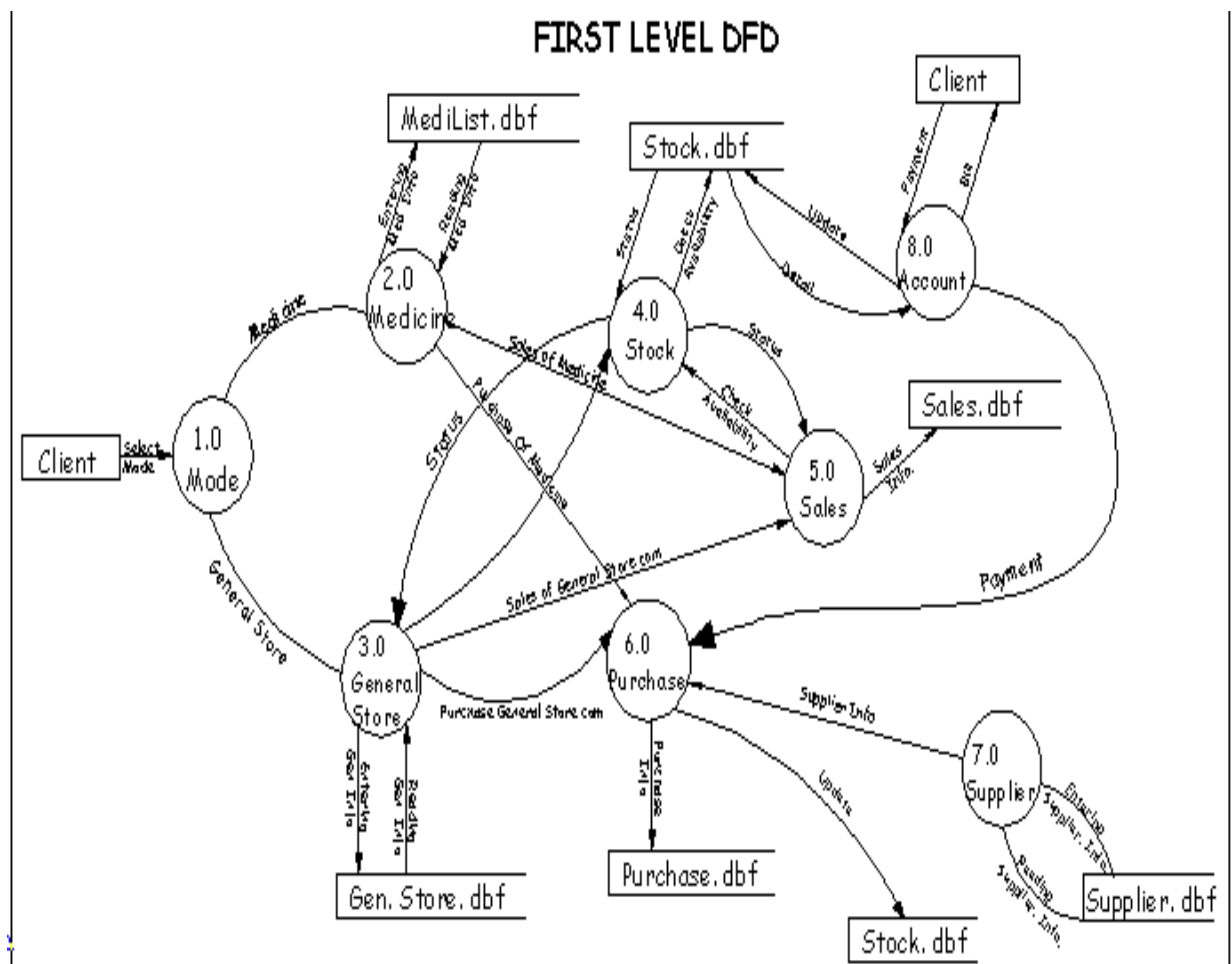


Fig. 1st Level DFD

In level one of the DFD the client select the mode of the action i.e. whether he wants to buy a medicine or general store product. If he selects mode as medicine than the flow of data will be as follows:

The process 2.0 i.e. medicine can enter the medicine information into the Database or can retrieve the information from the Database. If the medicine has to soled, than the sales process will check the stock whether the requested medicine is

available or not, this will be done by checking the availability of the medicine and the stock process will reply by giving the status of the available stock.. If the requested medicine is available than the client will pay the bill and the account process will generate the bill for the purchased medicine.

If the medicine is purchased than, first the supplier's information is retrieved from the suppliers Database. After purchasing the bill amount of the purchased medicine is paid by the account process and the stock Database is updated automatically after the new medicines are purchased.

If the client selects the mode as general store i.e. if he wants to buy a general store product, the general store process will ask to the sales process for the requested product, than to check whether the product is available or not the sales process will check the stock by giving the check availability request to the stock process and than the stock process will reply by giving the stock status. If the stock of the requested product is available then the client will pay the bill and in turn the Account process will generate the bill for the product purchased by the client.

If the product is purchased than the information of the supplier from whom the product is purchased is retrieved from the supplier process by the purchase process and if the supplier is new than the supplier information is entered in the suppliers database.

The Account process also keeps all the details of the stock.

2.4.3 Second Level DFD

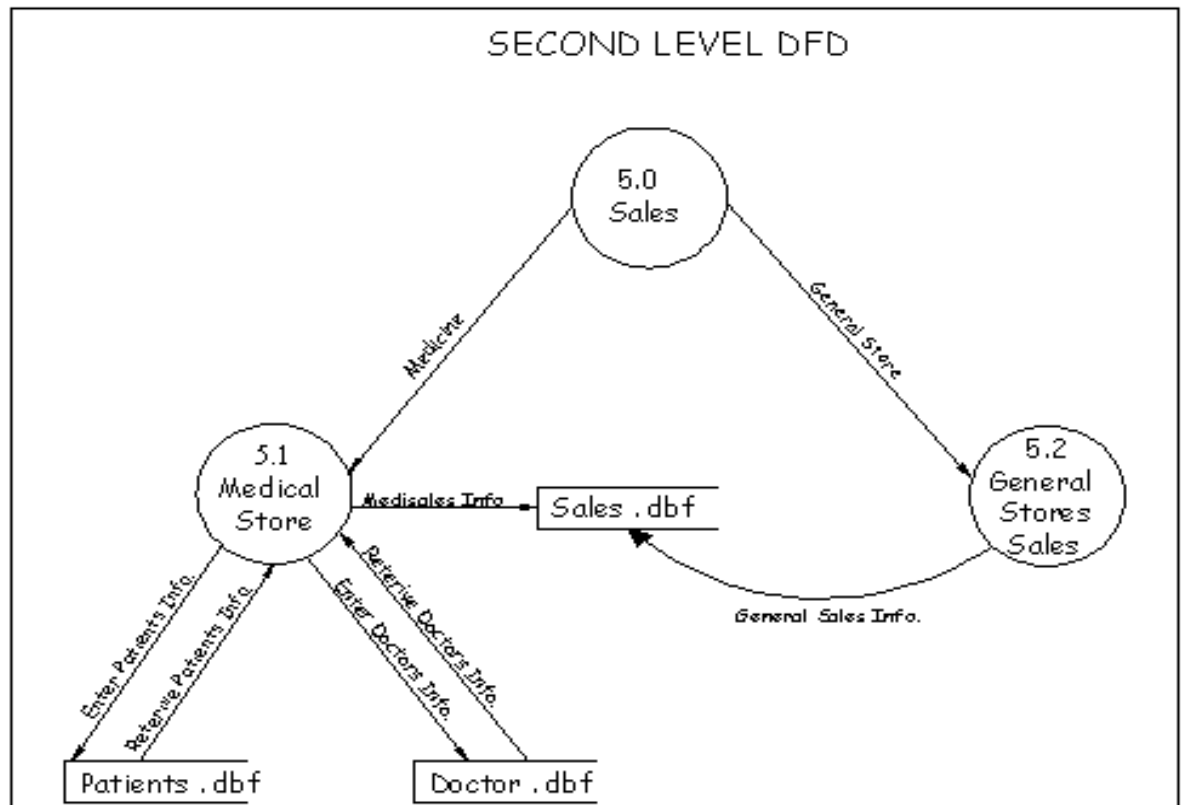


Fig. 2nd Level DFD

Further in the second level DFD the process 5.0 i.e. sales process is elaborated.

In the sale process, after selecting the mode i.e. medicine or general store the further operation is performed. The 5.1 process is the Medicine sale process in this process the patient's information can be retrieved from the patient Database. If the patient is visiting for the first time than his information or detail is entered in the patient database.

Similarly the Doctor's information or detail is also entered in the Doctor Database, who referred to the patient. If the patient is referred by the new Doctor than his information can be entered in the doctor's Database.

With the help of the medicine sales process the patient's and Doctor's information can be entered or retrieved from the respective Database.

Pharmacy Database Management System

If the medicine or general store product has to be sold the information is retrieved from the sales database.

2.5 System Design



Fig. Main Screen

When the system loaded the above screen appears.

Pharmacy Database Management System

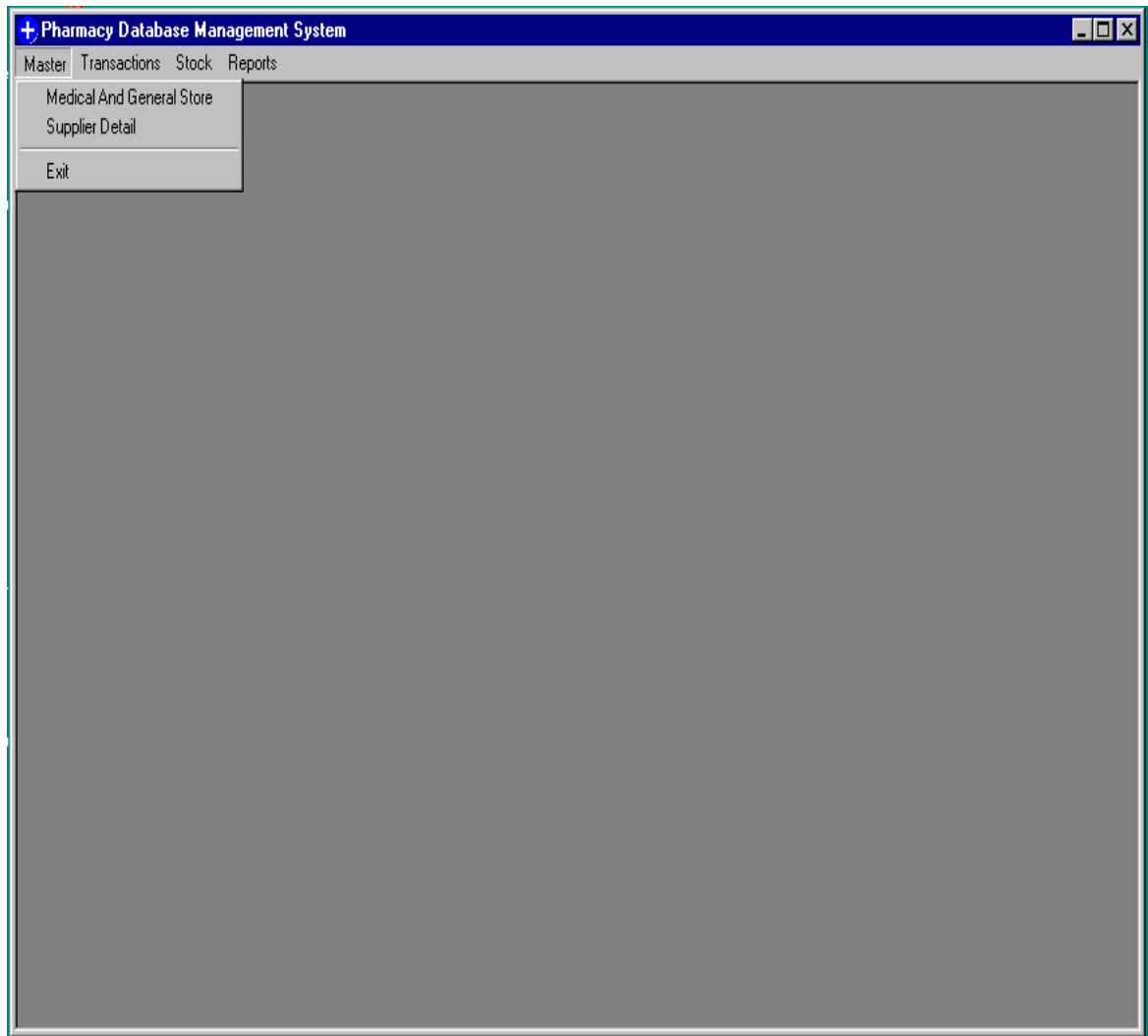


Fig. Master Menu

The above screen displays the option available under the Master menu.

The options are:

- ☐ Medical and General Stores
- ☐ Suppliers Detail
- ☐ Exit

Pharmacy Database Management System

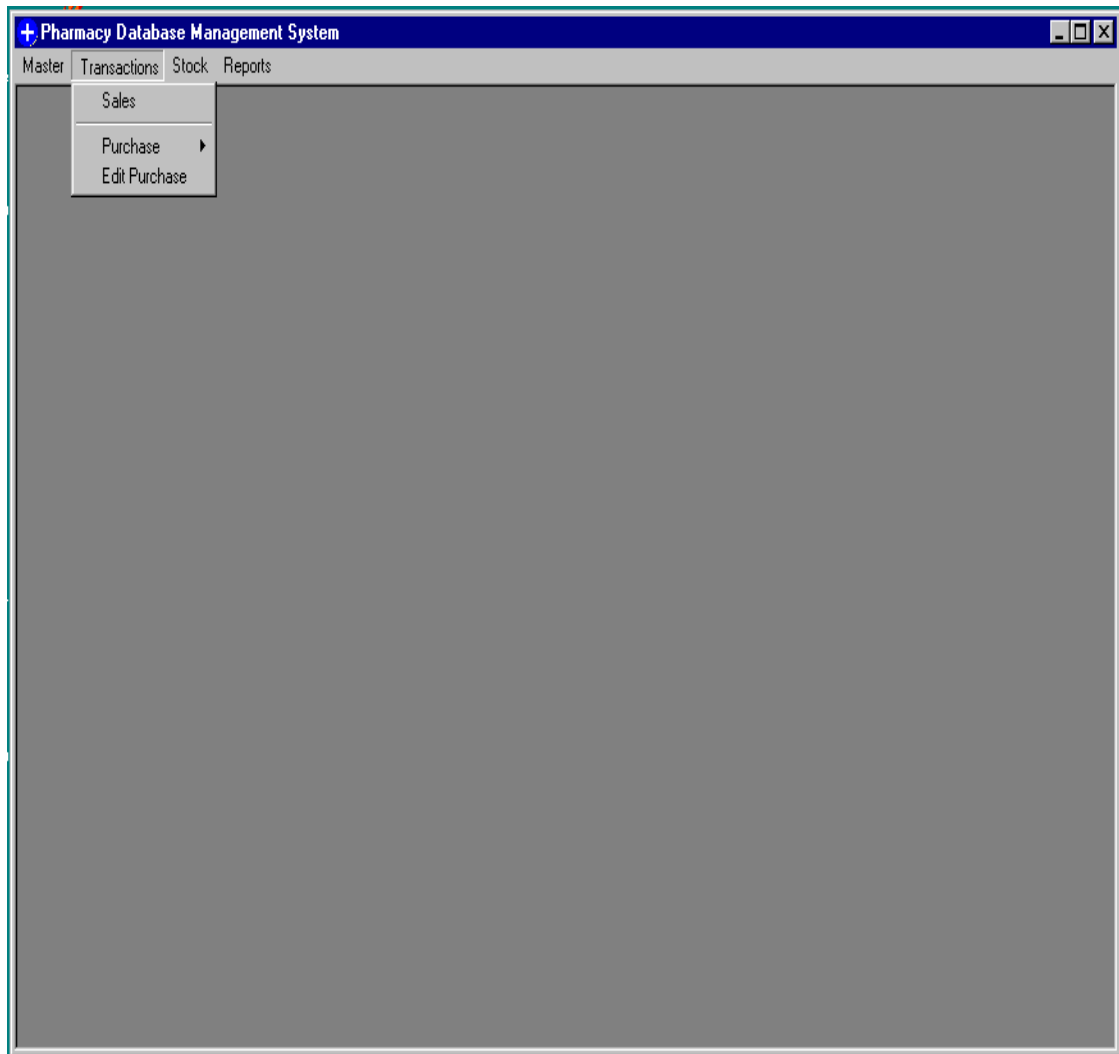


Fig. Transactions Menu

The above screen displays the options under the Transaction menu.

The options are:

- ☐ Sales
- ☐ Purchase
 - ☐ Medicine
 - ☐ General Stores
- ☐ Edit Purchase

Pharmacy Database Management System

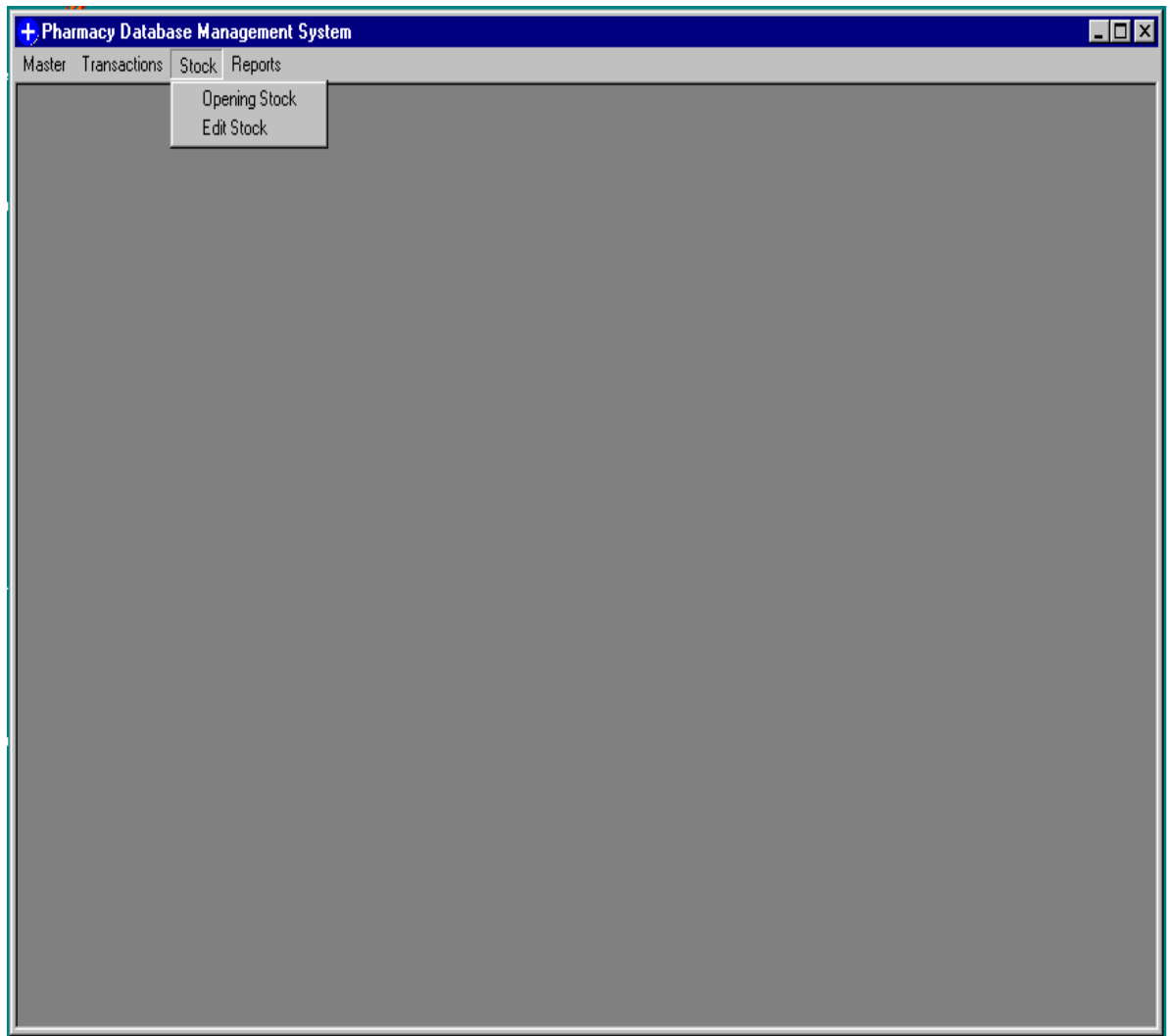


Fig. Stock Menu

The above screen displays the options under the Stock menu.

The options are:

- ☐ Opening Stock
- ☐ Edit Stock

Pharmacy Database Management System

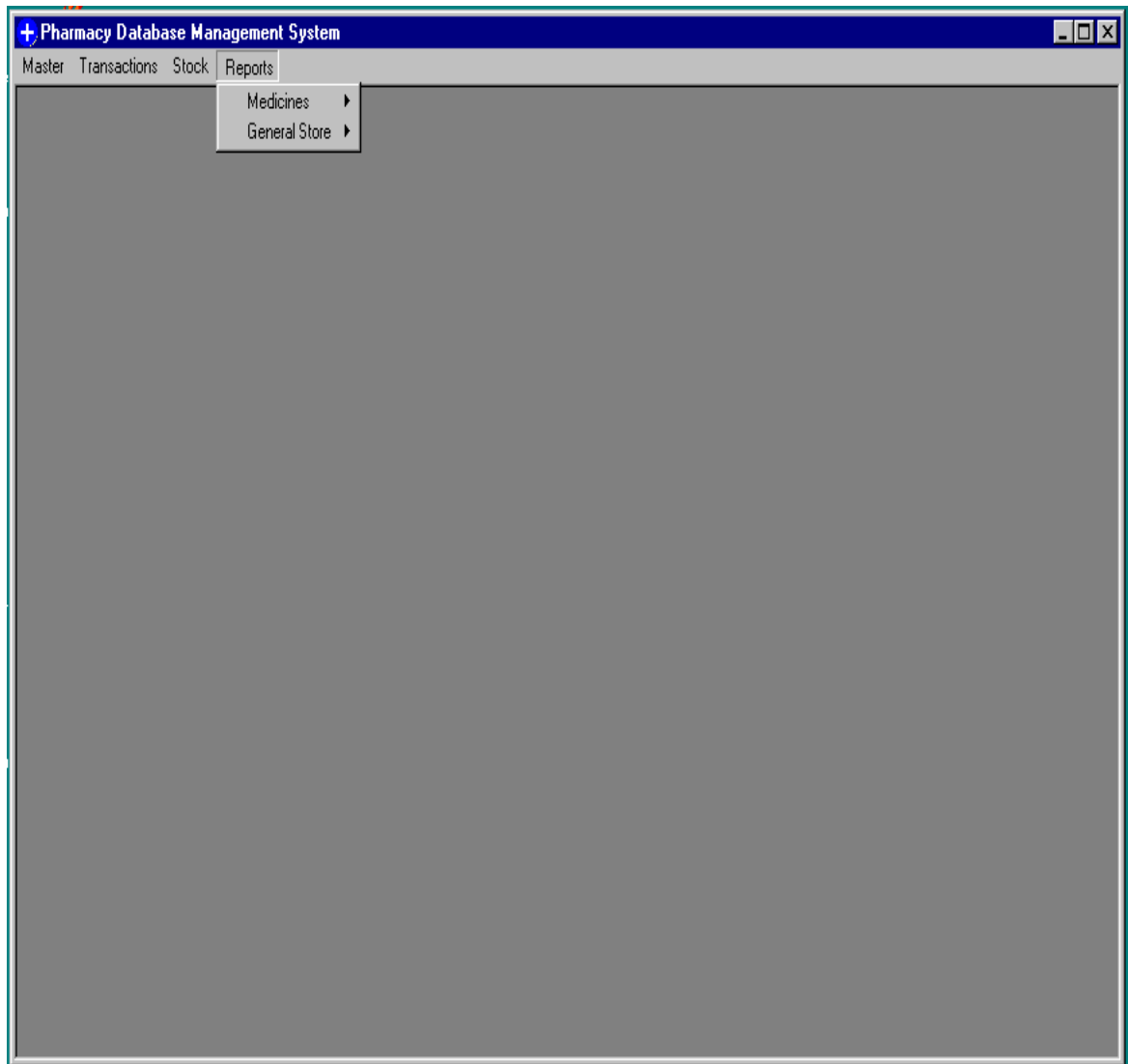


Fig. Report Menu

The above screen displays the Report menu. It consists of reports for the Medical department and General Stores department.

The Medical department reports consists of:

- ☐ Daily Sales
- ☐ Monthly Sales
- ☐ Current Stock
- ☐ Medicine Code List

The General Stores department consists of:

Pharmacy Database Management System

- Daily Sales
- Monthly Sales
- Current Stock
- Product Code List

Pharmacy Database Management System

The screenshot displays the 'Pharmacy Database Management System' window. At the top, there is a menu bar with 'Master', 'Transactions', 'Stock', and 'Reports'. Below the menu bar, a 'Select Action' section contains two radio buttons: 'Medicine' (selected) and 'General Store'. The 'General Store' section includes input fields for 'Code', 'Name', 'Type', and 'Minimum stock Level'. The 'Medicine' section includes input fields for 'Code', 'Name', 'Type' (a dropdown menu), 'Minimum Stock Level', and a 'Contents' text area. To the right of these sections is a 'Commands' panel with buttons for 'CLEAR', 'ADD NEW', 'SAVE', 'PREVIOUS', 'NEXT', 'EDIT', 'DELETE', and a button with a hand icon.

Pharmacy Database Management System

Master Transactions Stock Reports

Select Action

☒ Medicine ☐ General Store

General Store

Code:

Name:

Type: Minimum stock Level:

Medicine

Code:

Name:

Type: Contents:

Minimum Stock Level:

Commands

CLEAR

ADD NEW SAVE

PREVIOUS NEXT

EDIT DELETE

Fig. Medicine and General Stores

Pharmacy Database Management System

The screenshot displays the 'Pharmacy Database Management System' window. At the top, there is a menu bar with 'Master', 'Transactions', 'Stock', and 'Reports'. Below the menu bar, there is a 'Select Action' section with two radio buttons: 'Medicine' (selected) and 'General Store'. The 'General Store' section contains a 'Code' field with the value '1', a 'Name' field with the value 'APPY', a 'Type' dropdown menu with the value 'DRINK', and a 'Minimum stock Level' field with the value '10'. The 'Medicine' section contains a 'Code' field with the value '1', a 'Name' field with the value 'PARACETAMOL', a 'Type' dropdown menu with the value 'Tab', a 'Contents' text area with the value 'PARACETAMO L.I.P.500MG', and a 'Minimum Stock Level' field with the value '25'. To the right of the 'Medicine' section, there is a 'Commands' section with buttons for 'CLEAR', 'ADD NEW', 'SAVE', 'PREVIOUS', 'NEXT', 'EDIT', and 'DELETE'. A mouse cursor is pointing at the 'DELETE' button.

Fig. Medicine and General Stores with data

The Medical and General stores module basically deals with the information of medicine and general store products. The minimum stock level can be set here as shown in above figure. The minimum stock level will help the user to be warned when the stock level falls below the specified value.

In this module the product ID i.e. code for each section is automatically generated. This is done when the user clicks on ADD NEW button.

The navigation buttons provided helps the user to navigate through the records. The buttons are PREVIOUS and NEXT. The option to edit, delete and save are also given so that the user can modify the records.

The screenshot displays a software window titled "Pharmacy Database Management System" with a menu bar containing "Master", "Transactions", "Stock", and "Reports". The main area is titled "SUPPLIER'S INFORMATION". Below this, there is a "Details" section with several input fields: "Supplier's" (containing "1"), "Name:" (containing "JYOTI PHARMACEUTICALS LIMITED"), "Address:" (containing "PLOT 2, GIDC INDUSTRIAL ZONE KALOL"), "City:" (containing "BARODA"), "Pin:" (containing "390002"), "Tel. No.:" (containing "95-269-45652"), "Fax No.:" (containing "95-269-45123"), and "E-mail:". To the right of these fields is a "Commands" section with buttons for "CLEAR", "ADD NEW", "SAVE", "PREVIOUS", "NEXT", "EDIT", "DELETE", and a button with a hand icon.

Fig. Supplier's Information

The Supplier's module deals with all the supplier's information. This information will be used when we want to check which company supplies what products. The basic information such as address, telephone number, etc.

Whenever we add new supplier's information by clicking on ADD NEW button the suppliers ID is generated automatically.

The suppliers ID textbox is locked so that the user will not enter an invalid code.

Pharmacy Database Management System

Pharmacy Database Management System
Master Transactions Stock Reports

Type Of Sales
☐ Medicines ☐ Gen. Store

Mode Of Sales
☐ Cash Sale ☐ Credit Sale

Cash Balances
Today: Total:

Patient's Details

Sales Date:

Customer's Name Locality Ref By Dr

Product: Batch No.:

Code	Product Name	Type	Batch	Exp.Date	Stock	Qty	Price	Total

Stock Level
Current Stock:
Reorder Level:

Supplier/Manufacturer
Supplier

Manufacturer

Cash Memo./Invoice Details
Total Sales:
V.A.T %:
Total Amount:

CLEAR ADD NEW SAVE PRINT 

Fig. Sales Module

Pharmacy Database Management System

The screenshot shows the 'Sales Module' of the Pharmacy Database Management System. The interface includes several sections for data entry and display:

- Type Of Sales:** Radio buttons for 'Medicines' (selected) and 'Gen. Store'.
- Mode Of Sales:** Radio buttons for 'Cash Sale' (selected) and 'Credit Sale'.
- Cash Balances:** Fields for 'Today' and 'Total' (1,993.00).
- Patient's Details:** Fields for 'Cash Memo' (1), 'Sales Date' (03/01/2006), 'Customer's Name' (PETER JOHNSON), 'Locality' (MUMBAI), 'Ref. By Dr.' (DR.P.R.SOLANKI), 'Product' (CROCIN), and 'Batch No.' (cr-345).
- Table:** A table with 9 columns: Code, Product Name, Type, Batch, Exp.Date, Stock, Qty, Price, and Total. It contains two rows of data.
- Stock Level:** Fields for 'Current Stock' (150) and 'Reorder Level' (20).
- Supplier/Manufacturer:** Fields for 'Supplier' and 'Manufacturer', both showing 'Not Available'.
- Cash Memo./Invoice Details:** Fields for 'Total Sales' (39.00), 'VAT %' (10.20), and 'Total Amount' (43.00).
- Buttons:** CLEAR, ADD NEW, SAVE, PRINT, and a hand icon.

Code	Product Name	Type	Batch	Exp.Date	Stock	Qty	Price	Total
1	PARACETAMOL	Tab	pa-123	29/01/2008	200	15	1.00	15.00
2	CROCIN	Tab	cr-345	29/01/2009	150	30	.80	24.00

Fig. Sales Module with Data

The Sales Module deals with the sales of products. Whether the product is of type Medicine or General Store is dependent on the action selected. The option for cash or credit sales also has been provided for the user.

If the payment is in cash the cash balance for today's sales and total sales will be accordingly updated. If payment is credit then no changes will be reflected on cash balance.

When we click on ADD NEW button the cash memo or credit memo number is generated automatically.

The patient's information is also saved at the same time when the entire detail is saved. This help for keeping the details of the creditors. Along with the patient's information the doctor's information, who refer the patients to the pharmacy is also kept.

Pharmacy Database Management System

The module also gives the information about the current stock level and the minimum stock level. If the stock level falls below the minimum value it will display a critical message, warning the user that the stock quantity level is low.

We click on SAVE to save the information and then click on PRINT to print the receipt.

Pharmacy Database Management System

[illegible]

Fig. Medicine Purchase Module

[illegible]

Fig. Medicine Purchase Module with Data

The Medicine Purchase module deals with the purchase of the medicine from the suppliers. These medicines are then updated in the stock table.

Note that the bill number and bill date is left blank as shown on the figure above.

There are cases when the bill is sent later than the delivery of the products.

In the above figure, user enters the supplier's code and the supplier's name automatically is displayed.

[illegible]

Fig. General Store's Purchase Module

General Store's Purchase

Invoice Details

Supplier's Code: 3 Supplier's Name: SAINATH

Challan No.: 5436 Challan Date: 09/01/2006 Bill No.: B-123 Bill Date: 09/01/2006

Select Product: APPY Invoice Total: 10.00

Product Name	Packing	Quantity	Cost	Manufactured By	Ur
APPY	10	100	10.00	COCA COLA	

Commands

CLEAR NEW SAVE

Fig. General Store's Purchase Module with Data

The General Store's Purchase module deals with the purchase of the general store products from the suppliers. These products are then updated in the stock table.

the bill number and bill date may be left blank in the figure above. There are cases when the bill is sent later than the delivery of the products.

In the above figure, user enters the supplier's code and the supplier's name automatically is displayed.

Pharmacy Database Management System

Pharmacy Database Management System

Master Transactions Stock Reports

Select Type Of Product

☐ Medicines ☐ General Stores

Search

Challan No Challan Date **Search**

Edit

Sup.Code Supplier's Name

Challan No Challan Date Bill No Bill Date

Product Code Product Name Type

Packing Quantity Cost Batch Exp.Date

Mfg.Date Manufacturer's Name

Commands

Clear

Next

Previous

Save

Exit

Fig. Edit Purchase Module

Pharmacy Database Management System

Pharmacy Database Management System

Master Transactions Stock Reports

Select Type Of Product

☒ Medicines ☐ General Stores

Search

Challan No 123 Challan Date 09/01/2006 Search

Edit

Sup.Code 1 Supplier's Name JYOTI PHARMACEUTICALS LIMITED

Challan No 123 Challan Date 09/01/2006 Bill No B-42 Bill Date 10/01/2005

Product Code 3 Product Name BENADRYL Type Syrup

Packing 6 Quantity 36 Cost 1620 Batch BA-768 Exp.Date 12/02/2009

Mfg.Date 12/02/2005 Manufacuter's Name EMCURE PHARMACEUTICALS LTD

Commands

Clear

Next

Previous

Save

Exit

Fig. Edit Purchase Module with Data

The Edit Purchase Module is for checking whether the entries made were correct or not. The user can search for particular record by entering the challan number and the date. The navigation buttons have also been provided to move through the records. The user can then edit the particular record and then save it.

Pharmacy Database Management System

Pharmacy Database Management System

Master Transactions Stock Reports

Select Action

☐ Medicine ☐ General Store

Opening Stock Of General Store

Code: Opening Date:

Name:

Type:

Quantity: Selling Price:

Opening Stock of Medicine

Code: Opening Date:

Name:

Type:

Batch No.: Expiry Date:

Quantity: Selling Price:

Commands

CLEAR

ADD NEW SAVE

PREVIOUS NEXT

EDIT

Fig. Opening Stock Module

The screenshot displays the 'Pharmacy Database Management System' window. At the top, there are menu items: Master, Transactions, Stock, and Reports. Below the menu is a 'Select Action' section with two radio buttons: 'Medicine' (selected) and 'General Store'. The main area is divided into two sections: 'Opening Stock Of General Store' and 'Opening Stock of Medicine'. The 'Opening Stock of Medicine' section contains the following data:

Code	Opening Date	Name	Type	Batch No.	Expiry Date	Quantity	Selling Price
1	25/12/2005	PARACETAMOL	Tab	pa-123	29/01/2008	200	1.00

On the right side, there is a 'Commands' section with buttons: CLEAR, ADD NEW, SAVE, PREVIOUS, NEXT, EDIT, and a button with a hand icon.

Fig. Opening Stock Module with Data

The Opening Stock Module deals with the initial stock entry when the system will be implemented for the first time.

This module takes the opening stock of the medicine and the general store products. The option button is provided for this purpose. When selecting the respective option the records that are available are displayed.

Then navigation buttons are also provided for moving through the records. And accordingly edit the record.

Pharmacy Database Management System

Pharmacy Database Management System

Master Transactions Stock Reports

Select Action

☐ Medicines ☐ General Stores

Current Stock Of General Stores

Code Date

Name

Type

Quantity Selling Price

Current Stock Of Medicines

Code Date

Name

Type

Batch No Exp.Date

Quantity Selling Price

Commands

CLEAR

SEARCH

FIRST LAST

PREVIOUS NEXT

EDIT

EXIT

Fig. Edit Stock Module

Pharmacy Database Management System
Master Transactions Stock Reports

Select Action
☒ Medicines ☐ General Stores

Current Stock Of General Stores

Code Date
 Name
 Type
 Quantity Selling Price

Current Stock Of Medicines

Code Date
 Name
 Type
 Batch No Exp.Date
 Quantity Selling Price

Commands

CLEAR
 SEARCH
 FIRST LAST
 PREVIOUS NEXT
 EDIT
 EXIT

Fig. Edit Stock Module with Data

This module deals with the new entries of the stock. Both the medicine and general store products are available here. We can search for a particular stock by entering their respective code or product ID.

Database structure

The tables used are as follows:

Pharmacy Database Management System

- MediList – keeps medicine information.
- GenList – keeps general store product information.
- SupList – keeps the supplier's information.
- PurchReg – purchase register, which store information on purchase of goods for pharmacy.
- SalesReg – sales register, which store information on sales made.
- MediStock – keeps the medicine stock record.
- GenStock – keeps the general store product's stock records.
- MedOpStock – keeps opening stock records for medicine.
- GenOpStock – keeps opening stock record for products.
- CustInfo – keeps customer information.
- Cash – keeps records of cash sales.
- DocDetail – keeps doctors information.

2.6 Coding Details

The Standard Controls, Forms, and Menus

Properties

The properties describe the appearance of the GUI component. When adding a component, the Name property should be set immediately, according to the three-letter mnemonic naming conventions. The properties are displayed in the Properties Window in Name/Value pairs in alphabetical order.

Event Procedures

An event procedure is a piece of code that responds to events that can occur for that object. Most of the events are generated by the user, enabling them to dictate the order of execution.

Forms

The Form is the main stage of your application. By default, the Standard Exe option starts with a form called "Form1". The Name property of the Form should be named with a three-letter mnemonic prefix of "frm". Each Form will be a Window in your application. Controls are added to the form by either double-clicking them in the toolbox, or by selecting the control and drawing a bounding rectangle on the form. Your application may use more than one form.

To add a new Form to the project, either select "Add Form" from the "Project" menu or right-click the Forms folder in the Project Explorer and select, "Add", and then "Form".

To load a new form, use the Show method. The parameter, vbModal, is optional. If used, vbModal means that the form has focus until closed within the application.

Standard Controls

Controls are added to the Form from the Toolbox. Each control has a set of properties, and a set of event procedures associated with it. The following lists the control, reading left to right, top to bottom as they appear in the standard Toolbox.

When coding the system we used a more Advanced ActiveX Control call **MSFlexGrid** Control.

MSFlexGrid control:

One of the most impressive controls of Visual Basic is the MSFlexGrid control. MSFlexGrid control provides all the functionality for building spreadsheet applications, word processing applications. The MSFlexGrid control is an extremely useful tool for displaying information in a tabular form.

You can add a flex grid to a Visual Basic project easily; just follow these steps:

1. Select the Project [vbar] Components menu item.
2. Click the Controls tab in the Components dialog box.
3. Select the Microsoft FlexGrid Control entry in the Components dialog box.
4. Close the Components dialog box by clicking on OK. This displays the Flex Grid Control tool in the toolbox.
5. Add a flex grid control to your form in the usual way for Visual Basic controls, using the Flex Grid Control tool.
6. Set the flex grid's **Rows** and **Cols** properties to the number of rows and columns you want in your flex grid. You can also customize your flex grid by setting such properties as **BorderStyle**, **ForeColor**, **BackColor**, and so on.

Several flex grid properties will help us here:

- **Row**—The current row in a flex grid
- **Col**—The current column in a flex grid
- **Rows**—The total number of rows
- **Cols**—The total number of columns

□ **Text**—The text in the cell at (**Row**, **Col**)

The screenshot displays the 'Pharmacy Database Management System' window. It features a menu bar with 'Master', 'Transactions', 'Stock', and 'Reports'. The main interface is divided into several sections:

- Type Of Sales:** Radio buttons for 'Medicines' and 'Gen. Store'.
- Mode Of Sales:** Radio buttons for 'Cash Sale' and 'Credit Sale'.
- Cash Balances:** Fields for 'Today:' and 'Total:'.
- Patient's Details:** Fields for 'Sales Date:', 'Customer's Name', 'Locality', 'Ref By Dr', 'Product', and 'Batch No.'.
- Data Grid:** A table with columns: Code, Product Name, Type, Batch, Exp.Date, Stock, Qty, Price, and Total.
- Stock Level:** Fields for 'Current Stock:' and 'Reorder Level:'.
- Supplier/Manufacturer:** Fields for 'Supplier' and 'Manufacturer'.
- Cash Memo./Invoice Details:** Fields for 'Total Sales:', 'VAT %:', and 'Total Amount:'.

At the bottom, there are buttons for 'CLEAR', 'ADD NEW', 'SAVE', 'PRINT', and a hand icon.

Fig. MSFlexGrid Control

Data Entry

The MSFlexGrid control is an extremely useful tool for displaying data, but it lacks a basic compatibility: the user cannot edit single cell.

There are two approaches to editing a grid's cell. The first is to place a TextBox control on the form. Each time the user clicks on a cell, the programs copies the contents of the active cell to the TextBox control and lets the user edit it. The second approach is a bit more elegant, but it involves more complex coding: if u place a TextBox control with the exact same dimensions of the cell being edited right on top of the cell, the user gets the impression of editing a cell directly on the grid.

To integrate the TextBox control with the MSFlexGrid control, use the **CellWidth**, **CellHeight**, **CellTop** and **CellLeft** properties of the grid, which determine the current cell's dimensions and placement on the grid.

After the text control is placed exactly on top of the cell, the contents of the current cell are copied to the TextBox. When the user moves to another cell by clicking it, the TextBox's contents are copied to that cell, and then the TextBox control is placed over it.

MSFlexGrid cells support formatting, including word wrap. You can format text using these properties of flex grids:

- **CellFontBold**
- **CellFontItalic**
- **CellFontName**
- **CellFontUnderline**
- **CellFontStrikethrough**
- **CellFontSize**

2.7 System Testing

Testing of the software as a mean of accessing or measuring the software to determine its quality. The area of testing is one of the key process areas in ensuring the quality of the software known as Software Quality Assurance (SQA).

Testing is done with one primary objective to ensure the quality of the software before it is actually implemented. The main purpose of testing from developer's point of view is to gain confidence. If no error are found at least he is sure that the product under development is meeting it required goals in terms of quality.

There is no way to find when to stop testing however people have followed certain norms and guidelines over the course of time. They are as follows:

- ☐ When the rate of finding errors has reached an acceptable level.
- ☐ Based on the size of the system, "enough" errors have been found where "enough" can be a quantitative parameter based on historical data.
- ☐ Measuring coverage of testing as percentage of line executed divided by total number of lines in the system. If the percentage is below acceptable limits, then the test plans have to be enhanced to test areas where code not been tested.

Validation Testing

In validation testing we ensure that all the data entered was displayed properly on the view screen and that we got the expected results.

Pharmacy Database Management System

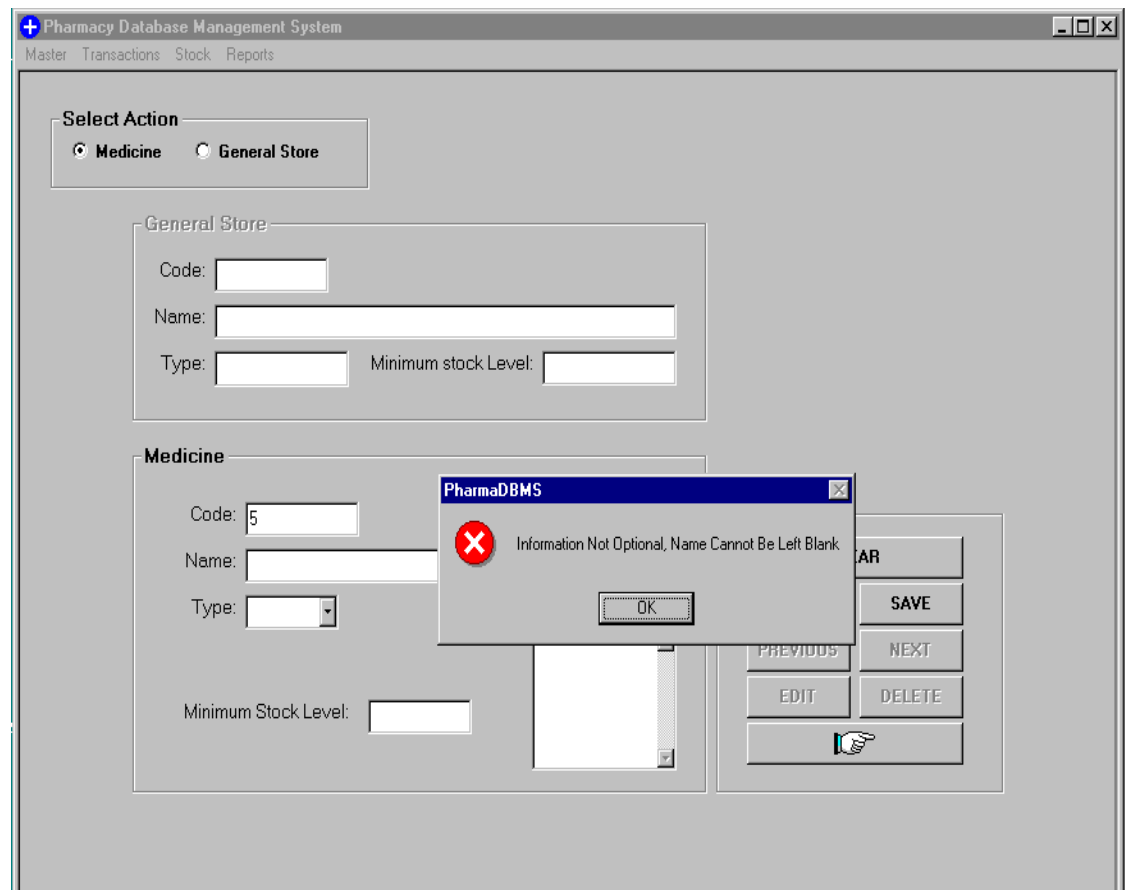


Fig. Critical Error Message

In this way all the screens were checked again and again to ensure proper validation.

Integration Testing

The last step was to integrate the various modules in the software. We implemented top-down integration. The integration was performed in five steps:

- ☐ The main control module was used as test driver and stubs were substituted for all components directly subordinate to the main control module.
- ☐ Depending on depth first search subordinate stubs was replaced with actual components.
- ☐ Tests were conducted as each component with real component.
- ☐ On completion of each set of tests, another stub is replace with real component.
- ☐ Integration testing was conducted to ensure execution of some subset of tests have already been conducted had not propagated unintended side effects.

Chapter 3

WHY?

3.1 Why Visual Basic?

Visual Basic provides more of the actual code for a programmer than any other non-visual programming language. This makes it a widely used programming language for visual applications.

If you have ever programmed in the older BASIC or other command line programming language, then you will remember that the programmer had to write the code for entire user interface. Today's windows, buttons, lists and other application features such as menus were not built-in to the BASIC programming language. Programmers had to create the code for these features on their own.

As much as 80% of a programmer's time was spent writing code to create the user interface to his applications (the visual interface). To eliminate this huge drain on a programmer's time, Microsoft had provided Visual Basic with built-in capability to create the user interface using nothing more than a mouse.

Visual Basic is itself a window application. You load and execute the VB system just as you do other Windows programs; we use this running VB program to create other programs. VB is just a tool, albeit an extremely powerful tool, that programmers use to write, test, and run windows applications.

Visual Basic (VB) is an event driven programming language and associated development environment created by Microsoft. In business programming, it has one of the largest user bases.

It is derived heavily from BASIC and enables rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using DAO, RDO, or ADO, and creation of ActiveX controls and objects. A programmer can put together an application using the components provided with Visual Basic itself.

This built-in interface creation capability has had the future benefit of standardizing on the user interface to Windows applications. Today, user can move from one window program to another and see the same basic interface tools to work with- allowing them to concentrate solely on the unique capabilities of the application.

The bottom line is that you can create an entire application shell (the user interface) very quickly and then spend most of your time working on the features, which differentiate your application from its competitor.

Today you need much more than just a language; you need a graphical development tool that can work inside the window system and applications that take advantage of all the graphical, multimedia, online and multiprocessor activities that windows offer.

Visual Basic is such a tool. More than a language, Visual Basic lets you generate applications that interact with every aspect of today's windows operating systems.

Language feature

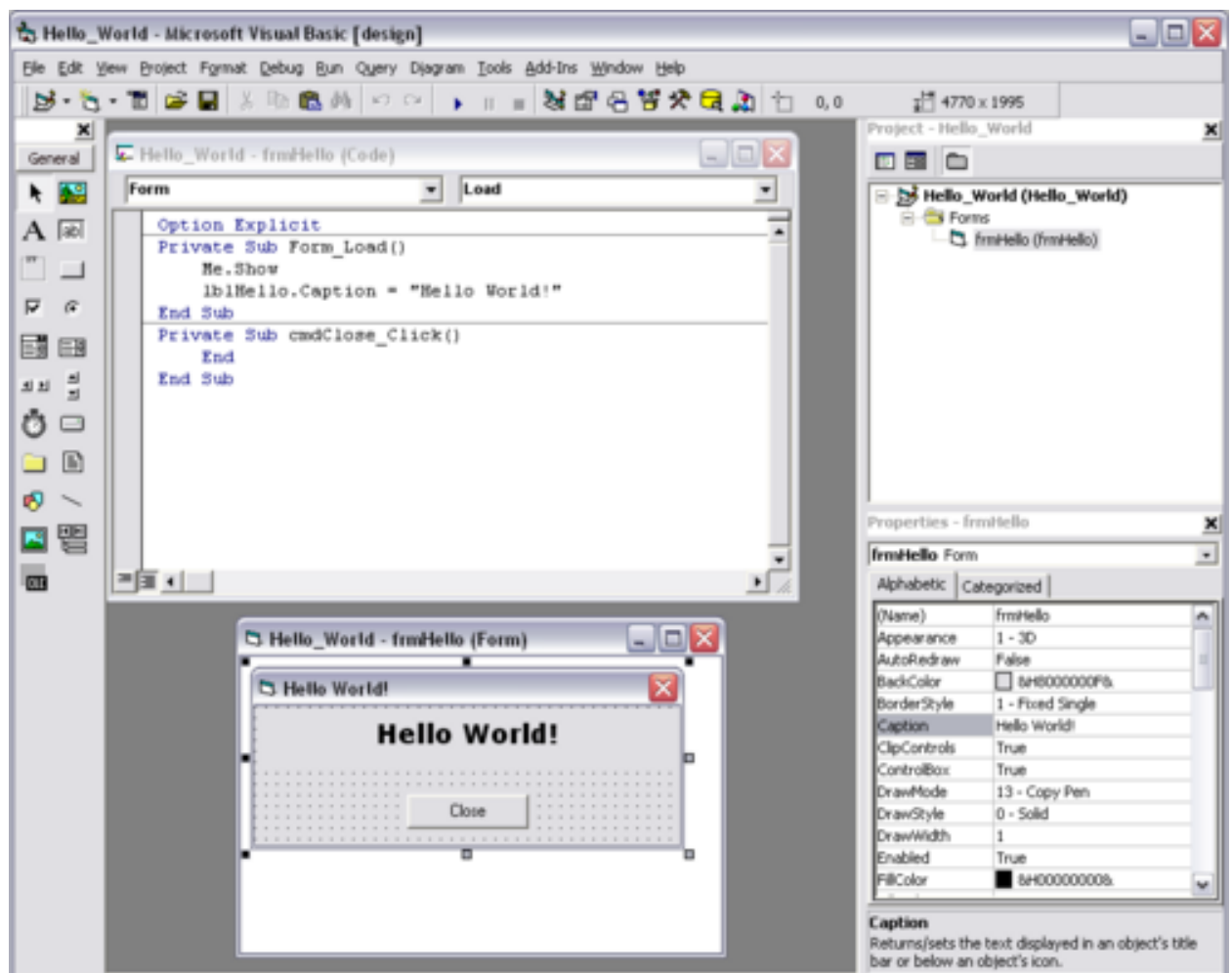


Fig. VB Integrated Development Environment

A typical Session in Microsoft Visual Basic 6

Visual Basic was designed to be usable by all programmers, whether novice or expert. The language is designed to make it easy to create simple GUI applications, but also has the flexibility to develop fairly complex applications as well. Programming in VB is a combination of visually arranging components on a form, specifying attributes and actions of those components, and writing additional lines of code for more functionality. Since default attributes and actions are defined for the components, a simple program can be created without the programmer having to write many lines of code. Performance problems were experienced by earlier versions, but with faster computers and native code compilation this has become less of an issue.

Although programs can be compiled into native code executables from version 5 onwards, they still require the presence of runtime libraries of approximately 2 MB in size. This runtime is included by default in Windows 2000 and later, but for earlier versions of Windows it must be distributed together with the executable.

Forms are created using drag and drop techniques. A tools palette is used to place controls (e.g., text boxes, buttons, etc.) on the form (window). Controls have attributes and event handlers associated with them. Default values are provided when the control is created, but may be changed by the programmer. Many attribute values can be modified during run time based on user actions or changes in the environment, providing a dynamic application. For example, code can be inserted into the form resize event handler to reposition a control so that it remains centered on the form, expands to fill up the form, etc. By inserting code into the event handler for a keypress in a text box, the program can automatically translate the case of the text being entered, or even prevent certain characters from being inserted.

A Visual Basic application can consist of one or more windows, or a single window that contains MDI child windows, as provided by the operating system. Dialog boxes with less functionality (e.g., no maximize/minimize control) can be used to provide pop-up capabilities. Controls provide the basic functionality of the application, while programmers can insert additional logic within the appropriate event handlers. For example, a drop-down combination box will automatically display its list and allow the user to select any element. An event handler is called when an

item is selected, which can then execute additional code created by the programmer to perform some action based on which element was selected, such as populating a related list.

Alternatively, a Visual Basic component can have no user interface, and instead provide ActiveX objects to other programs via Component Object Model (COM). This allows for server-side processing or an add-in model.

The language is garbage collected using reference counting, has a large library of utility objects, and has basic object oriented support. Since the more common components are included in the default project template, the programmer seldom needs to specify additional libraries. Unlike many other programming languages, Visual Basic is generally not case sensitive, although it will transform keywords into a standard case configuration and force the case of variable names to conform to the case of the entry within the symbol table entry. String comparisons are case sensitive by default, but can be made case insensitive if so desired.

Visual Basic was designed to meet all real world requirements with its key features, which are explained as follows: -

1. Event Driven Programming Language: -

Visual Basic makes it easy to locate event procedures code for controls on forms. Double-click any control to see one of its event procedures. For example, if you double-click the Exit command button, Visual Basic opens window and places the text cursor in the set of lines. The event procedures code, however, doesn't do anything until runtime.

2. Graphical User Interface (G.U.I): -

In graphical user interface user can work with several windows at same time, within each window user can work on specific task. User can overlap windows, hide, change size and locations and even shrink down to icons.

Using mouse user can move from one window to another, the window which is selected is an active window, most GUI are based on x-window system, x-window is a system used for supporting GUI.

3. Database Connectivity: -

Some databases, such as Microsoft Access, store all the related database files

in a single global file called the database file. Inside the database, the individual groups of records and fields are called *tables*. Other database systems, such as dBase, keep track of a database's data in multiple files. When you use database such as Microsoft Access, you must describe both the overall database and the individual table name within the database that the Data control is in use.

4. Controls: -

Controls are tools on the Toolbox window that you place on a form to interact with the user and control the program flow. The controls can be self generated and can be used again and again in the same application. As you add controls to the Form window, the properties window updates to show the properties for the currently selected control. The selected control is usually the control you last placed on the form. Visual Basic lets you see a control's properties in the Properties window by clicking to select the control or by selecting the control from the properties window's drop-down list box.

Benefits of Using VISUAL BASIC 6.0

- VB now supports a true compiler that creates standalone runtime .exe files that execute more quickly than previous VB programs
- VB also includes several wizards that offer step-by-step dialog box questions that guide you through the creation of applications.
- VB's development platform, a development environment called the Developer Studio.
- Now supports the same features as the advanced Visual C++ and Visual J++ Compilers.
- After you learn one of Microsoft's visual programming products, you will have the skills to use the other language products without a long learning curve ahead of you.
- Visual Basic lets you generate applications that interact with every aspect of today's Windows operating systems.

Pharmacy Database Management System

- VB's programming language is fairly simple and uses common English words and phrases for the most part.
- Microsoft Visual Basic 6.0, the latest and greatest incarnation of the old language, gives you a complete Windows application development system in one package.
- Visual Basic lets you write, edit, and test Windows applications. VB includes tools that can be used to write and compile help files, ActiveX controls and even Internet applications.

3.2 Why Microsoft Access

Access is widely used by small businesses and hobby programmers to create ad hoc customized systems for handling small tasks. Its ease of use and powerful design tools give the non-professional programmer a lot of power for little effort. However, this ease of use can be misleading. This sort of developer is often an office worker with little or no training in application or data design. Because Access makes it possible even for such developers to create usable systems, many are misled into thinking that the tool itself is limited to such applications.

Some professional application developers use Access for rapid application development, especially for the creation of prototypes and standalone applications that serve as tools for on-the-road salesmen. Access does not scale well if data access is via a network, so applications that are used by more than a handful of people tend to rely on a Client-Server based solution such as Oracle, DB2, Microsoft SQL Server, PostgreSQL, MySQL, or MaxDB. However, an Access "front end" (the forms, reports, queries and VB code) can be used against a host of database backends, including Access itself, SQL Server, Oracle, and any other ODBC-compliant product. This approach allows the developer to move a matured application's data to a more powerful server without sacrificing the development already in place.

Features

One of the benefits of Access from a programmer's perspective is its relative compatibility with SQL – queries may be viewed and edited as SQL statements, and SQL statements can be used directly in Macros and VBA Modules to manipulate Access tables. Users may mix and use both VBA and "Macros" for programming forms and logic and offers object-oriented possibilities.

The report writer in Access is similar to the other popular database report writer – Crystal Reports – but the two products are vastly different in their approach. MSDE (Microsoft SQL Server Desktop Engine) 2000, a mini-version of MS SQL Server 2000, is included with the developer edition of Office XP and may be used

with Access as an alternative to the Jet Database Engine. (*Early versions of MSDE and Microsoft Exchange Server actually use the Jet engine to handle huge volumes of data and placed a "fake" application layer for those applications on top of it. Lack of knowledge about this fact has contributed to an undeserved disrespect for Access/Jet family of software products, particularly as regards "large" projects.)

Access' cut and paste functionality can make it a useful tool for connecting between other databases (for example, Oracle and Microsoft SQL Server during data or database conversions. Access comes with various import and export features that allow integration with Windows and other platform applications, several of which can be executed on demand from within applications or manually by the user. For example the very compact SNP format for sharing perfectly formatted reports with people who don't have the full Access software. It can also easily be upgraded to Microsoft SQL Server.

Unlike complete RDBMSes, it lacks database triggers and stored procedures. It does allow forms to contain code that is triggered as changes are made to the underlying table, and it is common to use pass-through queries and other techniques in Access to run stored procedures in RDBMSs that support these.

Development

The programming language available in Access is, as in other products of the Microsoft Office suite, Microsoft Visual Basic for Applications. Two database access libraries of COM components are provided: the legacy Data Access Objects (DAO), only available with Access, and the new ActiveX Data Objects (ADO).

Microsoft Access is easily applied to small projects but scales inefficiently to large projects if applications are designed poorly.

All database queries, forms, and reports are stored in the database, and in keeping with the ideals of the relational model, there is no possibility of making a physically structured hierarchy with them.

One design technique is to divide an Access application between data and programs. One database should contain only tables and relationships, while another

would have all programs, forms, reports and queries, and links to the first database tables. Unfortunately, Access allows no relative paths when linking, so the development environment should have the same path as the production environment (Although you can write your own "dynamic-linker" routine in VBA that can search out a certain back-end file by searching through the directory tree, if it can't find it in the current path).

This technique also allows the developer to divide the application among different files, so some structure is possible.

The Microsoft Jet Database Engine is a database engine on which several Microsoft products were built. A database engine is the underlying component of a database, a collection of information stored on a computer in a systematic way. The first version of Jet was developed in 1992, consisting of three modules which could be used to manipulate a database.

JET stands for *Joint Engine Technology*, sometimes being referred to as *Microsoft JET Engine* or simply *Jet*. Microsoft Access, Microsoft Exchange Server and Visual Basic use or have used Jet as their underlying database engine. It has since been superseded, however, by Microsoft Desktop Engine (MSDE) and no longer exists as a component of Microsoft Data Access Components (MDAC). Jet databases can be upgraded (or in Microsoft parlance, "up-sized") to an MSDE database.

Security

Access to Jet databases is done on a per user-level. The user information is kept in a separate system database, and access is controlled on each object in the system (for instance by table or by query). In Jet 4, Microsoft implemented functionality that allowed database administrators to set security via the SQL commands CREATE, ADD, ALTER, DROP USER and DROP GROUP. These commands were a subset of ANSI SQL 92 standard, and they also applied to the GRANT/REVOKE commands. When Jet 2 was released, security could also be set programmatically through DAO.

Queries

Queries are the mechanisms that Jet uses to retrieve data from the database. They can be defined in Microsoft QBE (Query By Example), through the Microsoft Access SQL Window or through Access Basic's Data Access Objects (DAO) language. These are then converted to an SQL SELECT statement. The query is then compiled — this involves parsing the query (involves syntax checking and determining the columns to query in the database table), then converted into an internal Jet query object format, which is then tokenized and organised into a tree like structure. In Jet 3.0 onwards these were then optimised using the Microsoft Rushmore query optimisation technology. The query is then executed and the results passed back to the application or user who requested the data.

Jet passes the data retrieved for the query in a dynaset. This is a set of data that is dynamically linked back to the database. Instead of having the query result stored in a temporary table, where the data cannot be updated directly by the user, the dynaset allows the user to view and update the data contained in the dynaset. Thus, if a university lecturer queried all students who received a distinction in their assignment and found an error in that student's record, they would only need to update the data in the dynaset, which would automatically update the student's database record without the need for them to send a specific update query after storing the query results in a temporary table.

CONCLUSION AND FUTURE SCOPE

Conclusion and Future Scope

- Detailed information gathering has to be done. Without that the purpose for using the software won't be satisfied properly.
- However it can give good profits in the long run.
- Implementing the software requires change in the business practices.
- Efficient organization of all knowledge is the analysis company and easy analysis access and retrieval of information is possible.
- In this project we can also include BAR CODE facility using the bar code reader, which will detect the expiry date and the other information about the related medicines.
- Company using this software will always be able to plan in future and always be aware of their financial position in the market.
- It leads to streamlining of business processes.
- The implementation and maintenance costs run very high (about 2 to 3 % of the company's revenue.)

BIBLIOGRAPHY AND REFERENCES

BIBLIOGRAPHY AND REFERENCES

Bibliography: -

- Evangelos peroutsos: - Mastering Visual Basic 6.0.
- Greg Perry, Sanjaya Hettihewa: - Visual basic 6.0 in 24 hours.

References: -

- <http://www.google.co.in>
- <http://www.wikipedia.com>