Kumaraguru College of Technology

Department of Computer Science and Engineering
Coimbatore – 641006.

September 2004

BUSINESS SOLUTIONS

Project Work done at

P-124C

SANGUINNE GARMENT EXPORT INDIA PVT LTD

PROJECT REPORT

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

M.Sc. Applied Science (Software Engineering)
Bharathiar University, Coimbatore

Submitted by

N.Prasanna Devi
Reg. No. : 0137S0044

Internal Guide: Mrs. V.Geetha M.C.A.,
Dept. of Computer Science & Engineering,
Kumaraguru College of Technology,
Coimbatore.

External Guide: Mr.M.SELVARAJ M.C.A..
New no.40,8th street,
Tatabad, Coimbatore.

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to Bharathiar University)

COIMBATORE – 641006

BONAFIDE CERTIFICATE

This is to certify that this is the Bonafide Project Record Work done by N.Prasanna Devi, Reg.No **0137S0044** in Partial fulfillment for the award of Degree of **M.Sc. [SOFTWARE ENGINEERING]** , during the academic year 2004 – 2005

Prof & HOD

Guide

This Project entitled "BUSINESS SOLUTIONS " is submitted for the VII semester of M.Sc. [SOFTWARE ENGINEERING] for Bharathiar University Project Viva-voce examinations held on ....30.9.04..........

Internal Examiner

External Examiner

GUINNE GARMENT EXPORT INDIA PVT LTD

Importers
Exporters

P- 1240

This is to certify that Ms. N.Prasanna Devi Reg.No.0137S0044 student of IV Msc Software Engineering from Kumaraguru College of Technology, affiliated to Bharathiyar University, Coimbatore has satisfactorily completed the project entitled "BUSINESS SOLUTIONS" from June 2004 to September 2004.

The involvement and continuous sustained efforts put by the student in the project to achieve the results are highly commendable.

Project Manager,

DECLARATION

I hereby declare that the project entitled BUSINESS SOLUTIONS, submitted to Bharathiar University as the project work of M.Sc Applied Science (Software Engineering), is a record of original work done by me under the supervision of **Mr.M.Selvaraj M.C.A,** Manager, Sanguinne garment export India private limited coimbatore and **Mrs.V.Geetha M.C.A**., Senior Lecturer, Dept of Computer Science and Engineering, Kumaraguru College of Technology and this project work has not formed the basis for the award of any Degree/ Diploma/Associate-ship/Fellow-ship or any other similar title to any candidate of any University.

Place: COIMBATORE

V Prasanna Devi

Date: 24.9.04

Signature of the Student

ACKNOWLEGEMENT

I express my sincere gratitude to **Dr. K.K Padmanabhan,** Ph.D., esteemed Principal, Kumaraguru College of Technology for giving me this opportunity to do the project work and providing facilities in the college to make it possible for me to complete my work without difficulties.

I express my profound gratitude to **Prof. S. Thangasamy, B.E (Hons), Ph.D, Head of Department**, Computer Science and Engineering, Kumaraguru College of Technology who motivated me with his valuable ideas and support in doing this project.

I express my deep sense of gratitude and indebtedness to **Mrs. V.Geetha**, **M.C.A**, Senior Lecturer, Kumaraguru College of Technology for his invaluable guidance throughout the length of this project.

I am grateful to **Mr.M.Selvaraj MCA, Manager, Sanguinne garment export India Pvt Ltd.** for giving me an opportunity to do the project in their esteemed organization.

I am greatly privileged to express my deep sense of gratitude for his support and guidance in doing this project.

I also thank my **Beloved Parents** who have been a pillar of support for me in completing this project, my friends and the department teaching and non-teaching staff for their support in completing this project.

SYNOPSIS

INTRODUCTION

The project entitled "BUSINESS SOLUTIONS" has been developed for Sanguinne Garment Export India Pvt Ltd keeps record of all the information about the inventory and accounts of the company. This holds a huge responsibility in maintaining all the details needed to represent a proper structure of the organization.

This includes adding new supplier and customer details, preparing invoice for the goods sold and making new entries of primary invoice details, raising purchase order and sales order. Collection and payment details are send to the accounting department. Each cash transaction is entered to ledger or voucher based on debit or credit basis.

NEED

With this system, it is very efficient for the company to maintain inventory details and desired reports can be easily generated. The personal details of all the suppliers and customers are maintained in the database. This is very essential when work is to be completed at very high speed. The information of the stocks and payment details can be generated immediately and accurately.

TOOLS USED

For the efficiently of the project, Visual Basic version 6.0 is used as front-end tool and ORACLE 8 is used as back end tool. This helps to prove that the system is efficient in providing sufficient information required by the organization.

CONTENTS

PAGE NO

1.INTRODUCTION

1.1 PROJECT OVERVIEW

The Business Solution is a complete Accounts Package. It is an automated account and trading system that manages the daily operations of the business. It helps to analyze and understand the financial status of the company. The different modules in the project are General Ledger, Receipts and payments, Inventory Control.

The **General Ledger** module is the heart of the accounting system. This module is a complete system capable of performing stand-alone operations including entire book keeping. Each cash transaction is made an entry on debit or credit basis.

The **Receipts and Payments** module consists of two options namely Accounts Payable and Accounts Receivable. **Accounts Receivable** option helps to know about the customers from whom amounts are received and by using the **Accounts Payable** option helps to know about the suppliers to whom payments are to be made and assist in making prompt payment at time.

The **Inventory Control** module allows the user to do various updations and modifications to the inventory. Here we can maintain the details about the Suppliers and Customers. The Purchase order can be placed directly to the suppliers during the purchase of stock. The Sales order can also be prepared easily and quickly. Bills for sales and purchase can be generated and the Collections and Issues of cash can be recorded.

1.2 ORGANIZATION PROFILE

Sanguinne Exports was founded in 1997 with an enthusiastic and enterprising attitude. Sanguinne Exports has been one of the leading professional makers of Textile and Organic products in India. Today, Sanguinne Exports has been dedicated to the promotion of Organic Cotton Garments.

Sanguinne Exports is a Tirupur based company and accounts are dealing at Coimbatore in India. It is a joint venture promoted by a textile professional (BIRLA GROUPS) with over 33years experience in the textile industry in India.

2. SYSTEM STUDY AND ANALYSIS

System Study and Analysis involves the process of gathering the information regarding the existing system the details can be gathered. The details can be gathered either by visiting the Client place or by verifying the existing records. By meeting the users who are working in the system, we can easily get the details about the difficulties and disadvantages in the system, the details about the enhancements required in the system.

After gathering the information, the details are verified and a report is prepared based on the available data. The analysis involves identifying the problem area, the solution to be implemented to overcome the problem, identifying major requirements of the user etc. Then the report is given to the user, so that, he can verify whether the report specifies the actual problem he has. If any corrections the report is modified according to that.

2.1 EXISTING SYSTEM

The existing system is maintained using the software like FoxPro and other lower level database.

The main disadvantage of this system is, when there is large number of data in the database it consume a very long time in retrieving the information and also do not represent the information in a user friendly manner. The customer has to wait for a long time and this makes a tedious task during the regular activities of the day-to-day life.

So the existing system has been replaced by the system proposed to have a user-friendly environment and also to have a faster access to the data. Reports can be generated easily.

In order to have a user-friendly environment we use **Visual Basic 6.0** which has attractive forms and other facilities through which the user can have quick access to the information and to respond his customer very quickly.

LIMITATIONS OF THE EXISTING SYSTEM

- The existing system does not provide user-friendly environment.
- The existing system performs slowly when the database is heavy.
- It is not very efficient in storage and retrieval of the data.
- The reports cannot be generated according to the requirements.
- The security aspect is very poor in the existing system and so it lacks accuracy.

2.2 PROPOSED SYSTEM

The main aim of proposed system is to save time and to generate reports. Thus we have used Microsoft's Visual Basic 6.0 as front-end and Oracle8 as the back-end.

It is very easy to store and maintain the information. Processing is also done quickly. The designing of screens are very much easier and user friendly. Data entry forms are properly designed, so that it is very easy for the user to work. Querying the database can be done very efficiently & effectively. Validations are done properly wherever required in the proposed system.

Thus the proposed system involves reducing the workload of the user of the system and to have descriptive information whatever the customer is asking him. So the link has been given and the report can be taken immediately whenever the user need.

Updations, modifications and deletions can be done at any time. User name and password facilities provide security of data.

OBJECTIVES OF PROPOSED SYSTEM

- Easy to store and maintain the information.
- Processing can be done quickly.
- Less time consumption and work load.
- Reports can be generated.
- Updations, modifications and deletions can be done at any time.
- Control over unauthorized access of data.

3.PROGRAMMING ENVIRONMENT

This system has been developed using the following

3.1. HARDWARE CONFIGURATION

Processor	: Pentium III or Celeron
RAM Capacity	: 64 MB
Hard Disk	: 10 GB
Floppy Disk Drive	: 1.44 MB
Mouse	: 2 Button
Keyboard	: 104 Keys
Monitor	: Color Monitor

3.2.DESCRIPTION OF SOFTWARE AND TOOLS USED

Operating system	:	Windows 98
Front-end	:	Visual Basic 6.0
Back-end	:	ORACLE 8

INTRODUCTION TO THE SOFTWARE

VISUAL BASIC 6.0

Microsoft Corporation developed Visual Basic. In the name Visual Basic the word "**Visual**" refers to the method used to create the Graphical User Interface (GUI). Rather than writing numerous lines of code to describe the location and appearance of interface elements, pre-built objects are added into place on screen.

The word "**Basic**" refers to the BASIC (Beginners All Purpose Symbolic Instruction Code) language. Visual Basic has evolved from the original BASIC language and now contains several statements, functions, and keywords that are related directly to the Windows GUI. Visual Basic uses a Native code Compiler that is very faster than other compilers.

Forms and **controls** are the basic elements in the user interface of any Visual Basic application. These are called as Objects. The main windows are Form Window, Properties Window, Project Explorer Window and Form Layout Window.

Whenever a project is opened or a new project is started, the Visual Basic 6.0 **Integrated Development Environment** (IDE) appears. This IDE contains a group of tools like menus, toolbars and windows. The main windows are Form Window, Properties Window, Project Explorer Window and Form Layout Window.

ADO (ActiveX Data Object): ActiveX Data Objects are used to manipulate database data's and it's structure. In addition like other controls, ADO has default collections of properties. Using ADO we can make use of databases from MS-SQL Server, MS-Access, Oracle, FoxPro and other database engines supported by Microsoft.

ORACLE 8

The Oracle 8 is a state-of-the-art of information management environment. It is a repository for very large amounts of data, and it gives its users rapid access to the data. It allows sharing of data between the applications. The information stored in one place can be used by many systems. It now runs on dozens of different computers supporting the different configurations like Host-based, Client/Sever, Web-Enable Computers.

FEATURES OF ORACLE

- **Data accessibility**: It provides utilities for backing up the data. These include the capabilities to back up the information, while the user community is still using it. It is called "Hot Backup". It also provides data integrity. It also includes full row-level locking of data that resides within it.
- **Procedural capabilities**: It can implement the features of stored procedures, database triggers and packages.
- **Distributed Processing**: In many installations, portions of data reside on different computers. Each location can access the corporate data. It has the facility of local transportation.
- **Security Mechanisms**: Oracle's sophisticated security mechanisms control access to sensitive data through use of an assortment of privileges. Users are given rights to view, modify and create data according to the names they use to connect to the database.

- **Parallel Query**: The parallel query feature allows customers to take advantage of processing queries on computers with more than one Central Processing Unit. When using parallel query on multiple CPU machines, Oracle dispatches a number of query process that work alongside with one another.
- **Oracle Time Series**: This enables efficient storage of time-stamped data. Full analysis time series applications can be developed leveraging the power of the sever technology.
- **Advanced Replication**: Oracle Advanced Replication provides for the storage and distribution of data between the remote sites. Replication is a mechanism where data, stored at one or more sites, is propagated between the sites participating in replication exercise.
- **Visual Information Retrival (VIR)**: Oracle 8i VIR bundles the search capabilities of the server with the power of content-based retrival of VIR.
- **Advanced Networking Option**: this feature provides a single source of integration with the enterprise directory services, such as single sign-on services, network encryption, token and user authentication.

4. SYSTEM DESIGN

System Design plays an important role while developing a new system or enhancing the functions of the existing system. System Design helps us to find perfect steps towards the solution. Designing a model of the required system helps us to determine the problem areas and the steps to be followed to overcome those problems.

Designing a model of the required system also helps us to continue the development without any break as we have found the problem areas and the necessary steps to be followed at design time itself. The system should be designed in some method through which the flow of data and the process steps are understandable.

If any prediction of error, it should be noted down and the assumption and if possible the steps to be followed can also be noted down. This will help us to minimize the occurrence of errors in other parts; also it helps to know the problem and to determine the solution for the problem also.

4.1 INPUT DESIGN

The Input Design is a part of overall system design that requires careful attention. It is the point of contact for the users with the computer system. The input design involves the design of the input screens with Visual Basic 6.0. Using the GUI screens necessary information can be entered and the processing is carried out.

These are very user-friendly and with little training any user can work with this project. The input screens contain the controls like Command buttons, Combo box, List boxes, Tabs, Flex grids, etc.

To make it friendlier several options like VIEW, MODIFY and DELETE options are used. Command buttons like FIRST, LAST, NEXT and PREVIOUS are included in the forms to allow the users to navigate through the records.

The ADD and NEW button executes an automatic code generation to minimize the workload of the user. The SAVE button is used for inserting the records in the appropriate tables in the database. The MODIFY or ALTER button is used for updating the records in the tables. The DELETE button is used to delete a record from the table The EXIT button is used to quit the current form.

The Input Screens included in this project are Customer details, Supplier details. Purchase Order, Sales Order, Quotation details and Invoice details.

4.2 OUTPUT DESIGN

The output should contain all the details that the user requires. The output can be provided in two ways: Screen Output, Reports.

SCREEN OUTPUT

The output screen should be designed clearly so that the required information should be displayed. It should help the user to access the information quickly. To access & retrieve a particular data, a perfect and suitable searching technique should be used based on the type of access of data.

REPORT

The report generated may be simple or brief regarding the type of report, but it should provide the clear information for which the report has been taken. In this project the reports are generated for knowing the current debit and credit amount of the company, invoice details, quotation details, stock details, purchase returns and sales return. This will help the management to know the outstanding amount and the amount received and invested etc.

Some of the sample reports are:

- Secondary Invoice details
- Primary Invoice details
- Customer details
- Supplier details
- Stock details
- Purchase returns
- Sales returns
- Collection details
- Issue details

4.3 DATABASE DESIGN

The database design is a crucial factor in the performance of a system, both in terms of system timings and in the case with which the system can be maintained or modified.

Normalization is a key element of the database design. The tables are designed in such a way that the data are stored without redundancy and allow the data to be retrieved easily. Data redundancy is the elimination of duplicate record.

Database Normalization can be carried out for the following reasons:

- To permit simple retrieval of data in response to the query and report request.
- To simplify the maintenance of the data through updates, insertions and deletions.

The database design is classified into:

- Master
- Transactions

The Master Database consists of Regular entries. The frequently used records are entered into the Master database. Mostly the Master database consists tables like Company_det, Supplier_det, Supplier_item, Collection, Issues.

The Transaction database consists of the tables that are used frequently for executing the transactions of the business. The tables of the Transaction database includes Ledger entry, Voucher entry, Purchase Order, Sales Order, Primary Invoice, Secondary Invoice, Purchase Billing, Sales Billing and Payment Passing.

TABLE DESIGN

LEDGER

This table is used to maintain ledger details.

FIELD NAME	DATATYPE	DESCRIPTION
Comp_name	Varchar2(20)	Company Name
Name	Varchar2(20)	Name of the person
Alias	Varchar2(20)	Purpose
Under	Varchar2(20)	Account names
Mmode	Varchar2(10)	Debit or Credit
Entry_date	Date(20)	Entry date
Amount	Number(2,8)	amount

VOUCHER_ENTRY

This table is used to maintain voucher details.

FIELD NAME	DATATYPE	DESCRIPTION
Sno	Number(5)	Serial number
Comp_name	Varchar2(20)	Company Name
Name	Varchar2(20)	Name of the person
Alias	Varchar2(20)	Purpose
Under	Varchar2(20)	Account names
Mmode	Varchar2(10)	Debit or Credit
Entry_date	Date(20)	Entry date
Amount	Number(2,8)	amount

SUPPLIER_DETAILS

This table is used to maintain supplier details.

FIELD NAME	DATATYPE	DESCRIPTION
Supp_code	Varchar2(20)	Supplier Code
Supp_name	Varchar2(20)	Supplier Name
Comp_name	Varchar2(20)	Company Name
Person	Varchar2(20)	Contact Person
Supp_address	Varchar2(40)	Supplier Address
City	Varchar2(20)	City
State	Varchar2(20)	State
District	Varchar2(20)	District
Phone	Number(20)	Phone Number
Pay_mode	Varchar2(20)	Payment Mode

PURCHASE ORDER

This table is used to maintain purchase order details.

FIELD NAME	DATATYPE	DESCRIPTION
Comp_name	Varchar2(20)	Company Name
Porder_no	Varchar2(20)	Purchase Order Number
Date1	Varchar2(20)	Date
Supp_id	Varchar2(20)	Supplier ID

PURCHASE ORDER1

This table is used to maintain purchase item details.

FIELD NAME	DATATYPE	DESCRIPTION
Porder_no	Varchar2(20)	Purchase Order Number
Sno	Varchar2(20)	Serial Number
Particulars	Varchar2(20)	Item Name
Qty	Number(10)	Quantity
Type	Varchar2(20)	Type group
color	Varchar2(20)	color

PRIMARY INVOICE

This table is used to maintain primary invoice details.

FIELD NAME	DATATYPE	DESCRIPTION
Invoiceno	Varchar2(20)	Invoice Number
Dc_no	Varchar2(20)	Delivery Challan No
Dc_date	Date(10)	Delivery Challan date
Invoice_date	Date(10)	Invoice date
Porder_no	Varchar2(20)	Purchase order date
Supp_id	Varchar2(20)	Supplier ID

CUSTOMER DETAILS

This table is used to maintain customer details.

FIELD NAME	DATATYPE	DESCRIPTION
Comp_name	Varchar2(20)	Company name
Cust_code	Varchar2(20)	Customer code
Cust_name	Varchar2(20)	Customer name
Address1	Varchar2(30)	Door No/Street name
Address2	Varchar2(30)	Area name
City	Varchar2(20)	City
State	Varchar2(20)	State
Country	Varchar2(20)	Country
Zipcode	Number(20)	Zip Code
Phoneno	Varchar2(20)	Phone number
Email	Varchar2(20)	Email ID

SECONDARY INVOICE

This table is used to maintain secondary invoice details

FIELD NAME	DATATYPE	DESCRIPTION
Invoice_no	Varchar2(20)	Invoice number
Invoice_date	Date(15)	Invoice date
Dc_no	Varchar2(20)	Delivery Challan no
Dc_date	Date(15)	Delivery Challan date
Order_no	Varchar2(20)	Order number
Odate	Date(15)	Order date
Tngst_no	Varchar2(20)	TNGST number
Transport	Varchar2(20)	Mode of transport
Concern_name	Varchar2(20)	Concern name
Address1	Varchar2(30)	Address1
Address2	Varchar2(30)	Address2
City	Varchar2(20)	City
State	Varchar2(20)	State

SECONDARY BILLING

This table is used to maintain billing details.

FIELD NAME	DATATYPE	DESCRIPTION
Billno	Varchar2(20)	Bill number
Billdate	Date(15)	Bill date
Invoiceno	Varchar2(20)	Invoice number
Bill_amt	Number(2,7)	Bill amount
Discount	Number(3)	Discount rate
Totalamt	Varchar2(20)	Total amount
To1	Varchar2(20)	To person

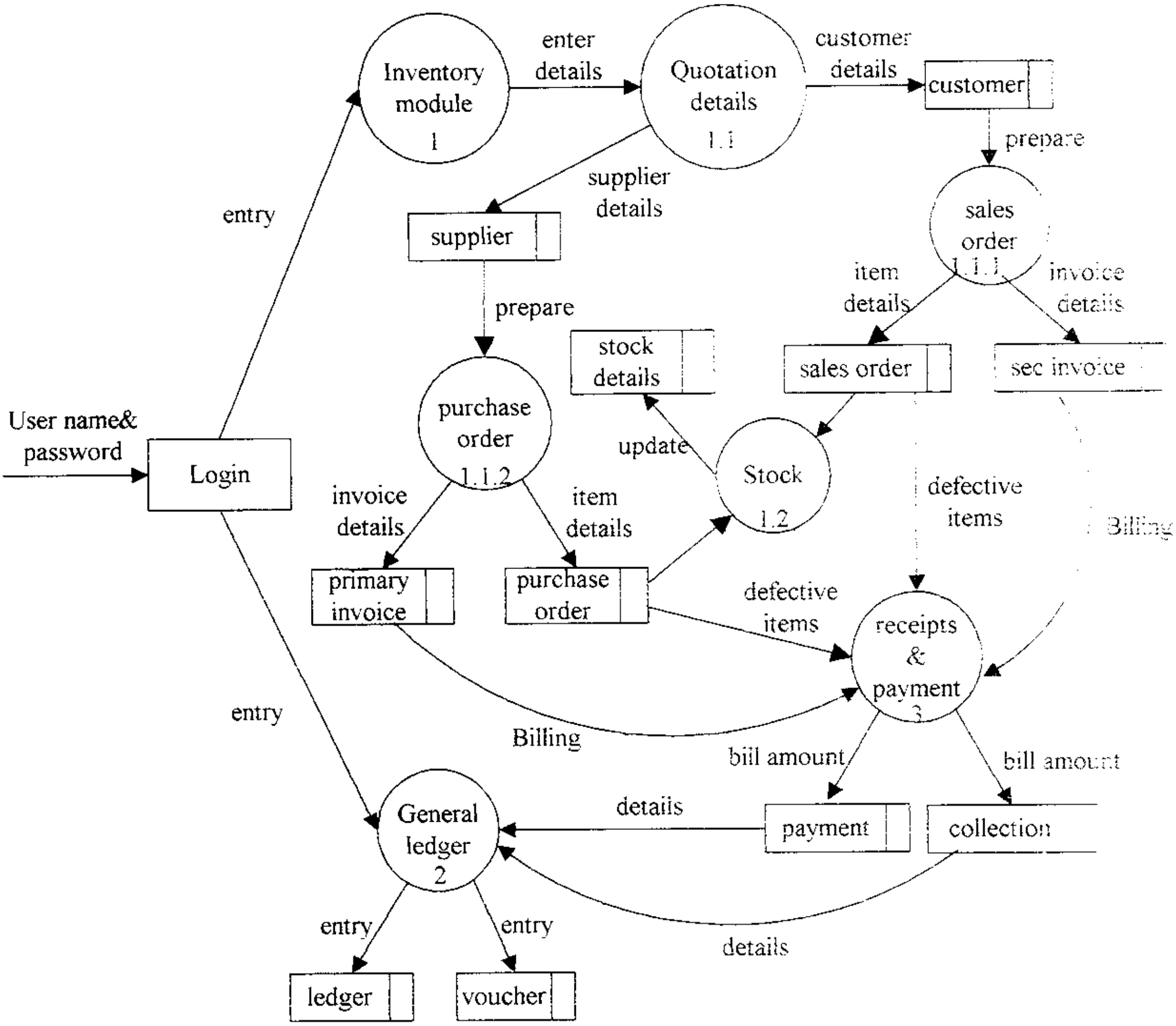
STOCK MASTER

This table is used to maintain stock details.

FIELD NAME	DATATYPE	DESCRIPTION
Prod_no	Varchar2(20)	Product Number
Prod_name	Varchar2(20)	Product Name
Qty	Number(6)	Quantity
Weight	Number(5)	Weight
Type	Varchar2(20)	Type group
Rate	Number(2,8)	Rate per product

DATA FLOW DIAGRAM

The dataflow diagram for the system is shown below



5. SYSTEM IMPLEMENTATION & TESTING

Testing and implementation is the final phase of any software development. In this phase most possible error are identified and rectified to make the system as error free.

5.1 IMPLEMENTATION

Implementation is process of making the system design into a working system. The most crucial stage is achieving a successful new system and is giving the user's confidence is that the new system will work and be effective in the implementation stage. This stage consists of

- Testing the developed program with sample data.
- Detection and correction of errors.
- Testing whether the system meets user requirements.
- Creating filers of the system with the actual data.
- Making necessary changes as desired by the user.
- Training user personnel.

It involves careful maintaining, investigation of the current system and its constraints and implementation, design of methods to achieve the changeover, and evaluation of changeover methods. Apart from maintenance there are two major tasks of preparing for implementation are education and training of users and system testing.

The system has been tested with sample data and adequate corrections were made as per user requirements. The user has very little chances of making data entry errors since enough validation checks and validation error messages are provided in the system. The end user even with minimum amount of computer knowledge will be able to key in the data and understand the error messages. All reports have been found to satisfy their requirements.

5.2 SYSTEM TESTING

System testing is recognized as an important part of quality assurance. One of the important parts of construction of a software product is testing. It is necessary to test all modules to make sure they are error-free once they are put into operation. The individual modules are first tested and then their interfaces are tested.

When the programmers have tested each programs individually using test data designed by themselves and have verified that these programs unit specification. The complete system and its environment must be test to the satisfaction of the system analysts and the user. The results of the processed test data should be maintained as a permanent manual throughout the operational life of the system, for audit purposes or to test any major subsequent major amendments.

UNIT TESTING

Unit testing focuses verification effort and smallest unit of software design of the module. The unit testing is always conducted in parallel for multiple modules. The Unit testing is done to test all independent paths by ensuring that all the statements are executed atleast once.

Checking whether the amount is updated correctly in both Ledger and Voucher tests the General Ledger module. Erroneous data is provided in order to check if the error message and the error routines are executed properly. The testing for error messages is done in Type group and Stock group options that do not allow duplicate values to be entered.

INTEGRATION TESTING

Integration Testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by the design.

Purchase order form many items like Supplier details, Item details and ordered item details are displayed. The General Ledger records all the cash transactions and also the bank transactions.

TOP-DOWN INTEGRATION

It is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with main control. Modules subordinate to the main control modules are incorporated into the structure in either depth-first or breadth-first manner.

SYSTEM SECURITY

Security measures will provide the unauthorized access of the database at various levels. An uninterrupted power supply should be provided so that the power failure or voltage fluctuations will not erase the data in the file.

The system allows the user to enter only through login facility. The user will have to enter the User ID and Password. If the User ID and Password does not match with any of the entries then it is unable for the user to enter. If the User ID and Password matches with the entries present then the user will be connected to the database used for the entire system

6.CONCLUSION

This project software **Business Solution** is a complete Accounts Package. The developed system is highly interactive and user friendly. Reports have been generated so that it satisfies user requirements to the maximum possible extend. The system provides accurate updating, data validation and integrity.

The system has been designed and run to satisfy the needs of the entire organization. The existing system makes the job very difficult. This process will reduce clerical work and also resulted in quick retrieval of information, which is very vital to the progress of an organization.

The project has been successfully completed and tested using the sample database information.

7. SCOPE FOR FUTURE DEVELOPMENT

The system has been designed and developed flexibly to maintain inventory and account details according to the current requirement of the user. Proper modular design has been made. As the requirements may still increase in the near future, the application can be upgraded for the new ones and it is possible to do modifications according to new requirements and specifications.

In future the system can be extended to maintained employee details and banking details of the organization. The system can also be developed to maintain branch details. The modular approach and the programming techniques incorporated during the development of system will be of great help for future enhancements.

8. BIBLIOGRAPHY

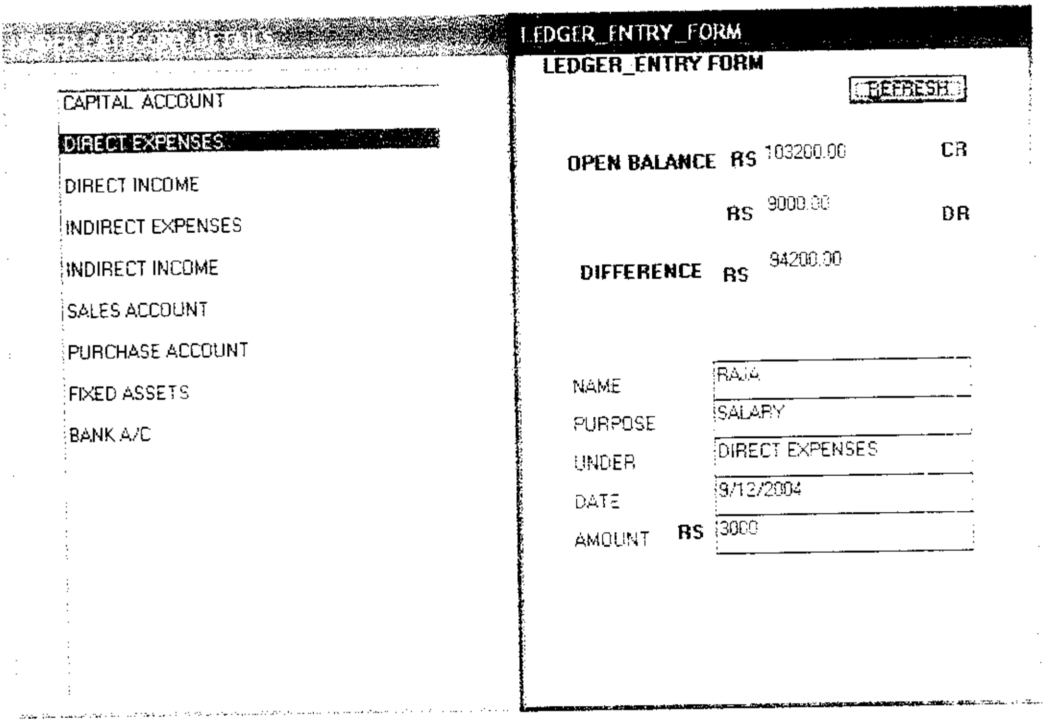
- Gary Cornell, **Visual Basic 6 - From the Ground Up**, Tata Mc-Graw Hill. Edition 1999.
- Neol Jerke, **Visual Basic 6 - The Complete Reference,** Sams publishers. Edition 1999.
- Michael Abbey, Michael J. Corney, Ian Abramson-**Oracle 8 A Beginner's Guide,** Tata Mc-Graw Hill Publications, Edition 2001.

WEBSITES

- www.vbexplorer.com
- www.visualbasic101.com
- www.vbworld.com

SAMPLE FORMS

LEDGER ENTRY



CAPITAL ACCOUNT
DIRECT EXPENSES
DIRECT INCOME
INDIRECT EXPENSES
INDIRECT INCOME
SALES ACCOUNT
PURCHASE ACCOUNT
FIXED ASSETS
BANK A/C

LEDGER_ENTRY FORM

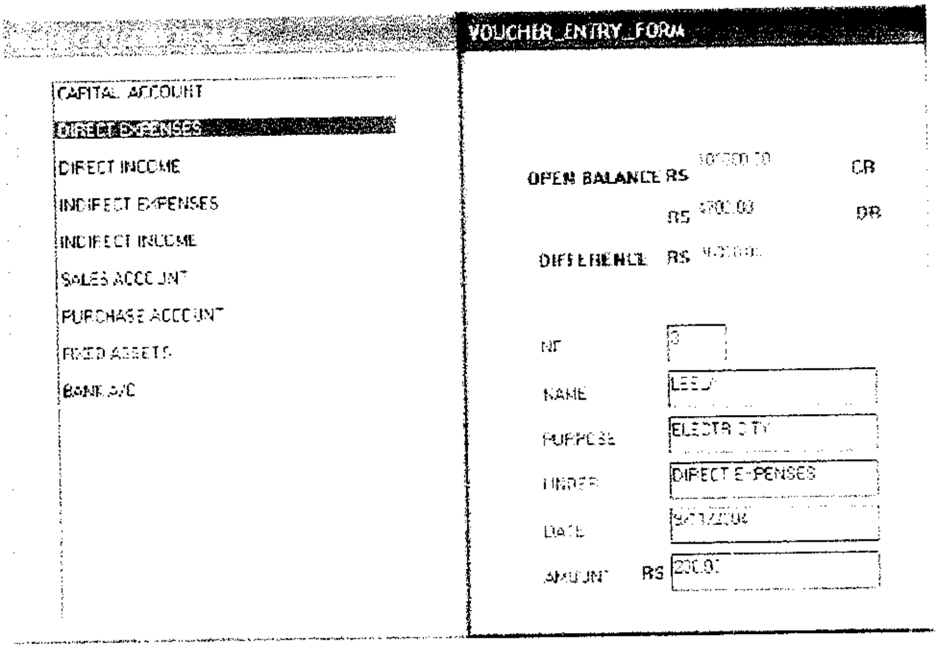
OPEN BALANCE RS 103200.00 CR

RS 9000.00 DR

DIFFERENCE RS 94200.00

NAME RAJA
PURPOSE SALARY
UNDER DIRECT EXPENSES
DATE 9/12/2004
AMOUNT RS 3000

VOUCHER ENTRY



VOUCHER_ENTRY_FORM

CAPITAL ACCOUNT

DIRECT EXPENSES

DIRECT INCOME

INDIRECT EXPENSES

INDIRECT INCOME

SALES ACCOUNT

PURCHASE ACCOUNT

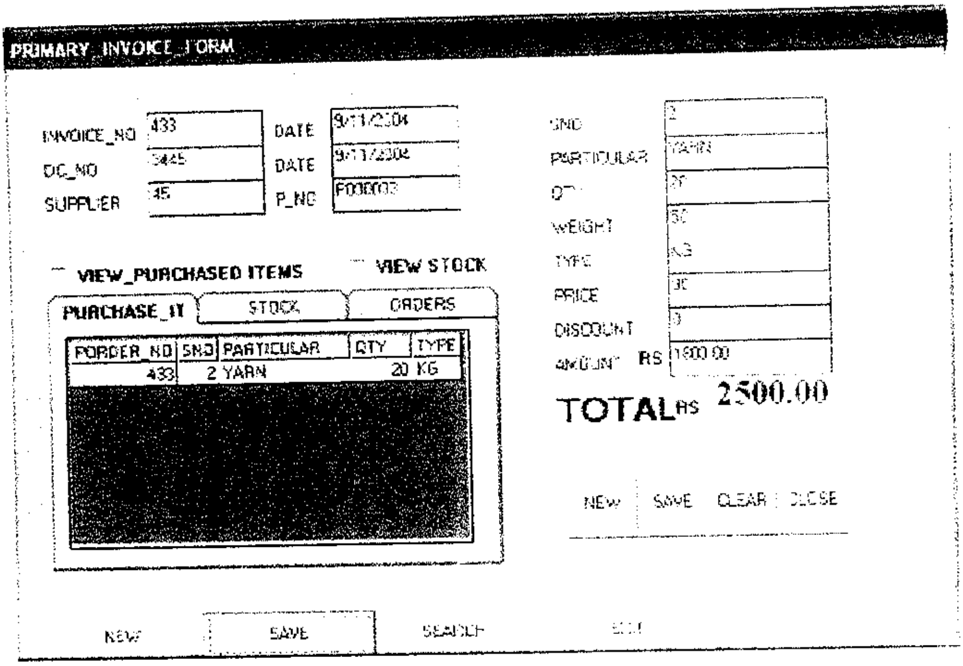
FIXED ASSETS

BANK A/C

OPEN BALANCE	RS	100000.00	CR
	RS	4700.00	DR
DIFFERENCE	RS	[illegible]	

NR	3
NAME	LEEJ
PURPOSE	ELECTRICITY
UNDER	DIRECT EXPENSES
DATE	9/12/2004
AMOUNT RS	200.00

PRIMARY INVOICE



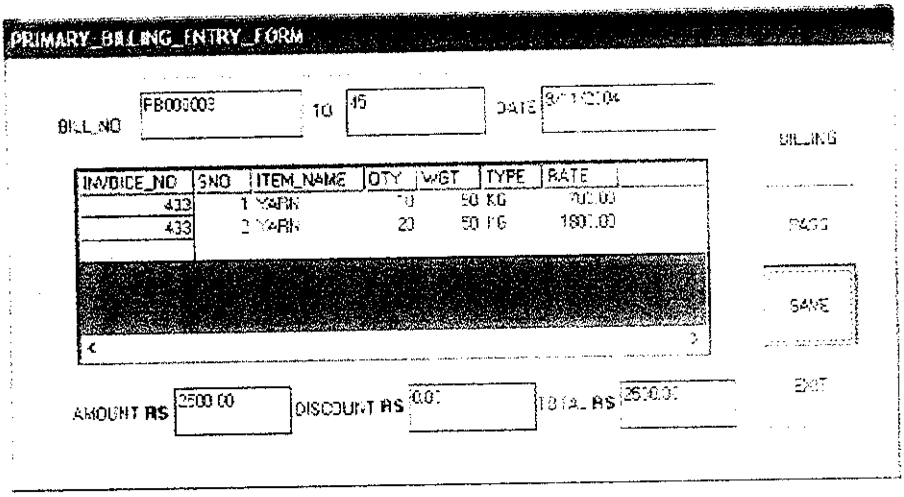
INVOICE_NO	433	DATE	9/11/2004
DC_NO	3445	DATE	9/11/2004
SUPPLIER	45	P_NO	F000003

PORDER_NO	SNO	PARTICULAR	QTY	TYPE
433	2	YARN	20	KG

SNO	2
PARTICULAR	YARN
QTY	20
WEIGHT	50
TYPE	KG
PRICE	30
DISCOUNT	0
AMOUNT RS	1500.00

TOTAL RS 2500.00

PRIMARY BILL FROM SUPPLIER

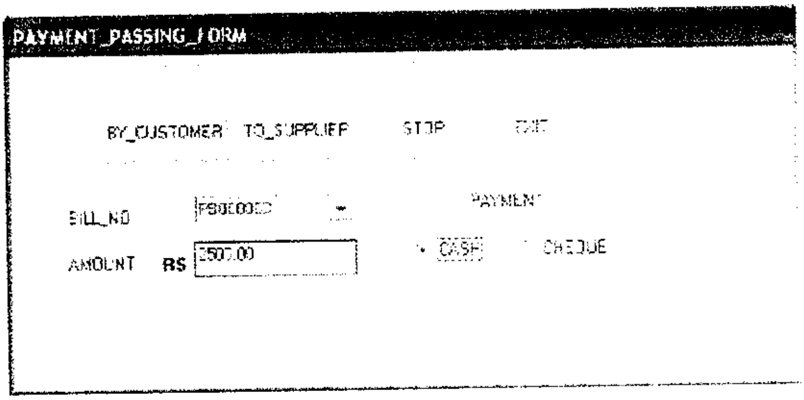


PRIMARY_BILLING_ENTRY_FORM

BILL_NO FB000003 TO 45 DATE 9/11/2004

INVOICE_NO	SNO	ITEM_NAME	QTY	WGT	TYPE	RATE
433	1	YARN	10	50	KG	700.00
433	2	YARN	20	50	KG	1800.00

AMOUNT RS 2500.00 DISCOUNT RS 0.00 TOTAL RS 2500.00



PAYMENT_PASSING_FORM

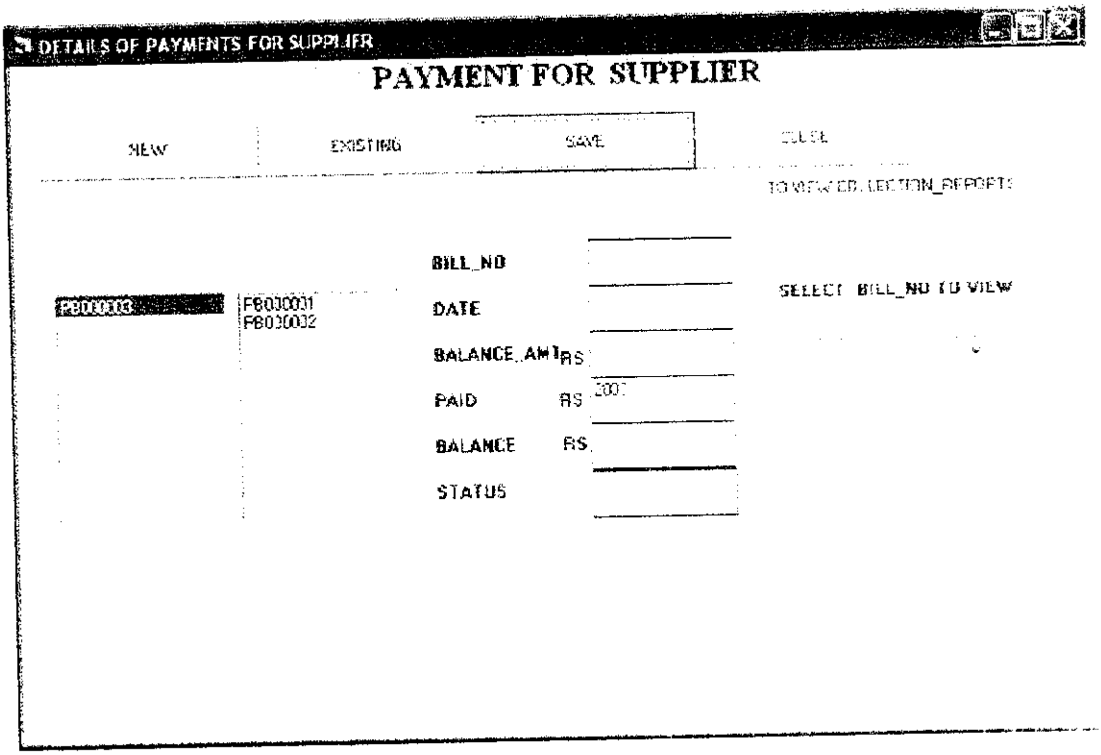
BILL_NO FB000003

AMOUNT RS 2500.00

PAYMENT

● CASH ○ CHEQUE

PAYMENT FOR SUPPLIER



PAYMENT FOR SUPPLIER

FB030003

FB030001
FB030002

BILL_NO

DATE

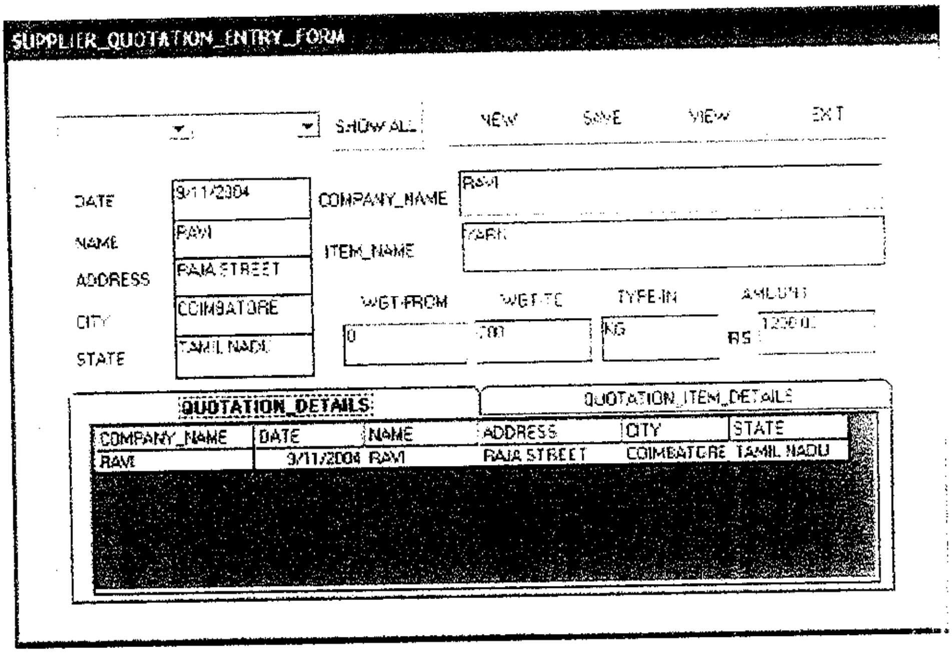
BALANCE_AMT RS

PAID RS 2001

BALANCE RS

STATUS

QUOTATION DETAILS



DATE 9/11/2004

NAME RAVI

ADDRESS RAJA STREET

CITY COIMBATORE

STATE TAMIL NADU

COMPANY_NAME RAVI

ITEM_NAME YARN

WGT-FROM 0

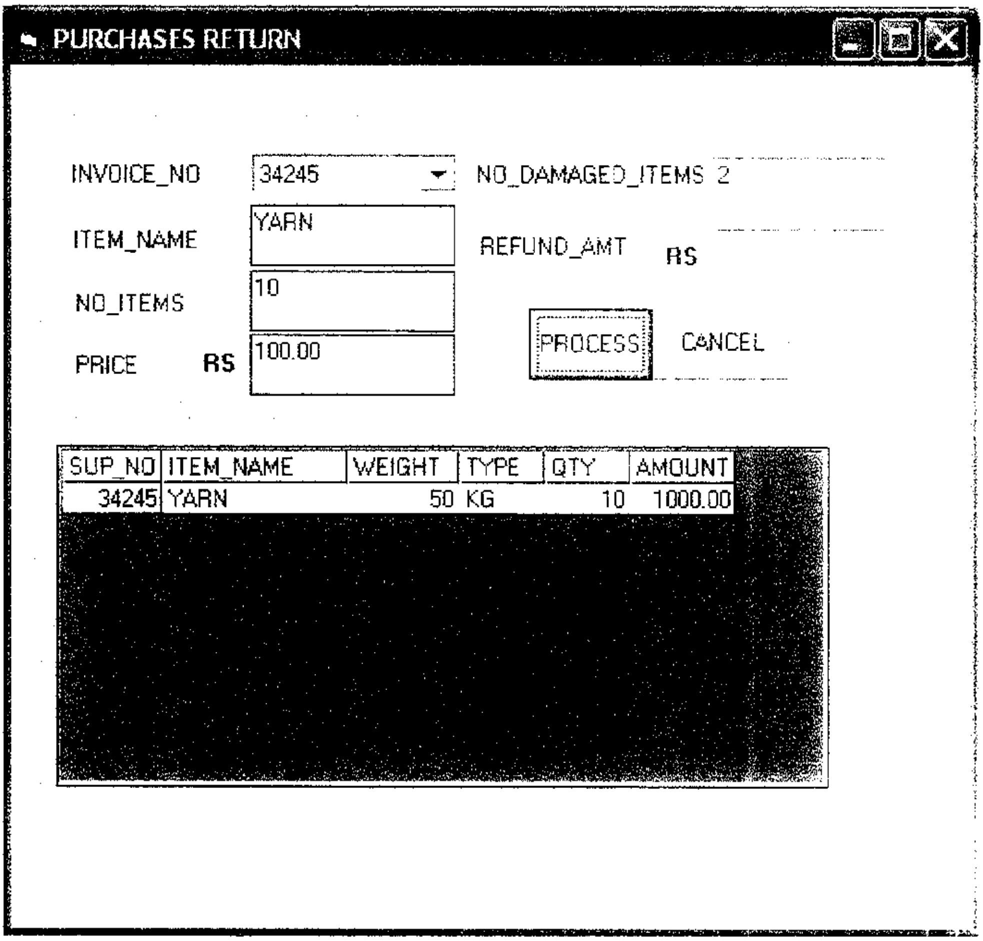
WGT-TO 100

TYPE-IN KG

AMOUNT RS 1200.00

COMPANY_NAME	DATE	NAME	ADDRESS	CITY	STATE
RAVI	9/11/2004	RAVI	RAJA STREET	COIMBATORE	TAMIL NADU

PURCHASES RETURN



INVOICE_NO 34245

NO_DAMAGED_ITEMS 2

ITEM_NAME YARN

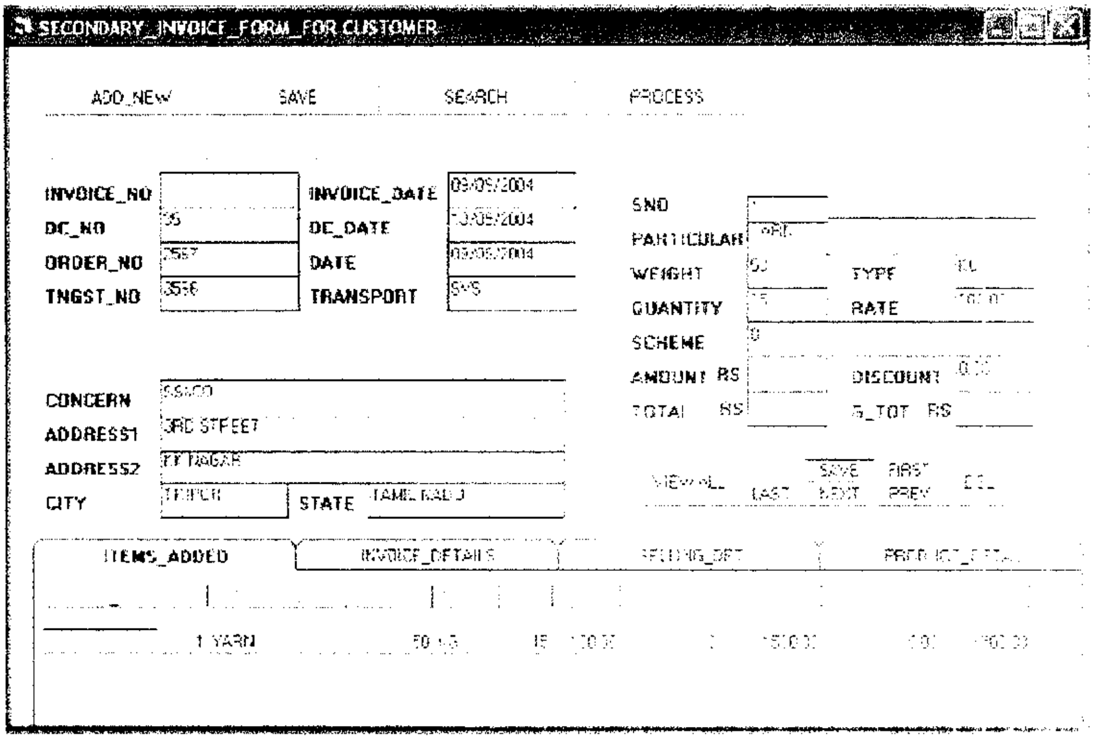
REFUND_AMT RS

NO_ITEMS 10

PRICE RS 100.00

SUP_NO	ITEM_NAME	WEIGHT	TYPE	QTY	AMOUNT
34245	YARN	50	KG	10	1000.00

SECONDARY INVOICE



INVOICE_NO

DC_NO 35

ORDER_NO 2567

TNGST_NO 3556

INVOICE_DATE 09/09/2004

DC_DATE 10/09/2004

DATE 09/09/2004

TRANSPORT SVS

CONCERN SSACO

ADDRESS1 3RD STREET

ADDRESS2 KK NAGAR

CITY TRIPUR

STATE TAMIL NADU

SNO 1

PARTICULAR YARN

WEIGHT 50

TYPE KG

QUANTITY 15

RATE 100.00

SCHEME 0

AMOUNT RS

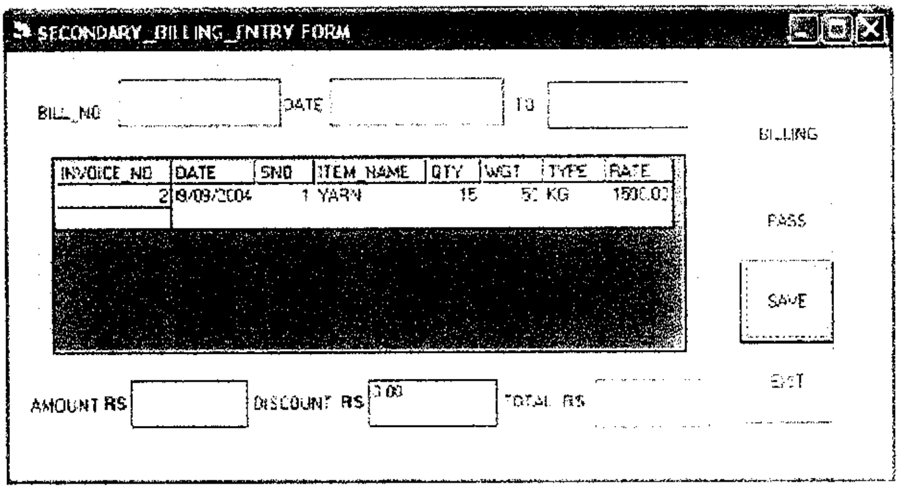
DISCOUNT 0.00

TOTAL RS

G_TOT RS

1 YARN 50 KG 15 100.00 0 1500.00 0.00 1500.00

SECONDARY BILL FOR CUSTOMER

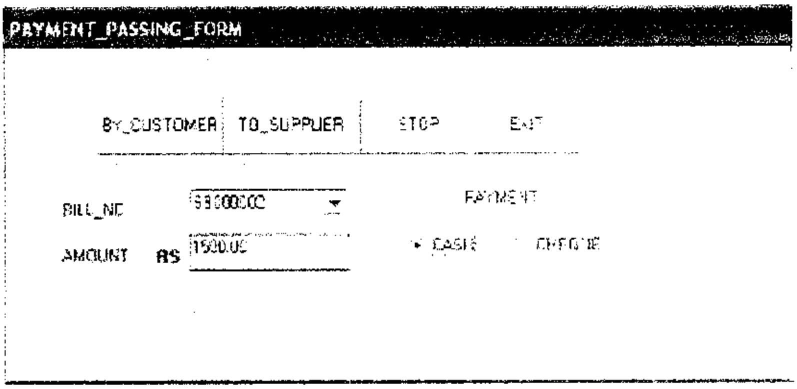


SECONDARY_BILLING_ENTRY FORM

BILL_NO DATE TO

INVOICE_NO	DATE	SNO	ITEM_NAME	QTY	WGT	TYPE	RATE
2	9/09/2004	1	YARN	15	50	KG	1500.00

AMOUNT RS DISCOUNT RS 0.00 TOTAL RS



PAYMENT_PASSING_FORM

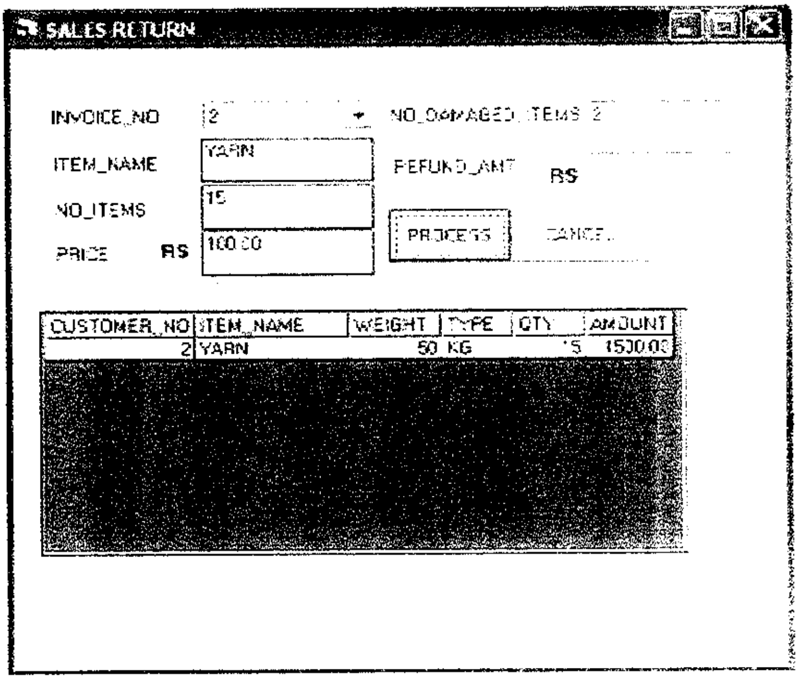
BILL_NO 9900000C

AMOUNT RS 1500.00

PAYMENT

● CASH ○ CHEQUE

SALES RETURN



INVOICE_NO 2

NO_DAMAGED_ITEMS 2

ITEM_NAME YARN

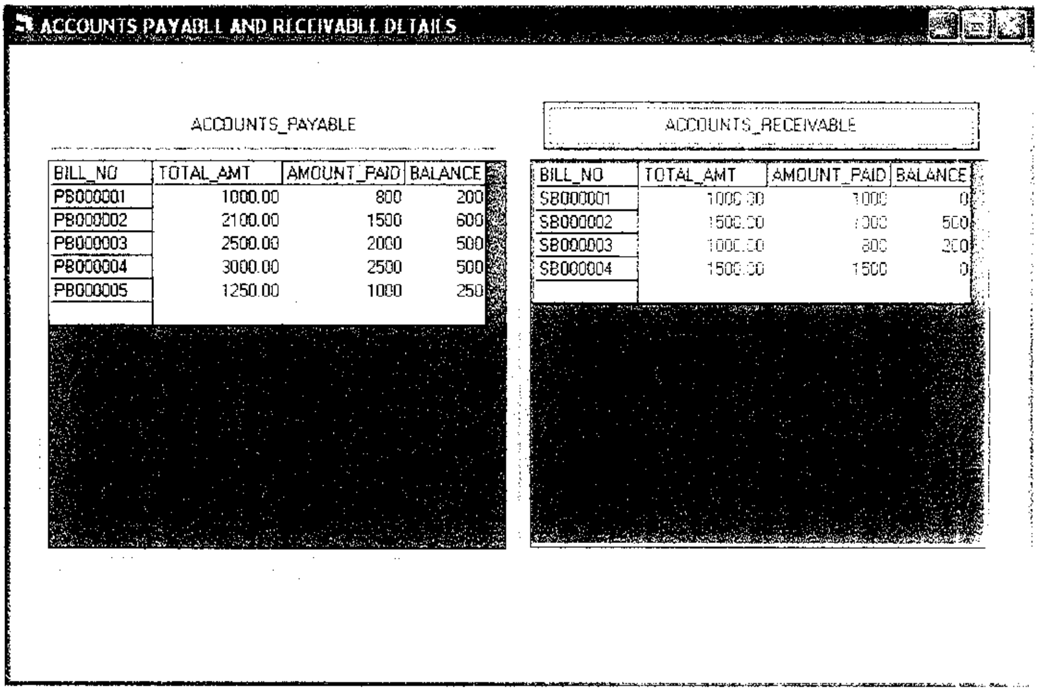
REFUND_AMT RS

NO_ITEMS 15

PRICE RS 100.00

CUSTOMER_NO	ITEM_NAME	WEIGHT	TYPE	QTY	AMOUNT
2	YARN	50	KG	15	1500.00

ACCOUNTS PAYABLE AND RECEIVABLE



ACCOUNTS PAYABLE AND RECEIVABLE DETAILS

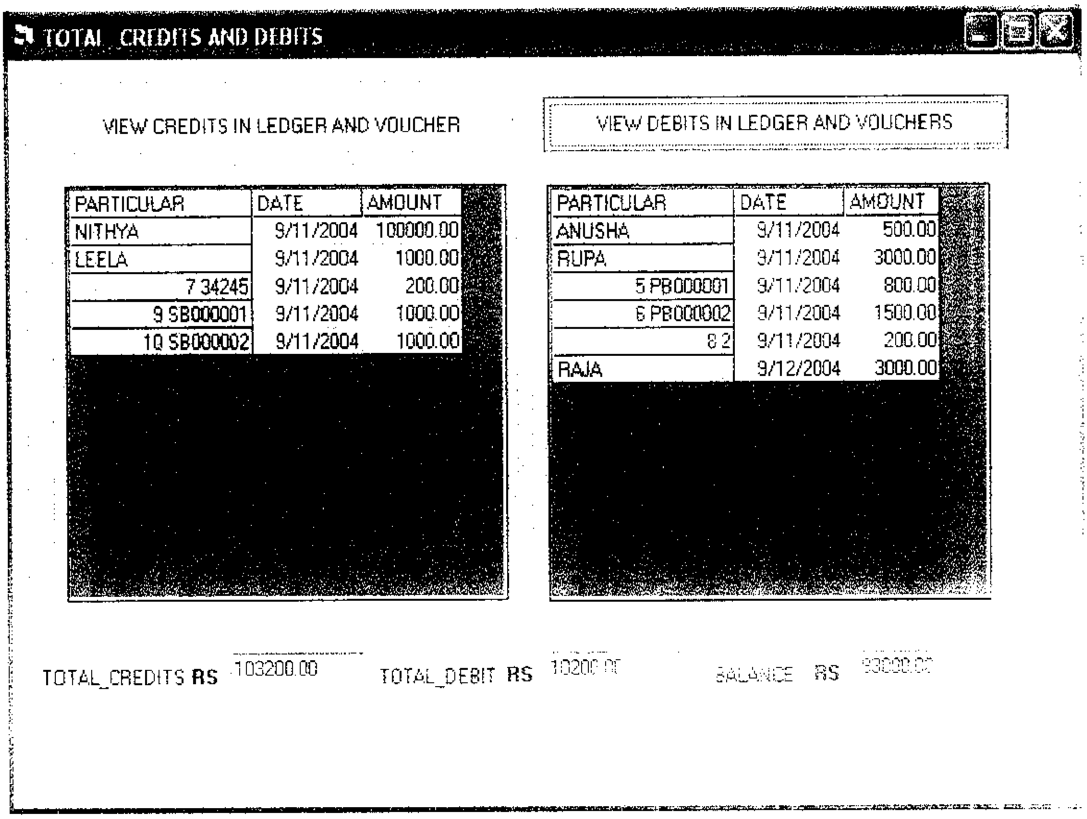
ACCOUNTS_PAYABLE

BILL_NO	TOTAL_AMT	AMOUNT_PAID	BALANCE
PB000001	1000.00	800	200
PB000002	2100.00	1500	600
PB000003	2500.00	2000	500
PB000004	3000.00	2500	500
PB000005	1250.00	1000	250

ACCOUNTS_RECEIVABLE

BILL_NO	TOTAL_AMT	AMOUNT_PAID	BALANCE
SB000001	1000.00	1000	0
SB000002	1500.00	1000	500
SB000003	1000.00	800	200
SB000004	1500.00	1500	0

TOTAL DEDITS AND CREDITS



TOTAL_ CREDITS AND DEBITS

VIEW CREDITS IN LEDGER AND VOUCHER

VIEW DEBITS IN LEDGER AND VOUCHERS

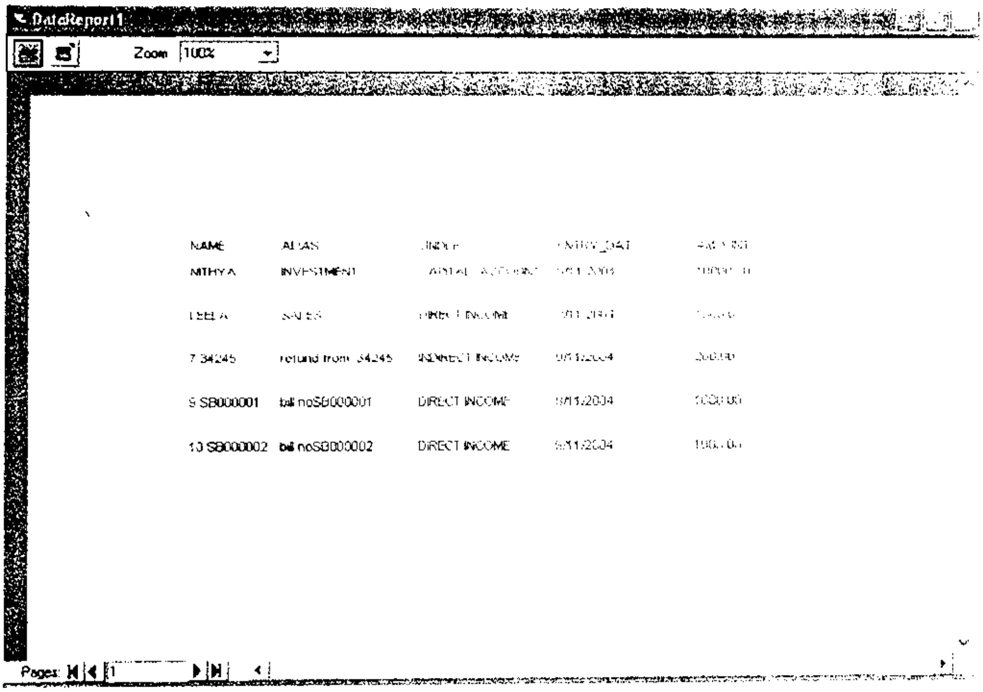
PARTICULAR	DATE	AMOUNT
NITHYA	9/11/2004	100000.00
LEELA	9/11/2004	1000.00
7 34245	9/11/2004	200.00
9 SB000001	9/11/2004	1000.00
10 SB000002	9/11/2004	1000.00

PARTICULAR	DATE	AMOUNT
ANUSHA	9/11/2004	500.00
RUPA	9/11/2004	3000.00
5 PB000001	9/11/2004	800.00
6 PB000002	9/11/2004	1500.00
8 2	9/11/2004	200.00
RAJA	9/12/2004	3000.00

TOTAL_CREDITS **RS** 103200.00 TOTAL_DEBIT **RS** 10200.00 BALANCE **RS** 93000.00

SAMPLE REPORTS

CREDITS IN LEDGER



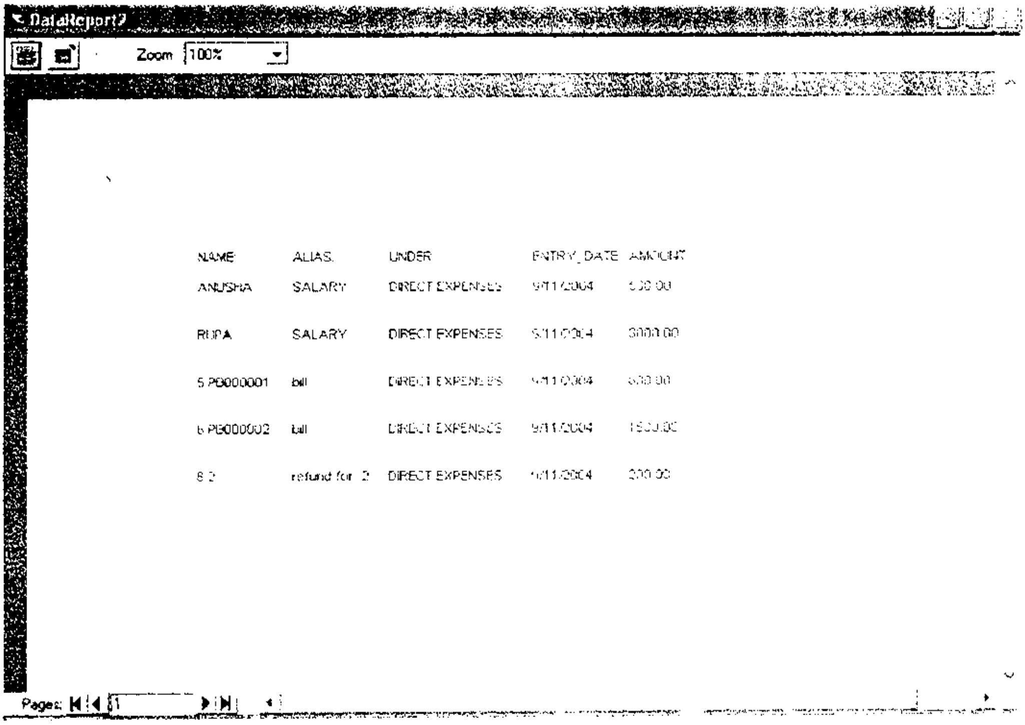
DataReport1

Zoom 100%

NAME	ALIAS	UNDER	ENTRY_DAT	AMOUNT
NITHYA	INVESTMENT	CAPITAL ACCOUNT	[illegible]	[illegible]
LEELA	SALES	DIRECT INCOME	[illegible]	[illegible]
7 34245	refund from 34245	INDIRECT INCOME	9/11/2004	2000.00
9 SB000001	bill noSB000001	DIRECT INCOME	9/11/2004	1000.00
10 SB000002	bill noSB000002	DIRECT INCOME	9/11/2004	1500.00

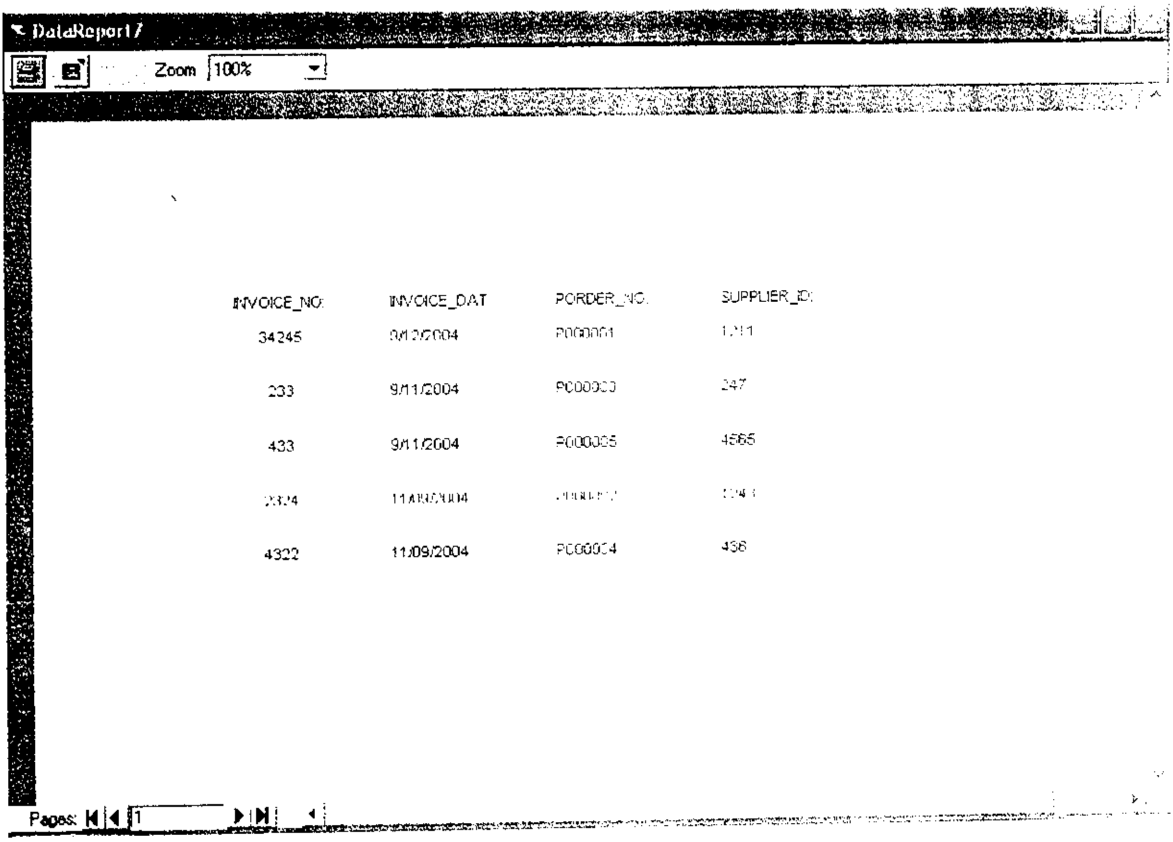
Pages: 1

DEBITS IN LEDGER



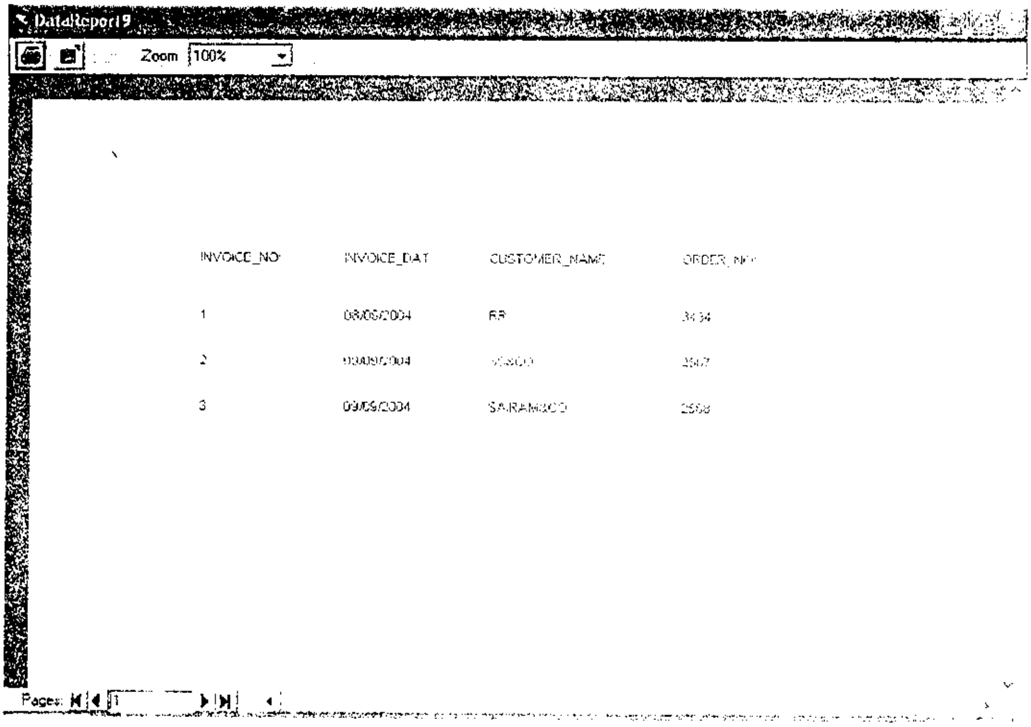
NAME	ALIAS	UNDER	ENTRY_DATE	AMOUNT
ANUSHA	SALARY	DIRECT EXPENSES	9/11/2004	500.00
RUPA	SALARY	DIRECT EXPENSES	9/11/2004	3000.00
5 PB000001	bill	DIRECT EXPENSES	9/11/2004	600.00
6 PB000002	bill	DIRECT EXPENSES	9/11/2004	1500.00
8 2	refund for 2	DIRECT EXPENSES	9/11/2004	200.00

PRIMARY INVOICE



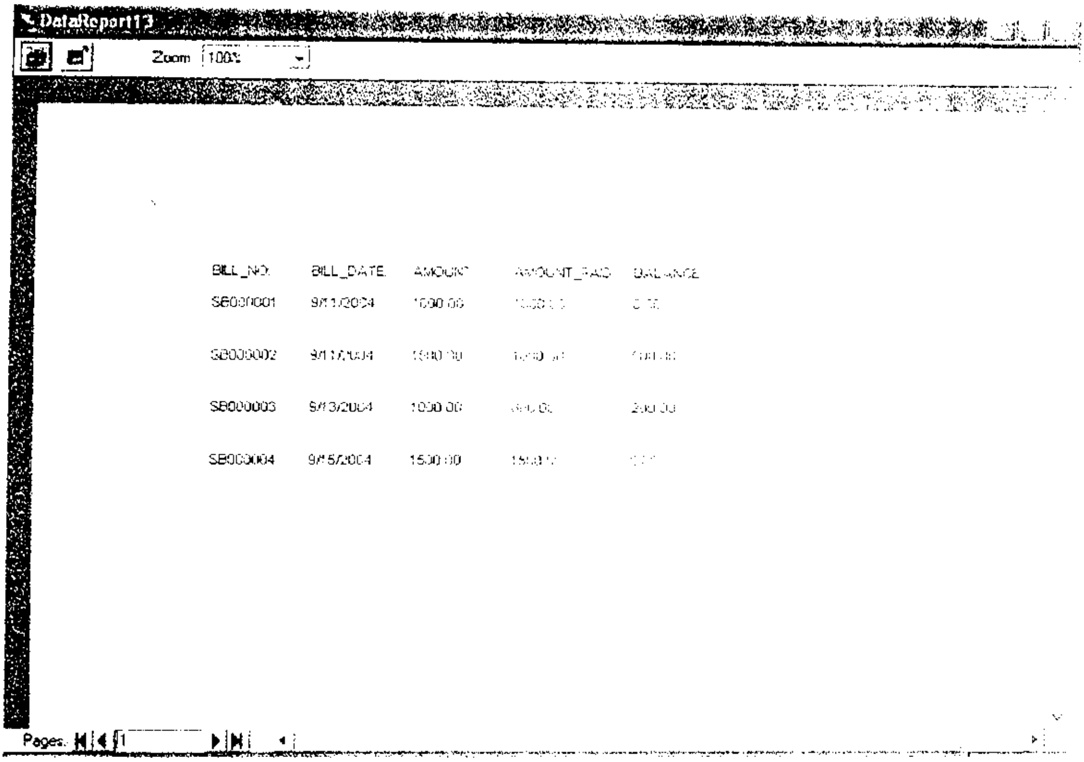
INVOICE_NO	INVOICE_DAT	PORDER_NO	SUPPLIER_ID
34245	9/12/2004	P000001	1211
233	9/11/2004	P000003	247
433	9/11/2004	P000005	4565
2324	11/09/2004	[illegible]	[illegible]
4322	11/09/2004	P000004	436

SECONDARY INVOICE



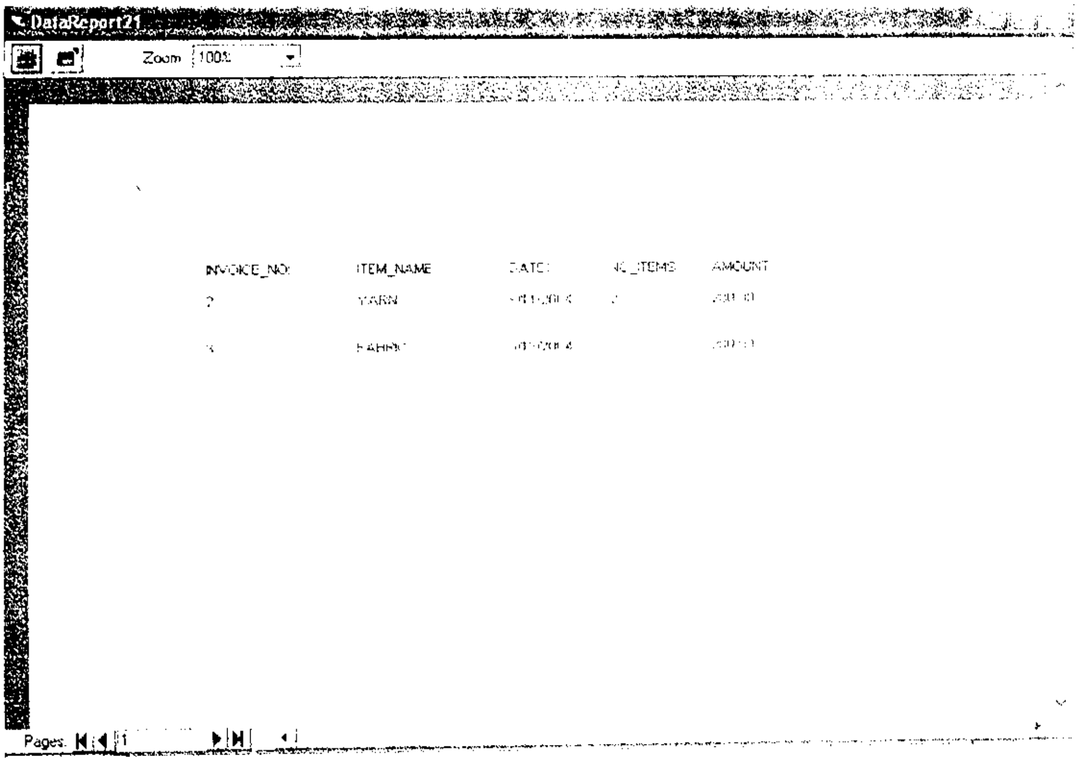
INVOICE_NO	INVOICE_DAT	CUSTOMER_NAME	ORDER_NO
1	08/09/2004	RR	3434
2	03/09/2004	[illegible]	2567
3	09/09/2004	SAIRAMCO	2568

COLLECTION



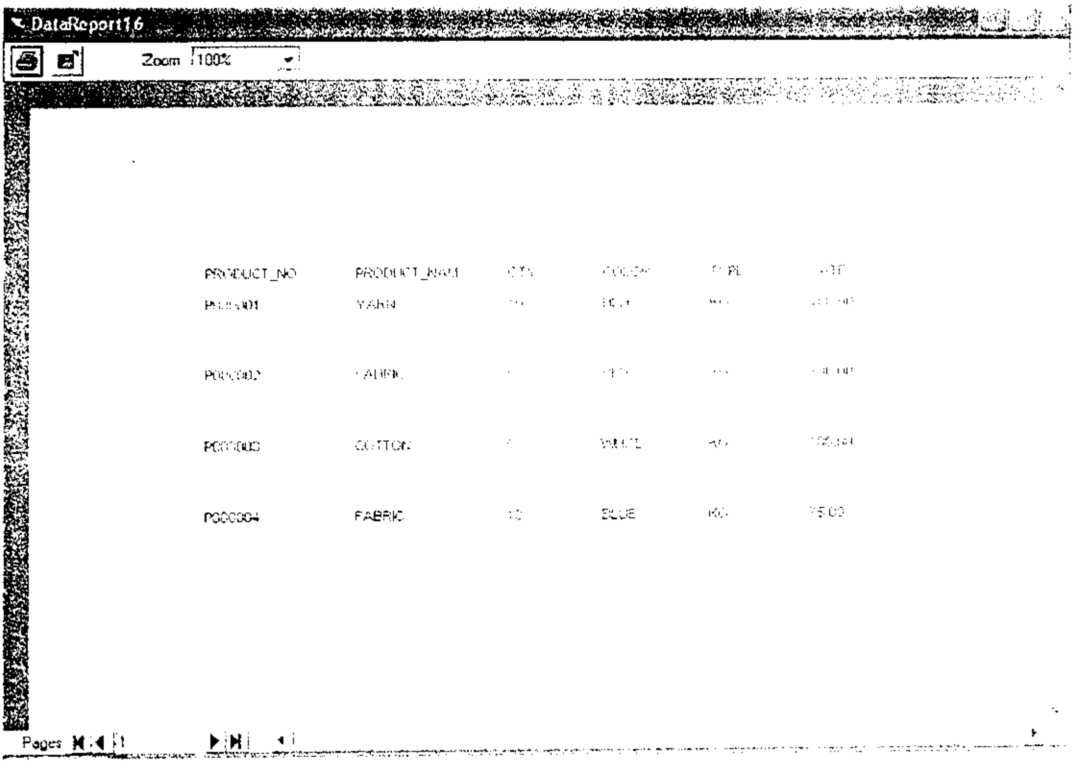
BILL_NO:	BILL_DATE	AMOUNT	AMOUNT_PAID	BALANCE
SB000001	9/11/2004	1000.00	1000.00	0.00
SB000002	9/11/2004	1500.00	1000.00	500.00
SB000003	9/13/2004	1000.00	800.00	200.00
SB000004	9/15/2004	1500.00	1500.00	0.00

SALES RETURN



INVOICE_NO	ITEM_NAME	DATE	NO_ITEMS	AMOUNT
2	YARN	[illegible]	2	[illegible]
3	FABRIC	[illegible]		[illegible]

STOCK_DETAILS



PRODUCT_NO	PRODUCT_NAM	QTY	COLOUR	TYPE	RATE
P000001	YARN	[illegible]	[illegible]	[illegible]	[illegible]
P000002	FABRIC	[illegible]	[illegible]	[illegible]	[illegible]
P000003	COTTON	[illegible]	WHITE	KG	[illegible]
P000004	FABRIC	10	BLUE	KG	75.00

SAMPLE CODE

```
Dim cn As ADODB.Connection
Dim str As String
Dim rs As ADODB.Recordset
Dim rs1 As ADODB.Recordset
Dim rs2 As ADODB.Recordset
Dim rs3 As ADODB.Recordset
Dim str3 As String
Dim flag As Boolean

Private Sub Check1_Click()
If Check1.Value = 1 Then
Dim rr As ADODB.Recordset
Dim ss As String
Set rr = New ADODB.Recordset
ss = "select * from stock_master"
rr.Open ss, cn, adOpenDynamic, adLockOptimistic
MSFlexGrid1.FormatString = "PRO_ID   |NAME       |QTY |WEIGHT|TYPE"
MSFlexGrid1.Rows = 1
Do Until rr.EOF
MSFlexGrid1.AddItem (rr("product_no") & vbTab & rr("product_name") & vbTab &
rr("qty") & vbTab & rr("weight") & vbTab & rr("type"))
rr.MoveNext
Loop
Else
MSFlexGrid1.Clear
MSFlexGrid1.FormatString = "PRO_ID   |NAME       QTY |WEIGHT|TYPE"
End If
End Sub

Private Sub Check2_Click()
If Check2.Value = 1 Then
Dim rr As ADODB.Recordset
Dim ss As String
Set rr = New ADODB.Recordset
ss = "select * from order_invoice1"
rr.Open ss, cn, adOpenDynamic, adLockOptimistic
MSFlexGrid2.FormatString = "PORDER_NO|SNO|PARTICULAR      QTY     |TYPE"
MSFlexGrid2.Rows = 1
Do Until rr.EOF
MSFlexGrid2.AddItem (rr("invoice_no") & vbTab & rr("sno") & vbTab &
rr("particular") & vbTab & rr("qty") & vbTab & rr("type"))
rr.MoveNext
Loop
```

```
Else
MSFlexGrid2.Clear
MSFlexGrid2.FormatString = "PORDER_NO|SNO|PARTICULAR     |QTY    |TYPE"
End If
End Sub

Private Sub new1cmd_Click()
invoicenotxt.Text = ""
dcnotxt.Text = ""
date1txt.Text = ""
date2txt.Text = ""
supnotxt.Text = ""
pnotxt.Text = ""
invoicenotxt.SetFocus
End Sub

Private Sub save1cmd_Click()
If Len(invoicenotxt.Text) = 0 Or Len(dcnotxt.Text) = 0 Or Len(date1txt.Text) = 0 Or
Len(date2txt.Text) = 0 Or Len(supnotxt.Text) = 0 Or Len(pnotxt.Text) = 0 Then
   MsgBox "PLEASE FILL ALL TEXTS", vbCritical
Else
   rs.AddNew
      rs("INVOICE_NO") = invoicenotxt.Text
      rs("INVOICE_date") = date1txt.Text
      rs("DC_no") = dcnotxt.Text
      rs("dc_date") = date2txt.Text
      rs("supplier_id") = supnotxt.Text
      rs("porder_no") = pnotxt.Text
   rs.update
   MSFlexGrid3.Rows = 1
   MSFlexGrid3.FormatString = "INVOICE_NO|DATE    |DC_NO|DATE    |PUR_NO
|SUP_NO"
   MSFlexGrid3.Rows = MSFlexGrid3.Rows + 1
   Call AddToGrid2(MSFlexGrid2.Rows - 1)
End If
End Sub

Private Sub new2cmd_Click()
snotxt.Text = auto1
End Sub

Private Sub save2cmd_Click()
If Len(snotxt.Text) = 0 Or Len(particulartxt.Text) = 0 Or Len(qtytxt.Text) = 0 Or
Len(typetxt.Text) = 0 Or Len(pricetxt.Text) = 0 Or Len(discounttxt.Text) = 0 Or
Len(amounttxt.Text) = 0 Then
MsgBox "PLEASE FILL TEXTS", vbCritical
```

```
Else

  rs1.AddNew
     rs1("invoice_no") = invoicenotxt.Text
     rs1("sno") = snotxt.Text
     rs1("particular") = particulartxt.Text
     rs1("qty") = qtytxt.Text
     rs1("type") = typetxt.Text
     rs1("rate") = pricetxt.Text
     rs1("discount") = discounttxt.Text
     rs1("amount") = amounttxt.Text
     rs1("weight") = weighttxt.Text
  rs1.update
  full
   SSTab1.Tab = 0
   MSFlexGrid2.Rows = 1
   MSFlexGrid2.FormatString = "PORDER_NO|SNO|PARTICULAR      |QTY
|TYPE"
   MSFlexGrid2.Rows = MSFlexGrid2.Rows + 1
   Call AddToGrid1(MSFlexGrid2.Rows - 1)
   Dim str11 As String
   str11 = MsgBox("DO U WANT TO CONTINUE", vbYesNo)
   If str11 = vbYes Then
      snotxt.Text = ""
      particulartxt.Text = ""
      qtytxt.Text = ""
      typetxt.Text = ""
      pricetxt.Text = ""
      discounttxt.Text = ""
      amounttxt.Text = ""
      snotxt.Text = auto1
   ElseIf str11 = vbNo Then
      save1cmd.SetFocus
   End If

End If
End Sub

Private Sub closecmd_Click()
Form29.Show
Form29.Top = 400
Form29.Left = 1785
Form29.Height = 5000
Form29.Width = 9300
Form29.totxt.Text = supnotxt.Text
End Sub
```

```
Private Sub searchcmd_Click()
Dim r As ADODB.Recordset
Dim s, s1 As String
s1 = InputBox("Enter the invoice no")
Set r = New ADODB.Recordset
s = "select * from order_invoice where invoice_no ='" & s1 & "'"
r.Open s, cn, adOpenDynamic, adLockOptimistic
If Not r.BOF And Not r.EOF Then
invoicenotxt.Text = r("invoice_no")
dcnotxt.Text = r("dc_no")
date1txt.Text = r("invoice_date")
date2txt.Text = r("dc_date")
supnotxt.Text = r("supplier_ID")
pnotxt.Text = r("porder_no")
Else
   MsgBox "NOT found"
End If
End Sub

Private Sub exitcmd_Click()
Unload Me
End Sub

Private Sub Form_Load()
Set cn = New ADODB.Connection
cn.Open "pd", "scott", "tiger"
Set rs = New ADODB.Recordset
str = "select * from order_invoice"
rs.Open str, cn, adOpenDynamic, adLockOptimistic

Set rs1 = New ADODB.Recordset
str1 = "select * from order_invoice1 where sno ='" & snotxt.Text & "'"
rs1.Open str1, cn, adOpenDynamic, adLockOptimistic

Set rs2 = New ADODB.Recordset
str2 = "select * from stock_master"
rs2.Open str2, cn, adOpenDynamic, adLockOptimistic
Set rs3 = New ADODB.Recordset
str3 = "select * from ORDER_INVOICE"
rs3.Open str3, cn, adOpenDynamic, adLockOptimistic
MSFlexGrid3.Rows = 1
Do Until rs3.EOF
MSFlexGrid3.AddItem (rs3("invoice_no") & vbTab & rs3("invoice_date") & vbTab &
rs3("dc_no") & vbTab & rs3("dc_date") & vbTab & rs3("porder_no") & vbTab &
rs3("supplier_id"))
```

```
rs3.MoveNext
Loop

End Sub
Public Function auto1()
Dim r2 As ADODB.Recordset
Dim str1 As String
Set r2 = New ADODB.Recordset
str1 = "select max(sno) from order_invoice1 where invoice_no ='" & invoicenotxt.Text &
"'"
r2.Open str1, cn, adOpenDynamic, adLockOptimistic
If r2.BOF Then
   auto1 = 1
Else
   auto1 = IIf(IsNull(r2(0)), 0, r2(0)) + 1
End If
Exit Function
End Function

Private Sub discounttxt_LostFocus()
Dim k
 k = Val(pricetxt.Text) * Val(discounttxt.Text) / 100
 amounttxt.Text = Val(pricetxt.Text) * Val(qtytxt) - Val(k)

End Sub

Private Sub amounttxt_LostFocus()
Label15.Caption = Val(Label15.Caption) + Val(amounttxt.Text)
End Sub
Public Sub AddToGrid1(Row As Integer)
   With MSFlexGrid2
      .TextMatrix(Row, 0) = invoicenotxt.Text
      .TextMatrix(Row, 1) = Trim(Replace(snotxt.Text, vbCrLf, ""))
      .TextMatrix(Row, 2) = Trim(Replace(particulartxt.Text, vbCrLf, ""))
      .TextMatrix(Row, 3) = Trim(Replace(qtytxt.Text, vbCrLf, ""))
      .TextMatrix(Row, 4) = Trim(Replace(typetxt.Text, vbCrLf, ""))
   End With
 End Sub
Public Sub AddToGrid2(Row As Integer)
   With MSFlexGrid3
      .TextMatrix(Row, 0) = invoicenotxt.Text
      .TextMatrix(Row, 1) = Trim(Replace(date1txt.Text, vbCrLf, ""))
      .TextMatrix(Row, 2) = Trim(Replace(dcnotxt.Text, vbCrLf, ""))
      .TextMatrix(Row, 3) = Trim(Replace(date2txt.Text, vbCrLf, ""))
      .TextMatrix(Row, 4) = Trim(Replace(supnotxt.Text, vbCrLf, ""))
      .TextMatrix(Row, 5) = Trim(Replace(pnotxt.Text, vbCrLf, ""))
```

```
            Dim s, s2
            Dim s3 As Integer
            s2 = Val(qtytxt.Text)
            MsgBox s2
            Set r = New ADODB.Recordset
            s = "select * FROM stock_master where product_name ='" & particulartxt.Text
& "' and weight ='" & weighttxt.Text & "' and type ='" & typetxt.Text & "'"
            r.Open s, cn, adOpenDynamic, adLockOptimistic
            MsgBox r("qty")
            s3 = r("qty")
            s3 = s3 + s2
            Dim rr1 As ADODB.Recordset
            Dim ss1 As String
            Set rr1 = New ADODB.Recordset
            s1 = "update stock_master set qty ='" & s3 & "' where product_name ='" &
particulartxt.Text & "' and weight ='" & weighttxt.Text & "' and type ='" & typetxt.Text
& "'"
            rr1.Open s1, cn, adOpenDynamic, adLockOptimistic
          Else
            r11.AddNew
                r11("product_no") = mrid
                r11("product_name") = particulartxt.Text
                r11("qty") = qtytxt.Text
                r11("weight") = weighttxt.Text
                r11("type") = typetxt.Text
                r11("rate") = amounttxt.Text
            r11.update
          End If
End Sub


Private Sub weighttxt_LostFocus()
Form20.Show
Form20.Left = 4500
End Sub
```