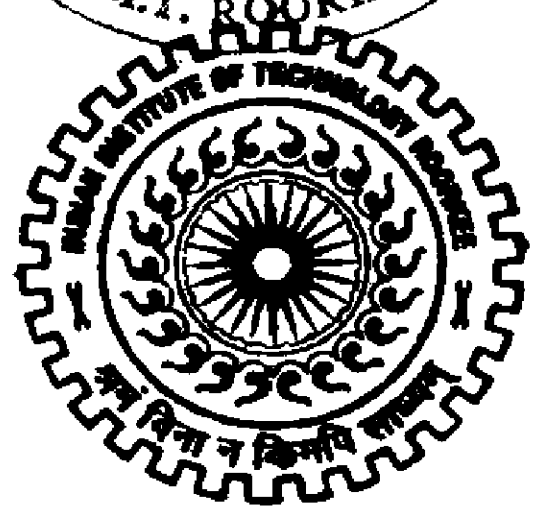
SALES AND BAR CODE PRINTING FOR ISMA

A DISSERTATION

Submitted in partial fulfilment of the
requirements for the award of the degree
of
MASTER OF COMPUTER APPLICATIONS

By

SANTOSH KUMAR SINGH



DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE-247 667 (INDIA)

MAY, 2002

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in this project entitled **"SALES AND BARCODE PRINTING FOR ISMA"** in partial fulfillment of the requirement for the award of the degree of Master of Computer Applications, submitted in the Department of Mathematics of the Indian Institute of Technology, Roorkee, is an *authentic record of my work carried out in the period from Jan 2002 to May 2002,under* the supervision and guidance of **Dr. R.P. Maheshwari**, Assistant Professor, Department of Electrical Engineering, **Mr. Sujit Pal Singh,** Project Manager (Cosmic Softech Ltd.) , **Mr. Gurmit Singh Ahluwalia,** Project Leader (Cosmic Softech Ltd.).

The matter embodied in this project has not been submitted by me for the award of any other degree.

Date: 30/5/2002
Place :IIT,Roorkee.

Santosh Kumar Singh
(Santosh kumar Singh)

CERTIFICATE

This is to certify that the above statements made by the candidate are correct to the best of our knowledge and belief.

Dr. R.P. Maheshwari
Assistant Professor,
Department of Electrical Engineering
IIT, Roorkee

Date: 06.06.2002
Place: Roorkee

SPSuri

Mr. Sujit Pal Singh
Project Manager,
Cosmic Softech Ltd.
New Delhi

Date: 30th May 200
Place: New Delhi

H. G. Sharma 7.6.02
Professor & Head
Department of Mathematics
I.I.T. Roorkee-247 667

CITeS

Cosmic InfoTech Solutions
B1/E-11, Mohan Industrial Estate,
Mathura Road,
New Delhi-110044. India
Tel.: +91-11-6941934, 6943140
6950829-34
Fax: +91-11-6945323
www.citesindia.com

COMPANY CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that the project titled **"Sales and Barcode Printing for ISMA"** being submitted by **Mr. Santosh Kumar Singh,** a student of Indian Institute of Technology, Roorkee in partial fulfilment of the award of the degree of Masters of Computer Applications, has been completed under my supervision at Cosmic Softech Ltd. (formerly Cosmic Infotech Solutions), New Delhi during the period from Jan-2002 to May-2002.

Sujit Pal Singh
Project Manager
Cosmic Softech Ltd.
New Delhi
Date : 30/05/02

ACKNOWLEDGEMENT

It is my proud privilege to express my profound gratitude to my guide **Dr. R.P. Maheshwari**, Assistant Professor, Department of Electrical Engineering, **Mr. Sujit Pal Singh,** Project Manager, Cosmic Softech Ltd., **Mr. Gurmit Singh Ahluwalia,** Project Leader, Cosmic Softech Ltd., for their invaluable inspiration, guidance and continuous encouragement throughout this project work.

I express my sincere thanks to **Prof. H.G. Sharma**, Head, Department of Mathematics, and other staff members of Department of Mathematics.

I also acknowledge **Mr. Tara Chand Sharma, Mr. Shyam Pratap Singh Shekhawat, Mr. Joseph Jacob, Mr. Manish Kumar Mittal, Mr. Kamal Kishore Ahuja** Team Member of the project team for providing the support and inspiration throughout the project work.

Finally, my greatest thanks to my parents for their patience and understanding.

Santosh Kumar Singh
06/06/02
Santosh Kumar Singh
MCA (Final Year)
IIT Roorkee

ABSTRACT

Every production company wants to automate its inventory management, sales management and their analysis. If company has large networks of its operation then it is necessary to update all the information related to sales and inventory from different offices, located at different places, to home office on daily basis to keep operation effective and up-to-date.

The work presents sales and bar code printing modules for ISMA (Inventory Sales-Management and Analysis system). Sales module provides facilities such as generating bill of sales, generating invoice at home office after receiving the bill of sale detail from other offices. It facilitates to make various sales schemes that are used when making bill of sale. This module also generates authorization number which is required by sales persons at the time of generating bill of sale to change fields like discount, term, maximum sale amount and also when doing memorandum sale. Bar code module provides interface so that user can easily generate bar code of inventory ID, can print price, vendor ID, style and description of inventory on a label of desired format.

The work has been developed using Visual Basic 6.0 as a front-end, SQL server 7.0 as a back-end and Win32 API on Windows 98 platform. C.Itoh S4 thermal printer has been used to generate bar code.

Table of Contents

Chapter – 1 Introduction

1.1 About the Company

Cosmic Softech Ltd. formerly known as, Cosmic InfoTech Solutions (CITeS) is a part of the Maharishi Group, acclaimed for its philanthropic activities worldwide. The diversified operations of the group are managed from its establishment in more than 26 countries, including India, USA, UK, Germany, Holland, Russia, Japan, Brazil and Australia. Cosmic Softech Ltd.' dizzying range of cutting-edge software solutions include all kind of web-related customized software application, e-commerce, on-site consultation, offshore development, and telecommunications. Recently it entered into a strategic alliance with CIT Solutions of Iowa, USA - the globally acclaimed developer of state-of-the-art telecom systems.

The one of the well-known product of Cosmic Softech Ltd. is development of e-book over palmtop, which is being used by doctors for quick reference of medicines. Cosmic Softech Ltd. also develops applications for handheld devices like Palmtop, Handheld PC.

1.2 About the project

The project ISMA (Inventory-Sales Management and Analysis) system is developed for Richa Creation. Richa Creation is a jewelry importer and wholesaler in the United States of America. The company has its manufacturing facilities in New Delhi from where it imports the jewelry. Richa Creation has its network of salespersons called REP, who are assigned to take the inventory to the customers in their respective regions. They display the items to the customers, accept orders, and raise the Bill of Sale to the customers. The HOME office in return generates the invoice after receiving the Bill of Sale from the REP.

HOME office has a computerized system of generating the invoices and managing the inventory. Lately Richa Creation realized that they have outgrown this system, which was posing problems for them in expanding their business. Hence, they decided to get a new system developed for their expanding business.

Richa Creation decided to make its inventory management, sales analysis and other operations very effective. ISMA fulfills all their requirements. In inventory management, new inventory can be added with all its associated information and automatic generated inventory ID. Data from Files having inventory information can be imported to database. Transfer of inventory from REPs to NSM, NSM to REP and from NSM to HOME can be handled easily. It can be analyzed that which inventory is with which REP. It also can be analyzed that which inventory is out of stock and which inventory has not been sold for some days. Administrator can define closeout days and if an inventory is not sold within those days then it is declared as closeout inventory.

In sales management, various sales schemes can be generated to increase the sales and these schemes may be restricted for some customers. REP can generate Bill of Sale and uploads Bill of Sale details to HOME so that invoice can be generated without any delay. REPs do not have privilege to generate invoice, only HOME can generate invoice. Exchange of sold items can be handled easily if these items are sold under such type of scheme.

Administrator can assign different privileges to different users. Administrator maintains all the master tables, declares closeout days for inventory, defines commissions for REPs, fast movers, slow movers etc.

Reports can be generated for inventory, sales, returns, customers, orders, search result and data analysis. In data analysis different types of analysis can be done. Graphs can be generated REPs versus sales, month versus sales etc.

In short, Inventory-Sales Management and Analysis System [1] provides the user and manager a fast way to handle multi store inventory transfers from REPs to NSM, NSM to REP and from NSM to HOME, manage physical inventory, maintain turnover of sales and provide trend/analysis of the style in demand along with the reasons behind it. The administrator can maintain the database and also import the new inventory details.

The company plans to give all their REPs a Laptop with new application loaded on them to simplify their work of generating Bill of Sale from their own system and post all details to the HOME Server. Thus delays in making invoices and analyzing data can be drastically cut down.

1.3 System Design

The scope of Designing and Development of the ISMA has been divided in the following modules:

Phase One:

- Inventory
- Sales
- Returns
- Search
- Orders
- Reports
- Data Analysis
- Administration

Phase Two:

- Web Site Hosting

Phase Three:

- Automation of the Manufacturing Unit

The integrated diagram for all the three phases of the project is shown in figure [1.1] [1].

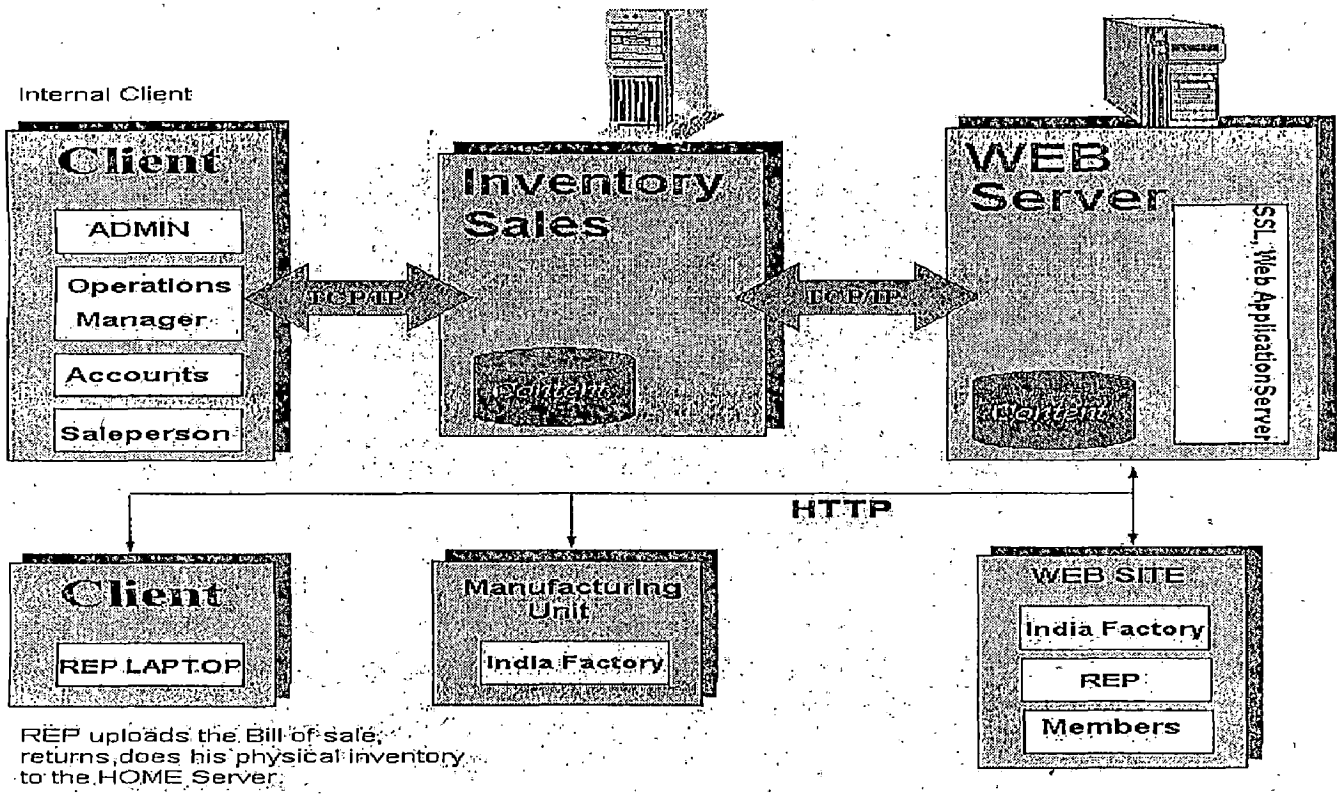


Figure [1.1] Integrated diagram of Phase 1,Phase 2 and Phase 3

1.4 Definitions, Acronyms, and Abbreviations

S. No.	Term/Acronym/Abbreviation	Meaning
	User	Person responsible for particular module
	Administrator	User with power to manage other users and do administrative jobs
	RDBMS	Relational Database Management System
	REP	Salesperson who is assigned a territory for the sale operations
	NSM	National Sales Manager, person to whom a REP reports
	Credit Memo	Return invoice generated when a Customer makes a Return.
	Bill of Sale	REP and HOME generates the Bill of Sale when any sale is made to a customer this is not a invoice
	Return Bill of Sale	REP and HOME generates the Bill of Sale when any Return is made by a customer which is not an invoice
	Transfer	When items are transferred from HOME to a REP's line or NSM's line/ or NSM transfers to the REP/ REPs or NSM transfer to HOME
	CASE#	A unique number called as the CASE number (CASE#) identifying every REP.
	HOME	Head Office of Richa Creation.
	SKU	Unique number of each inventory as generated by the system.
	ASCII	American Standard Code for Information and Interchange.
	Closeout	Ageing inventory or inventory when analyzed by Richa Creation as non-saleable is sold on discounted price.

1.5 Product Functions

Functions of the application can be described in brief as follows.

- **Inventory Module**
  - User menu is displayed according to his privileges.
  - New inventory can be entered into the system with all the details of the item.
  - Inventory details from an ASCII file / excel sheet can be imported into the system.
  - Cost Sheet of the inventory from ASCII file can be imported into the system.
  - Bar code for the new inventory can be generated by the system.
  - Transfer of inventory from REP to HOME or HOME to REP or HOME to NSM can be easily carried out.
  - HOME Server can inform the REPs about any transfers made from HOME location to REP or from NSM to REP.
  - Inventory details from the REPs at remote locations can be uploaded to HOME server.
  - Inventory Room Transfer Details can be captured by the system.
  - Physical inventory can be validated with the system inventory.
  - Entering the search criteria can search inventory.

- **Sales and Return Module**
  - User menu is displayed according to his privileges.
  - The REP and HOME can capture new sales in the Bill of Sale.
  - All Sales will have a Bill of sale.
  - The REPs at remote locations can upload Bill of Sale data to the HOME server.
  - HOME server can also download the Bill of Sales from the REPs' system.
  - Sales can also be made directly from the HOME location.
  - The HOME office generates the INVOICE from the Bill of Sale.
  - Invoice generated can be printed and posted to the Customer.
  - Returns can be reconciliation from the customers, or manufacturing defects, or due to some other reasons.

- The customer can make returns to REP or send directly to HOME location.
- REPs generate return Bill of Sale.
- HOME Server informs the REPs about their reconciliation and customer credentials.

- **Search / Lookup Module**
  - User menu is displayed according to the user's privileges.
  - User can search the module by entering the search criteria.
  - For each search screen possible search criteria are defined like Customer details can be searched by entering the criteria's like customer name, phone number etc.
  - Report module can also be called from other modules depending on the module requirements.

- **Reports and Data Analysis Module**
  - User menu is displayed according to the user's privileges.
  - User can select reports to be generated and also enter the conditions for report generation.
  - Managers and Administrator can do the data analysis.
  - Manager can also make decisions based on the analysis made.
  - Inventory can be declared as a Bestseller or Slow seller for the REP / HOME.

- **Orders Module**
  - User menu is displayed according to the user's privileges.
  - Special orders from customers can be entered in the system and information sent to the factory.
  - Factory can update the HOME Server with special order status.
  - User can view the status of the pending orders of a specific style.
  - Duplication of orders can be avoided by viewing the pending order status of the specific style.
  - Manager can analyze the inventory data and reorder the inventory.

- **Administration Module**
  - User menu is displayed according to the user's privileges.
  - Administrator can maintain the master table.
  - Administrator can maintain the user database with their privilege levels.
  - Administrator can enter details of the REP's Targets.
  - Database backup and recovery procedures can be controlled from this module.
  - Managers can decide and make programs and delivery line for the Sale module. Administrator will be responsible for implementing these changes in the system.
  - HOME server uploads the information of reconciliation and customer credentials to the REP's Laptop.
  - Administrator can intimate the REPs about the transfer details; REPs can confirm the receipt of the transfer.
  - Administrator can declare an inventories as closeout
  - The Administrator can validate REP's Physical inventory data once uploaded to the HOME server with the HOME physical inventory for that REP.
  - Transfers from NSM to REP / HOME & vice versa can be privilege functions that can be enabled / disabled by the administrator.
  - Administrator can maintain details of the REPs like insurance and target details.

To develop the functions of Inventory-Sales Management and Analysis system there is a need of database management system to store large amount of data and a user interface to access and manipulate the stored data.

Chapter – 2 Language and DBMS Used

The following language and database management system (DBMS) have been used to develop the Sales and Bar Code module.

- Front End: Visual Basic 6.0
- Back End: SQL Server 7.0

And the command reference manual of C.ITOH thermal printer is used for reference of printer commands.

Microsoft Visual Basic 6 is the newest version of the popular programming language. With its new features, Visual Basic is an even stronger contender in the application development arena than ever before.

The Visual Basic environment is great for creating almost any type of application. We can develop robust stand-alone applications, games, and utilities in less time than it takes in other language. Internet-enabled applications can be easily developed. When used in conjunction with the Windows API, almost anything can be done with project development.

2.1 Data Access Technology Used

In Visual Basic, [2] three data access interfaces are available: ActiveX Data Objects (ADO), Remote Data Objects (RDO), and Data Access Objects (DAO). A data access interface is an object model that represents various facets of accessing data. Using Visual Basic, programmer can programmatically control the connection, statement builders, and returned data for use in any application.

Why are there three data access interfaces in Visual Basic? Data access technology is constantly evolving, and each of the three interfaces represents a different state of the art. The latest is ADO, which features a simpler — yet more flexible — object model than either RDO or DAO.

ADO is Microsoft's premier data access technology. The ADO data access technology and its partner OLE DB comprise the recommended solution for all data access.

ADO is designed as an easy-to-use application level interface to Microsoft's newest and most powerful data access paradigm, OLE DB. OLE DB provides high-performance access to any data source, including relational and non-relational databases, email and file systems, text and graphics, custom business objects, and more. ADO is implemented for minimal network traffic in key Internet scenarios, and a minimal number of layers between the front-end and data source — all to provide a lightweight, high-performance interface.

2.1.1 OLE DB

OLE DB is a new low-level interface that introduces a "universal" data access paradigm. That is, OLE DB is not restricted to ISAM, Jet, or even relational data sources, but is capable of dealing with any type of data regardless of its format or storage method. In practice, this versatility means you can access data that resides in an Excel spreadsheet, text files, or even on a mail server such as Microsoft Exchange.

In Visual Basic 6.0, you leverage the flexibility of OLE DB through ADO, the programmer interface to OLE DB.

2.1.2 OLE DB and ADO

OLE DB is not designed to be accessed directly from Visual Basic due to its complex interfaces. Instead ActiveX Data Objects (ADO) encapsulates and exposes virtually all of OLE DB's functionality.

2.1.3 Accessing Data with ADO

ADO is the single data interface you need to know for all client/server and Web-based data access solutions. One of the strengths of ADO is that it can expose and use the unique properties of each data provider. No matter what data source is used, ADO is totally flexible and adaptable to your application's data access requirements. An important feature of using ADO is the availability of advanced record set cache management with Remote Data Services (RDS). RDS provides optional data caching on the client workstation. With RDS, you can easily marshal data back and forth between the server and client. For example, your application may use a large client-side result

set. This reduces the number of requests for data from the client-side application to the server, improving both the actual performance and the perceived performance of the client-side application.

A typical ADO-based application uses the following operations to access a data source.

- **Create the Connection object:** Specifies the connection string with information such as data source name, user identification, password, connection time-out, default database, and cursor location. A Connection object represents a unique session with a data source. You can even control transactions through the Connection object using the BeginTrans, CommitTrans, and RollbackTrans methods.

- **Open the connection:** Opens the ADO connection to the data source.

- **Execute an SQL statement:** Once the connection is open, you can run a query. You can run this query asynchronously and also choose to process the query's result set asynchronously; ADO signals the cursor driver to populate the result set in the background. This lets your application perform other processes without waiting.

- **Use the result set** The result set is now available to your application. Depending on the cursor type, you can browse and change the row data at either the server or client side.

- **Terminate the connection** Drops the connection to the data source.

Although ADO objects have many properties and methods, using ADO is really as simple as it looks. ADO definitely represents the future of data access technology.

2.2 SQL Server 7.0

Microsoft® SQL Server™ [3] version 7.0 is a significant new release of SQL Server. With version 7.0, SQL Server has been redesigned to provide important new server architecture and graphical administration features, while maintaining ANSI and SQL

Server 6.*x* compatibility. Enhancements have also been made to SQL Server architecture, server functionality, and development tools.

Microsoft® SQL Server™ is a Structured Query Language (SQL) based, client/server relational database. Each of these terms describes a fundamental part of the architecture of SQL Server.

Database

A database is similar to a data file in that it is a storage place for data. Like a data file, a database does not present information directly to a user; the user runs an application that accesses data from the database and presents it to the user in an understandable format.

Database systems are more powerful than data files. The data is more highly organized. In a well-designed database, there are no duplicate pieces of data that the user or application has to update at the same time. Related pieces of data are grouped together in a single structure or record, and relationships can be defined between these structures and records.

When working with data files, an application must be coded to work with the specific structure of each data file. In contrast, a database contains a catalog that applications use to determine how data is organized. Generic database applications can use the catalog to present users with data from different databases dynamically, without being tied to a specific data format.

A database typically has two components: the files holding the physical database and the database management system (DBMS) software that applications use to access data. The DBMS is responsible for enforcing the database structure, including:

- Maintaining the relationships between data in the database.
- Ensuring that data is stored correctly, and that the rules defining data relationships are not violated.
- Recovering all data to a point of known consistency in case of system failures.

Relational Database

There are different ways to organize data in a database but relational databases are one of the most effective. In a relational database, data is collected into tables (called relations in relational theory).

A table represents some class of objects that are important to an organization. For example, a company may have a database with a table for employees, another table for customers, and another for stores. Each table comprises columns and rows. Each column represents some attribute of the object represented by the table. For example, an **Employee** table would typically have columns for first name, last name, employee ID, department, pay grade, and job title. Each row represents an instance of the object represented by the table. For example, one row in the **Employee** table represents the employee who has employee ID 12345.

When organizing data into tables, there are many different ways to define tables. Relational database theory defines a process, normalization, which ensures that the set of tables you define will organize your data effectively.

Client/Server

In a client/server system, the server is a relatively large computer in a central location that manages a resource used by many people. When individuals need to use the resource, they connect over the network from their computers, or clients, to the server. Examples of servers are:

- Print servers

  Manage the printers used by a team or unit.

- File servers

  Store large files used by a team or unit by using large disk drives.

- E-mail servers

  Run a company's e-mail system.

In client/server database architecture, the database files and DBMS software reside on a server. A communications component is provided so that applications can run on separate clients and communicate to the database server over a network. The SQL Server communication component also allows communication between an application running on the server and SQL Server.

Server applications are usually capable of working with several clients at the same time. SQL Server can work with thousands of client applications simultaneously. The server has features to prevent the logical problems that occur if a user tries to read or modify data currently being used by others.

While SQL Server is designed to work as a server in a client/server network, it is also capable of working as a stand-alone database directly on the client. The scalability and ease-of-use features of SQL Server allow it to work efficiently on a client without consuming too many resources.

2.2.1 Client/Server Architecture

Microsoft® SQL Server™ is designed to work effectively in a number of environments:

- As a two-tier or multi-tier client/server database system
- As a desktop database system

Client/server systems are constructed so that the database can reside on a central computer, known as a server (A computer on a local area network (LAN) that controls access to resources, such as files, printers, and communication devices.), and be shared among several users. Users access the server through a client or server application:

- In a two-tier client/server system, users run an application on their local computer, known as a client that connects over a network to the server running SQL Server. The client application runs both business logic and the code to display output to the user, and is also known as a thick client, see figure [2.1].
- In a multi-tier client/server system, the client application logic is run in two locations:
    - The thin client is run on the user's local computer and is focused on displaying results to the user.
    - The business logic is located in server applications running on a server. Thin clients request functions from the server application, which is itself a multithreaded application capable of working with many concurrent users.

The server application is the one that opens connections to the database server and can be running on the same server as the database, or it can connect across the network to a separate server operating as a database server. See figure [2.1].

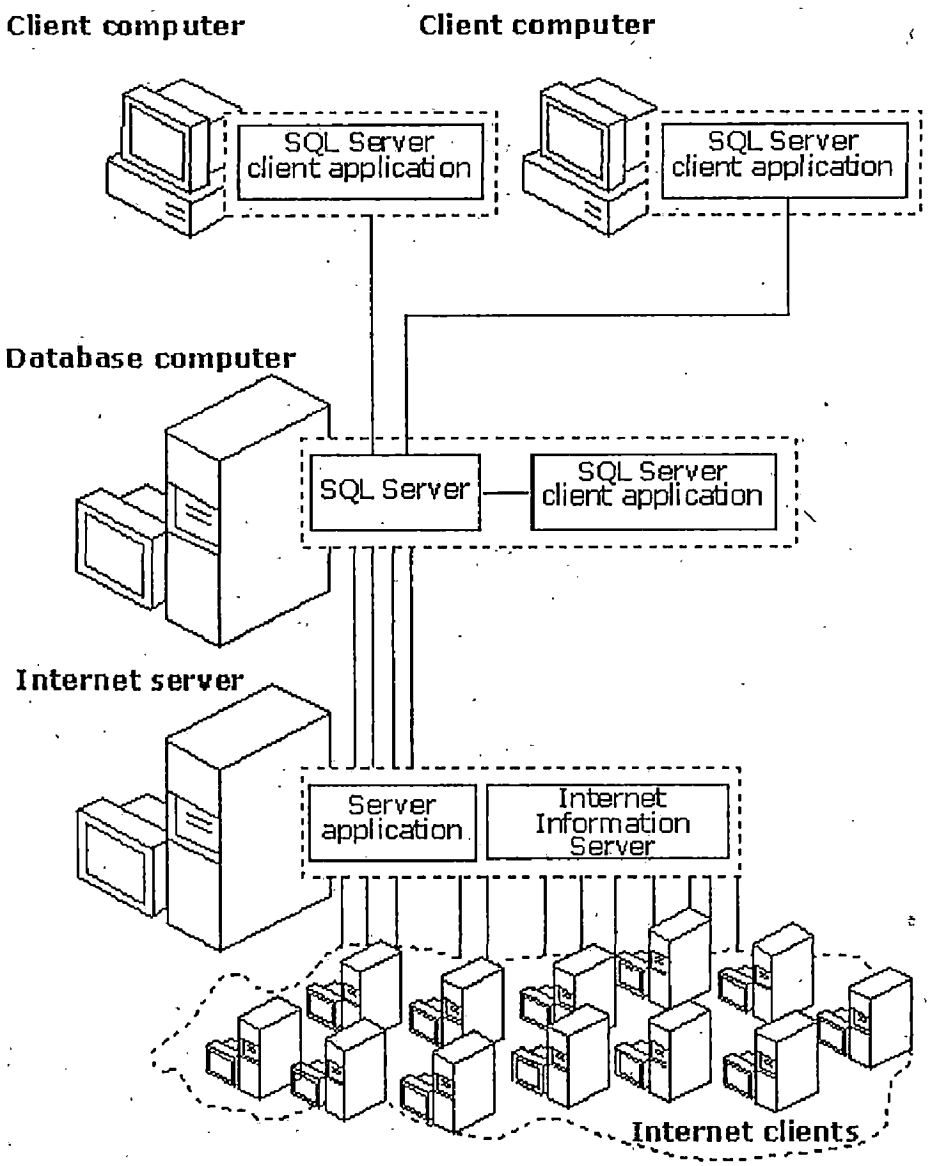


Figure [2.1] Two-Tier and Multi-Tier Architecture

This is a typical scenario for an Internet application. For example, a server application can run on a Microsoft Internet Information Services (IIS) and service thousands of thin clients running on the Internet or an intranet. The server application uses a pool of connections to communicate with a copy of SQL Server. SQL Server can be installed on the same computer as IIS, or it can be installed on a separate server in the network.

In large client/server systems, thousands of users may be connected to a SQL Server installation at the same time. SQL Server has full protection for these environments, with safeguards that prevent problems such as having multiple users trying to update the same piece of data at the same time. SQL Server also allocates the available resources effectively, such as memory, network bandwidth, and disk I/O, among the multiple users.

SQL Server applications can run on the same computer as SQL Server. The application connects to SQL Server using Windows Inter process Communications (IPC) components, such as shared memory, instead of a network. This allows SQL Server to be used on small systems where an application needs to store its data locally.

2.2.2 SQL Server As A Desktop Database Systems

While SQL Server works effectively as a server, it can also be used in applications that need stand-alone databases stored locally on the client. SQL Server can configure itself dynamically to run efficiently with the resources available on a client, without the need to dedicate a database administrator to each client.

When clients use local SQL Server databases, one copy of the SQL Server database engine runs on the client and manages all the SQL Server databases on the client. Applications connect to the database engine in much the same way they connect across the network to a database engine running on a remote server.

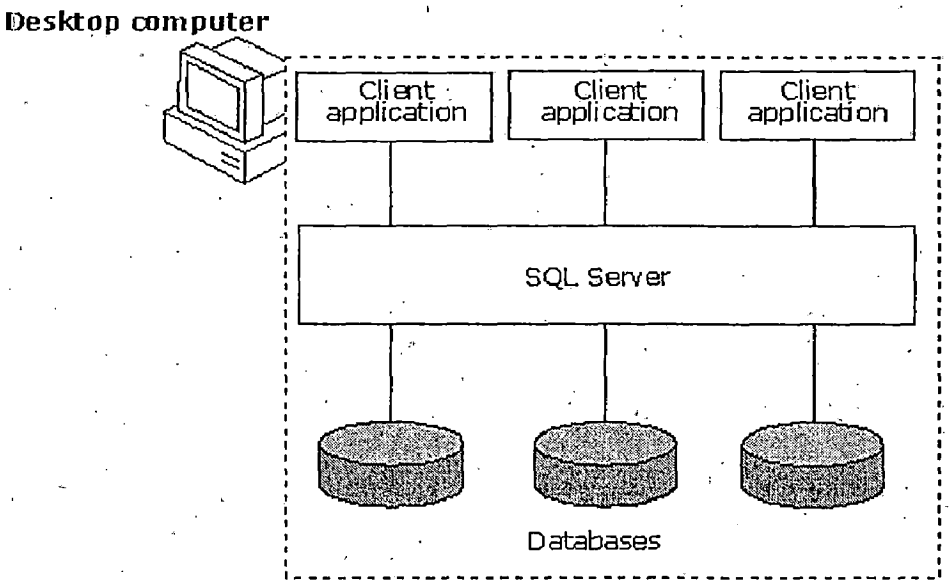


Figure [2.2] SQL Server Architecture for Desktop System

2.2.3 Advantage of using Client/Server Architecture

Having data stored and managed in a central location offers several advantages:

- Each data item is stored in a central location where all users can work with it.

  Separate copies of the item are not stored on each client, which eliminates problems with users having to ensure they are all working with the same information.

- Business and security rules can be defined one time on the server and enforced equally among all users.

  This can be done in a database through the use of constraints, stored procedures, and triggers. It can also be done in a server application.

- A relational database server optimizes network traffic by returning only the data an application needs.

  For example, if an application working with a file server needs to display a list of the names of sales representatives in Oregon, it must retrieve the entire employee file. If the application is working with a relational database server, it sends this command:

  ```
  SELECT first_name, last_name

  FROM employees

  WHERE emp_title = 'Sales Representative'

  AND emp_state = 'OR'
  ```

  The relational database sends back only the names of the sales representatives in Oregon, not all of the information about all employees.

- Hardware costs can be minimized.

Because the data is not stored on each client, clients do not have to dedicate disk space to storing data. The clients also do not need the processing capacity to manage data locally, and the server does not need to dedicate processing power to displaying data.

The server can be stored in a relatively secure location and equipped with devices such as an Uninterruptible Power Supply (UPS) more economically than fully protecting each client.

- Maintenance tasks such as backing up and restoring data are simplified because they can focus on the central server.

2.2.4 Structured Query Language (SQL)

To work with data in a database, you must use a set of commands and statements (language) defined by the DBMS software. There are several different languages that can be used with relational databases; the most common is SQL. Both the American National Standards Institute (ANSI) and the International Standards Organization (ISO) have defined standards for SQL. Most modern DBMS products support the Entry Level of SQL-92, the latest SQL standard (published in 1992).

Transact-SQL

The SQL used in Microsoft® SQL Server™ is known as Transact-SQL. Transact-SQL is central to the use of Microsoft® SQL Server™. All applications that communicate with SQL Server do so by sending Transact-SQL statements to the server, regardless of an application's user interface. Commands sent to Microsoft® SQL Server™ through the database API must comply with the Transact-SQL language. Transact-SQL complies with the Entry Level of the SQL-92 standard, but also supports many more features of SQL-92. It also supports some powerful extensions to the SQL-92 standard.

2.2.5 Features of SQL Server

Microsoft® SQL Server™ supports a set of features that result in the following benefits:

- **Ease of installation, deployment, and use**

  SQL Server includes a set of administrative and development tools that improve your ability to install, deploy, manage, and use SQL Server across several sites.

- **Scalability**

  The same database engine can be used across platforms ranging from laptop computers running Microsoft Windows® 95/98 to large, multiprocessor servers running Microsoft Windows NT®, Enterprise Edition.

- **Data warehousing**

  SQL Server includes tools for extracting and analyzing summary data for online analytical processing (OLAP). SQL Server also includes tools for visually designing databases and analyzing data using English-based questions.

- **System integration with other server software**

  SQL Server integrates with e-mail, the Internet, and Windows.

Chapter – 3 Implementation of Sales Module

3.1 Functions

The sales module has the following sub modules as shown in Figure [3.1]:

1. Exchange
2. Sale Schemes
3. Bill of sale
4. Authorization Number
5. Bill of Sale List
6. Enter Tracking Info
7. Reconciliation Due
8. Generate Order Invoice
9. Search

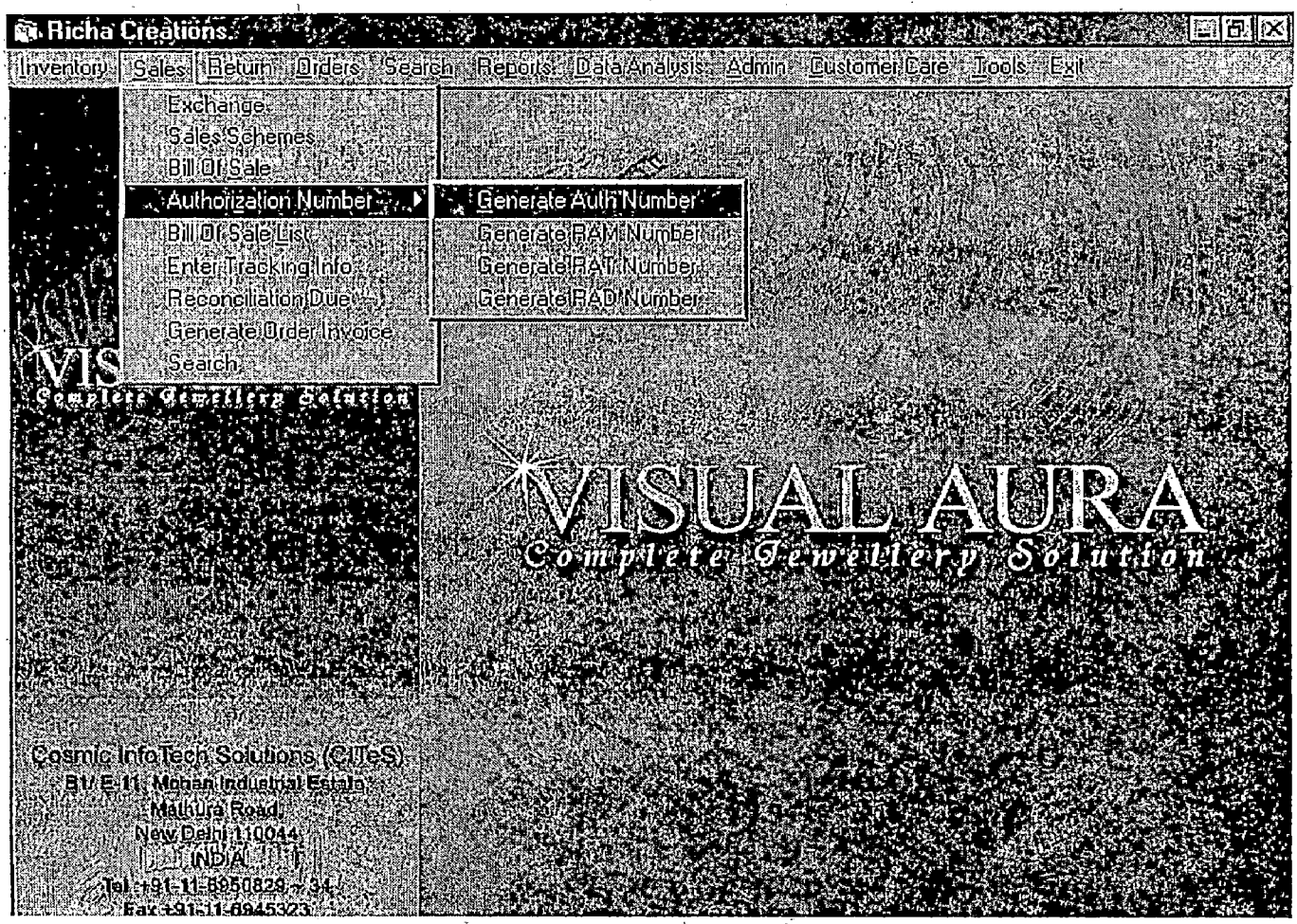


Figure [3.1] Main Screen of the Project

The Bill of Sale sub module captures the details of the sales made to a customer by the REP. When the REP sells a product to the customer he generates a Bill of Sale which is send to the HOME location from his laptop and a copy of the same is mailed. The HOME location generates an invoice by Generate Invoice sub module from the Bill of Sale in the name of the customer and the details of the sale captured in it. REP has to make separate Bill of Sale for Closeouts, Memorandum and general sales. The HOME server updates the closeout details in REP's system periodically.

Authorization Number sub module generates Auth Number, RAM Number, RAT Number and RAD Number. Only HOME can generate these Authorization numbers. REP uses RAM Number when he/she wants memorandum sale, RAD Number when he/she wants to give discount on sale, RAT Number when he/she wants to change term and Auth Number when he/she wants to change Bill Amount Available.

Bill of Sale List sub module lists all the Bill of Sale. Home can select any Bill of sale and can generate invoice. Sales Scheme sub module allows creating various sales schemes that can be used when making Bill of Sale.

The REP uploads the Bill of Sale to the HOME location. This is when the HOME location generates the Sales Invoice. When the Sale is made from the HOME location, the Bill of Sale is also generated.

User can lookup /search for sales information in the Search sub-module. Here the user has to enter his search criteria and the result will be fetched from the corresponding database.

3.2 Generating Bill of Sale

The REP from his laptop will generate Bill of Sale; his system will maintain the customer data like the maximum amount of sale he can make to a customer; the HOME office will update this customer account data. REP has to generate a Closeout Bill of Sale and memorandum bill of sale separately. The REP uploads the Bill of Sale data to the HOME Server.

User can enter the details of the new sales by first finding the details of the customer to which the sale has been made. User can lookup at the customer details and then enter the details of the sale made to him.

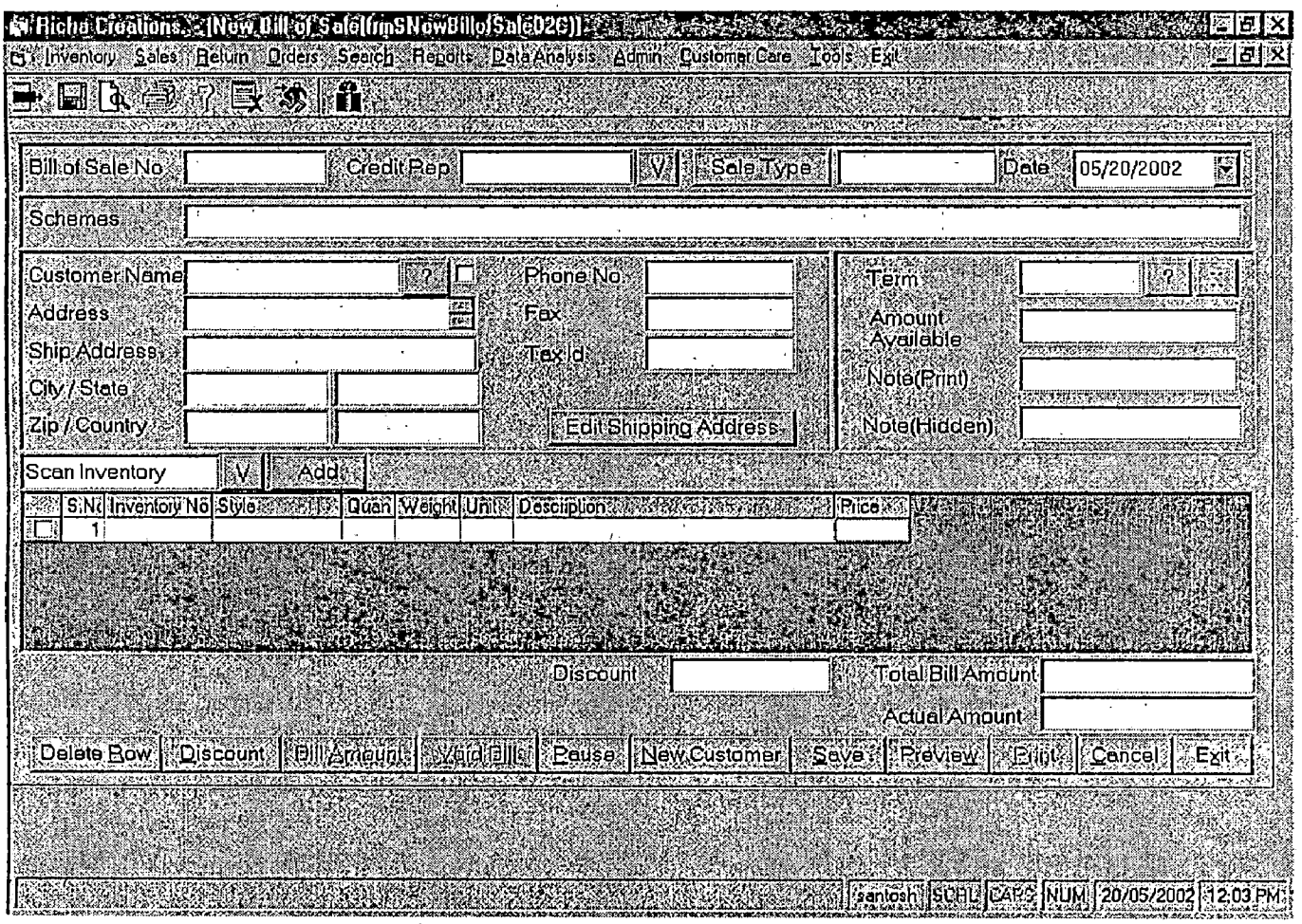


Figure [3.2] Form Interface to generate Bill of sale.

- Sale Type: This button selects Sale Type. There are following sale types available:
  - Whole Sale
  - Retail Sale
  - CloseOut Sale
  - Memorandom Sale

- Lookup Customer (? Button): Pressing this button brings up the Lookup for Customer details Figure [3.6].
- User Name: User Name Displayed on the screen depends on the user who logs in the system; used for maintaining the history of the new user added.
- Date: System date is displayed on the screen.
- Bill of Sale number: Displayed and auto generated by the REP's System in the PPP-XXXXX format where P is the REP's Code and X is the running number.
- REP Name: Display field, shows the name of the REP.
- DATE: Display field, shows the current date.
- New Customer: Pressing this button brings the screen to enter the new customers details.
- Term: Term is entered for sale as agreed by the customer.
- Amount available: Display field and shows the maximum amount of sale that can be made to this customer. The HOME Server updates this information. For a new customer a minimum amount is decided.
- Inventory No.: This is the inventory number picked up from the drop down list. The corresponding details are picked from the REP's inventory database and then by ADD button Inventory is added to the list.
- Discount: REP can allow discount after validation of RAD Number.
- Total Bill Amount: Display field calculated by the summation of the selling price of all items.
- Actual Amount: Display field, Actual Amount after discount if any is displayed here.
- Print: Pressing this button prints the Bill of Sale with all the details captured.
- Save: Saves the Bill of Sale in the database.

- Void the Bill: REP can void the Bill of Sale. He has to mention the reason for voiding.
- Pause: REP can pause the bill of sale.
- Schemes: Schemes made for sale will be displayed according to type of sale selected.
- Cancel: Resets all the fields.
- Note (Print): Notes that will be printed with Bill of Sale.
- Note (Hidden): Notes that will be stored in Database.

3.3 Bill of Sale List

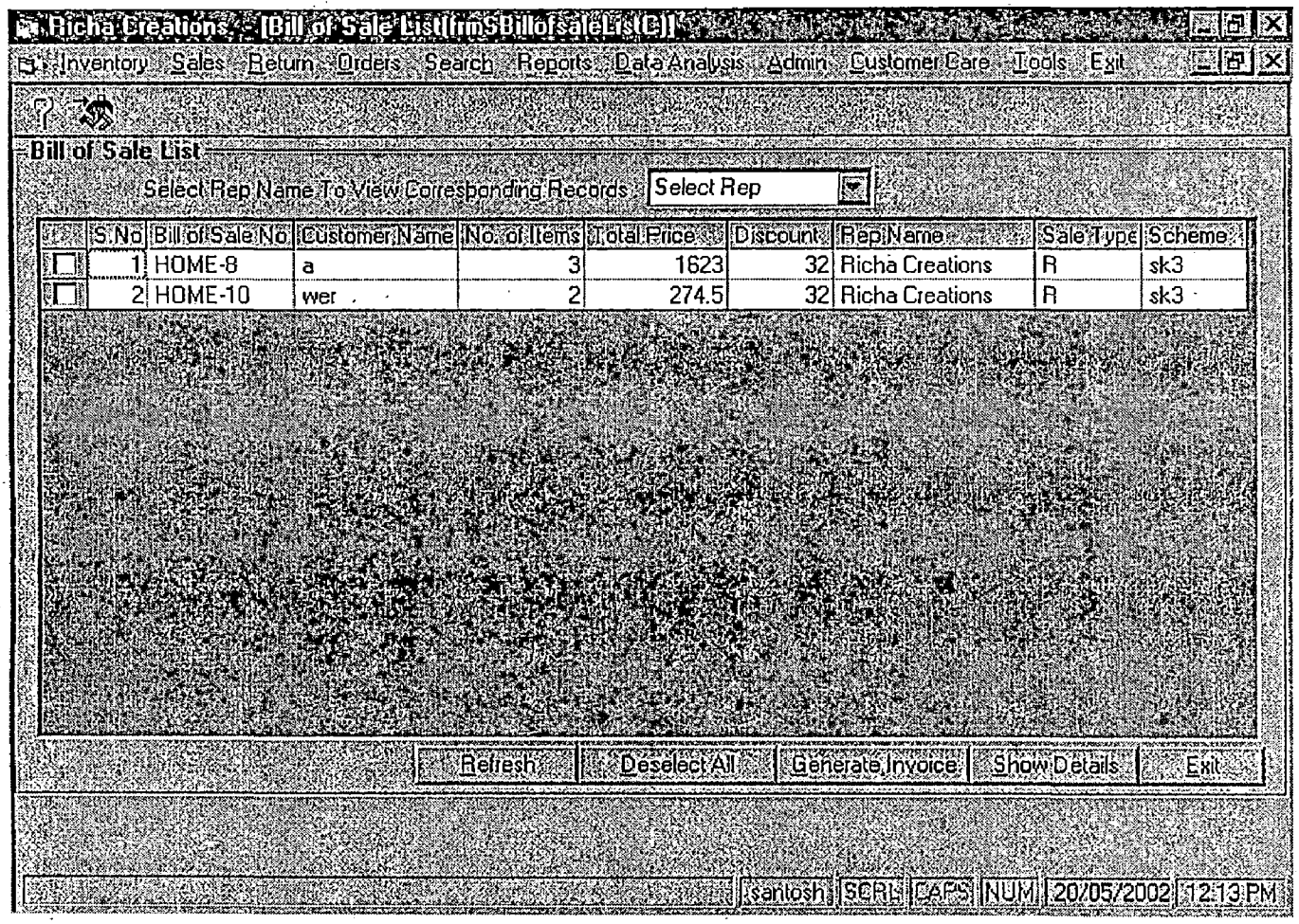


Figure [3.3] Form Interface for Bill of sale list.

This sub module shows the list of Bill of sale made by REP. Only HOME has privilege to see the list of Bill of Sale of any REP. REP can see only Bill of Sales made by him not of others. This sub module has the following functionalities.

- Refresh: Refreshes the data grid.
- Deselect All: Deselects all the rows.
- Generate Invoice: This Button will be visible to only Home not to any REP because only HOME can generate invoice. After pressing this button Generate Invoice form is populated where invoice can be generated Figure [3.4].
- Show Details: Select any row from data grid and then clicking this button will show the Details of Bill of sale where editing also can be done.
- Exit: Exits from this form.

3.4 Generating Invoice

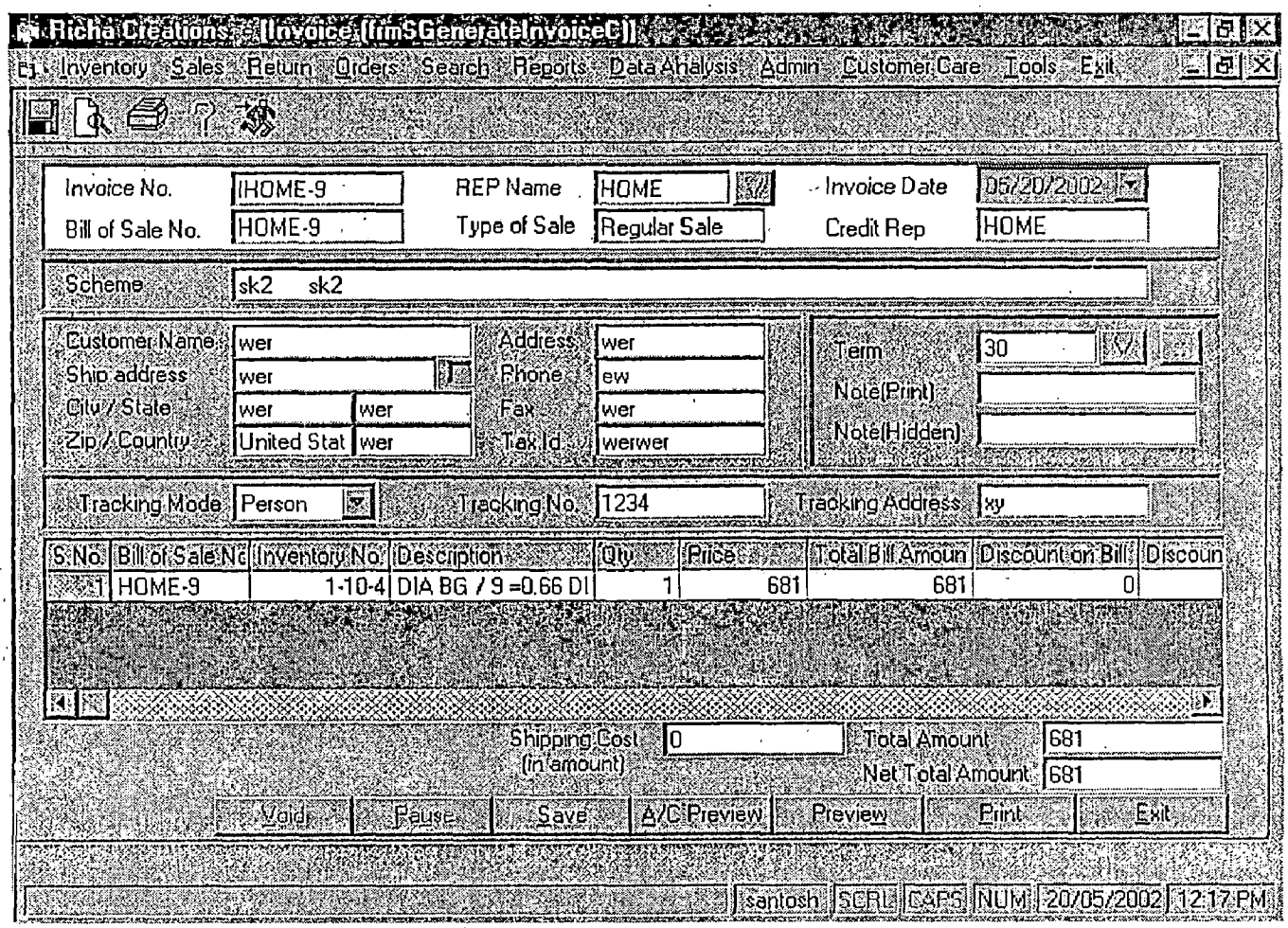


Figure [3.4] Form Interface to generate Invoice.

Invoice is only generated by HOME. Uploaded by the REPs the process of generating the invoice starts when a mail confirmation duly signed by the customer reaches the HOME office. Bill of Sale whose invoice has to generated is selected from the Bill of Sale Form and then Generate Invoice button is pressed to load the generate invoice form which is shown in figure [3.4].

- All the fields except (Note (Print), Note (Hidden), Tracking mode, Tracking No., Tracking address, Shipping Cost) will be populated with information stored with that Bill of Sale.
- Invoice No.: Invoice No. is automatically generated by preceding I with Bill of Sale No.
- Invoice Date: System date displayed on the screen.
- Tracking Mode: Tracking mode is selected from the drop down list. Tracking mode may be any one of the following:
    - Person
    - Telephone
    - DLF
    - FedEX
- Tracking No.: Tracking Number of the shipment is entered here.
- Tracking Address: Tracking address of the shipment is entered here.
- Shipping Cost: Shipping cost is entered if any in this field that is added into total amount.
- Save: Saves the Invoice in the database.
- Pause: HOME can pause the generation of invoice.
- Print: Pressing this button prints the Invoice with all the details captured.
- Void: HOME can void the invoice. He has to mention the reason for voiding.
- Exit: Exits from this form.
- Edit Shipping Address: User can edit shipping address by checking Check box.
- Term Schedule ((::) Button): Shows the Due date and Amount Due according to term.

3.5 Sales Schemes

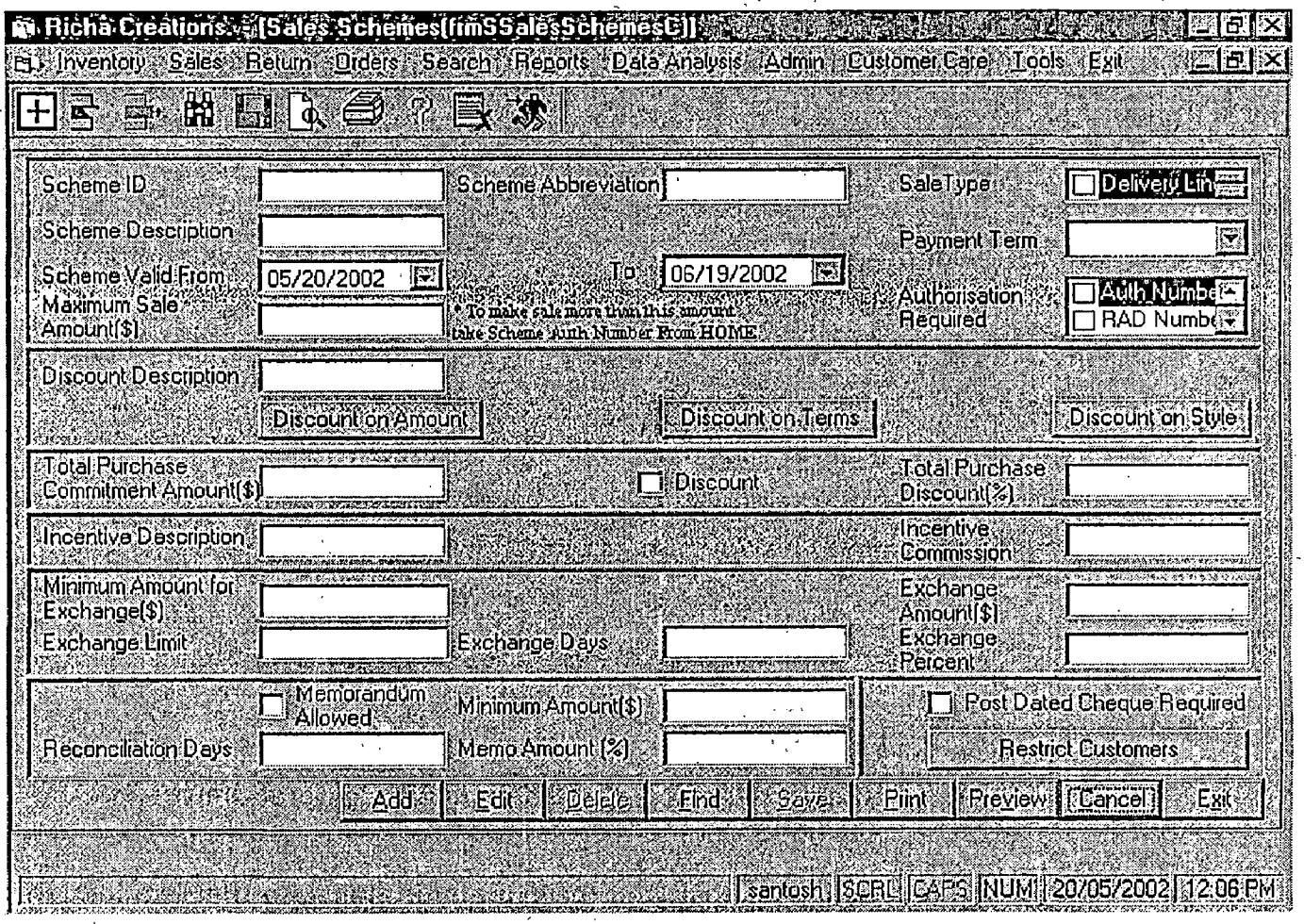


Figure [3.5] Form Interface to make sales schemes.

This module facilitates to make sale schemes for different sale type. Sales schemes can be created using combination of the following optional and mandatory fields.

- Scheme: The name of scheme is given here which is mandatory.
- Scheme Description: The description of scheme is given here which is mandatory.
- Scheme Id: Auto generated field SS+XXX. Where XXX is running number.
- Scheme Valid From: The duration of scheme is specified here.
- Maximum Sale Amount: Maximum sale amount allowed to sale by using this scheme. If REP wants to sale more than this then he/she has to take Authorization number from HOME.
- Sale Type: For which type of sale scheme is being created.

- Payment Term: What payment term will be available for this scheme is entered here.
- Authorization Required: Which authorization numbers will be needed when REP will be making Bill of Sale by using this scheme.
- Discount On Amount: Discount criteria will be specified here for this scheme.
- Discount Description: Description of discount can be entered here.
- Discount On Term: Discount criteria for term can be specified here.
- Discount On Style: Discount criteria for style can be specified here.
- Total purchase commitment amount and Total purchase Discount (%) can be associated with scheme.
- Incentive description and incentive commission can be associated with scheme.
- In the scheme, facility for exchanging item can be provided by Minimum amount for exchange, Exchange limit, Exchange days (interval), Exchange amount (Maximum), Exchange percentage (of total sale).
- Also Can be specified that post-dated cheque is required or not.
- Restrict Customer: Some customers can be restricted from taking advantage of scheme.
- Memorandum sale can be allowed for scheme and then related fields (reconciliation days, minimum amount, memo amount (%)) will be mandatory to enter.
- Find: Find button allows to search schemes and shows their details.

3.6 Lookup for the customer details

Giving any one criteria or combination of any customer search can be done Figure [3.6].

- Customer Name: User can enter the customer name in the Name of customer field to find the customer detail.
- Address: User can enter the address in the address field to find the customer detail.
- City: User can enter the city name in the city field to find the customer detail.
- State: User can enter the state name in the state field to find the customer detail.

- Zip: User can enter the zip in the zip field to find the customer detail.
- Country: User can enter the country name in the country field to find the customer detail.
- Phone Number: User can enter the Phone Number in the Phone Number field to find the customer detail.
- Case: User can enter the CASE name in the Case field to find the customer detail.
- Find/Lookup (button). Pressing this will take us to customer detail screen if customer exits. Otherwise it gives message customer not found.
- Exit: Exits out of this module.
- Cancel: Refreshes the details.

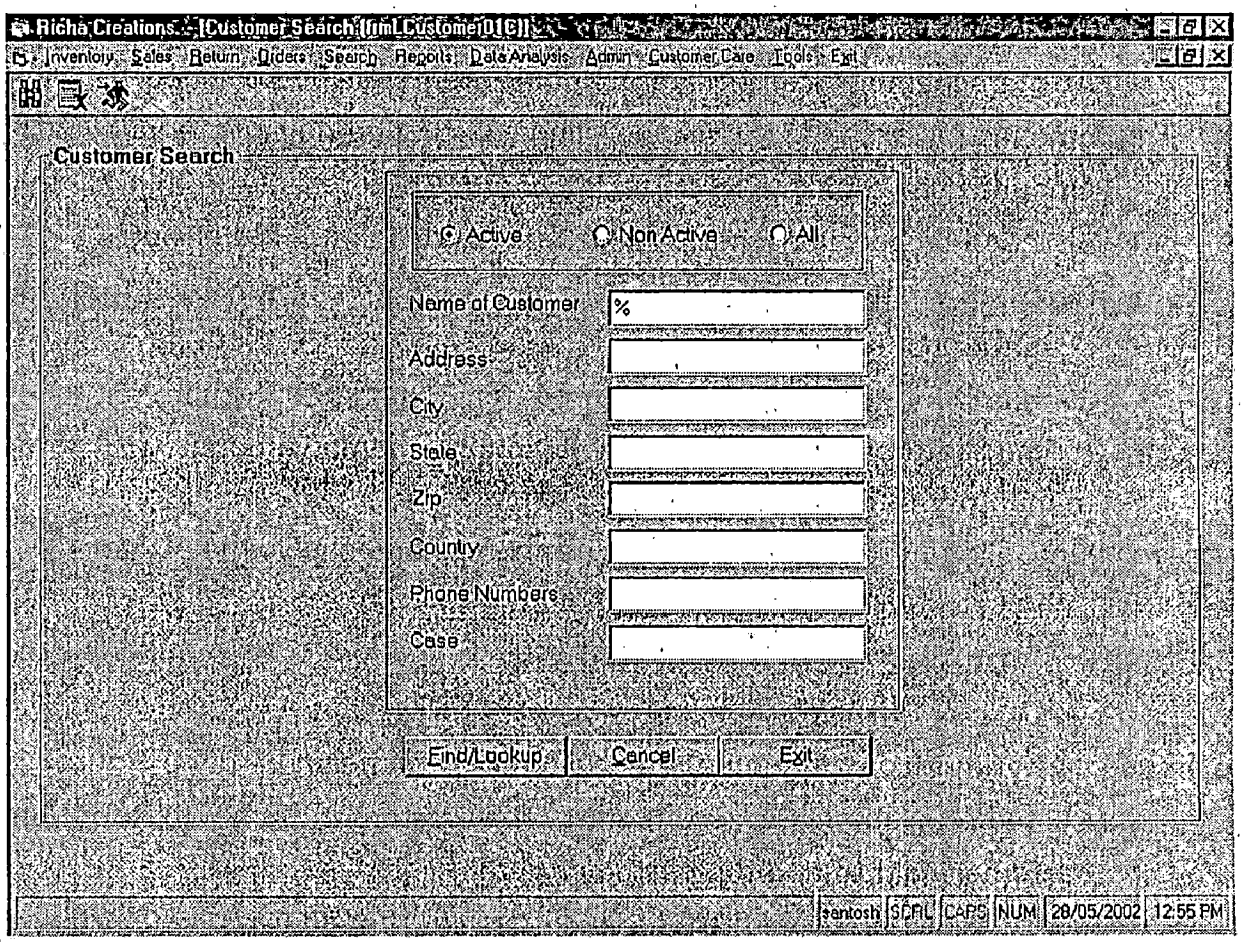


Figure [3.6] Form Interface for Customer search.

Customer Details: On double click of any row of customer details screen Figure [3.6] following fields will be populated from where Customer Lookup button has been pressed.

- Customer Name
- Address
- Ship Address
- City/state
- Zip/country
- Phone
- Tax Id

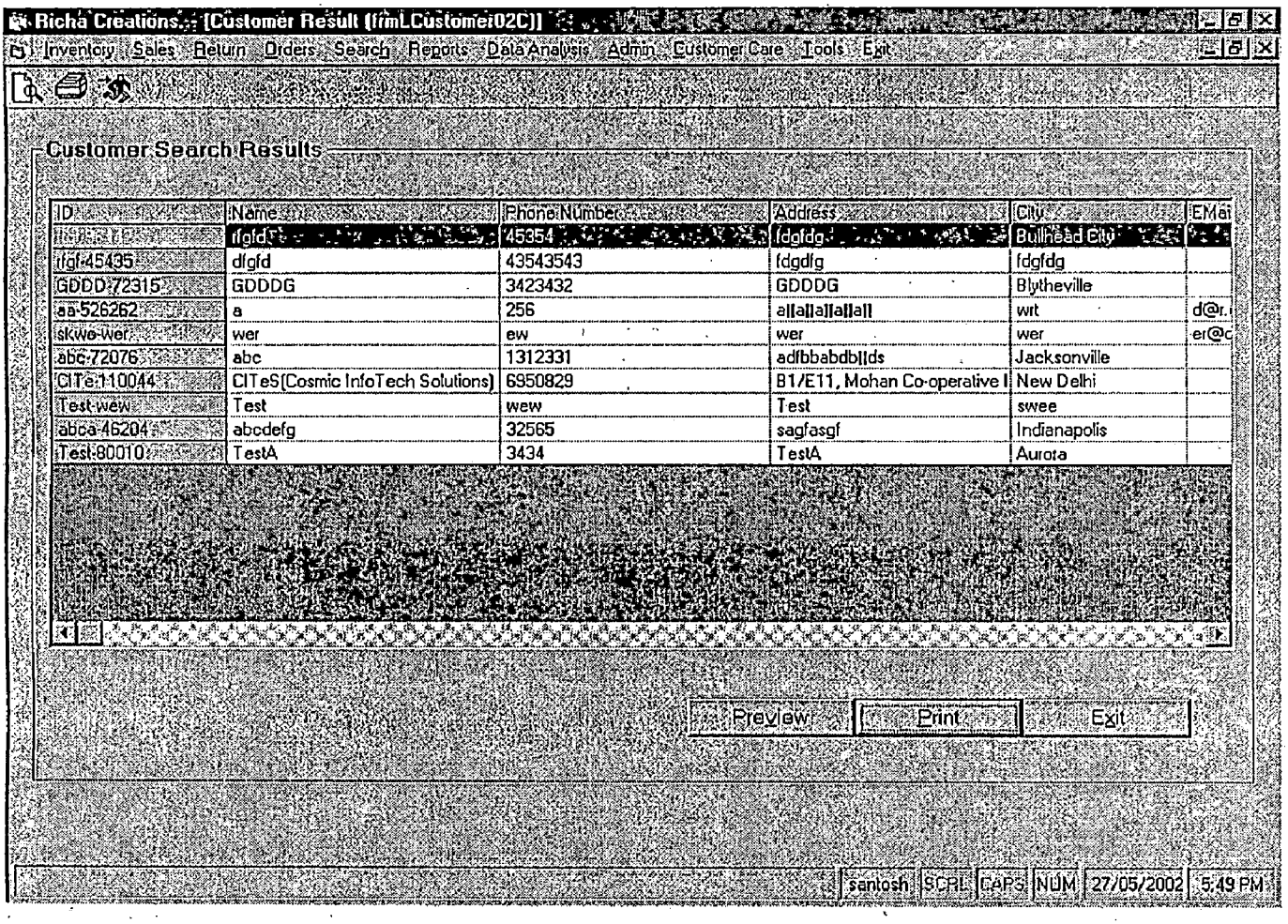


Figure [3.6] Result of Customer search

3.7 Generating Auth Number

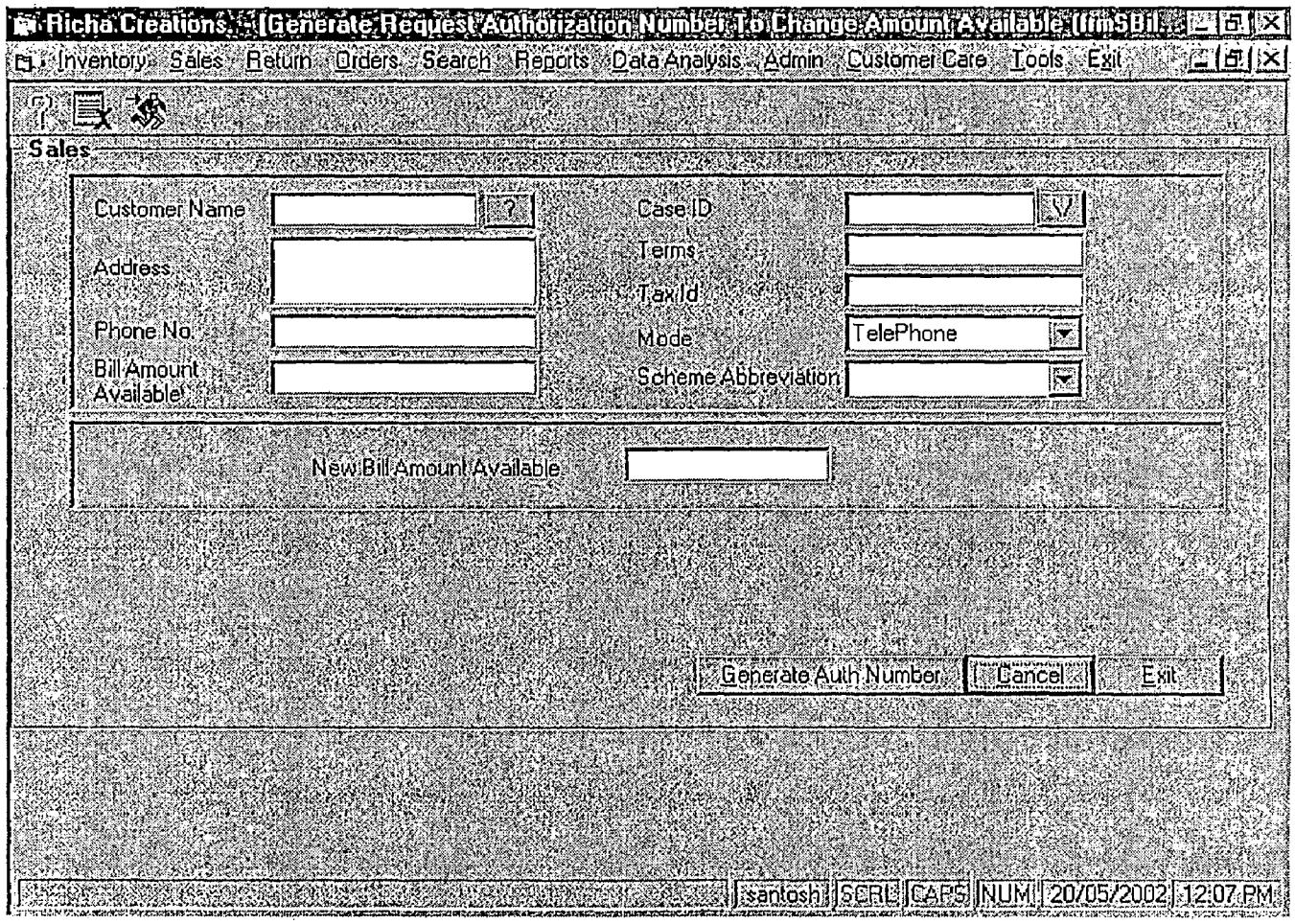


Figure [3.7] Form Interface to generate Auth Number.

This Authorization Number is generated by HOME. When REP wants to sell more than Bill Amount available (Every customer has a Bill Amount Available which is assigned to him when customer's database is created) then he has to take permission from HOME, which is done by Authorization Number. REP asks for Authorization Number and then HOME gives him by Telephone or mail then REP does validation of that number. Without doing validation REP can't make sale more than Bill Amount Available. REP cannot validation any randomly chosen number because by using complex algorithm this Auth Number is generated by HOME and reverse process is done at REP side. The form interface to generate the Auth Number is shown in Figure [3.7].

- Customer Name, address, phone number, bill amount available, term id and tax id are populated by customer search and customer detail process. All these fields are mandatory.
- Proper Case Id, Mode and Scheme Abbreviation (if any) are selected from drop down list.
- New Amount is entered in the New Bill amount available field.
- Generate Auth Number: To generate Authorization Number this button is clicked.
- Exit: Exits out of this module.
- Cancel: Refreshes all the fields.

3.8 Generating RAM Number

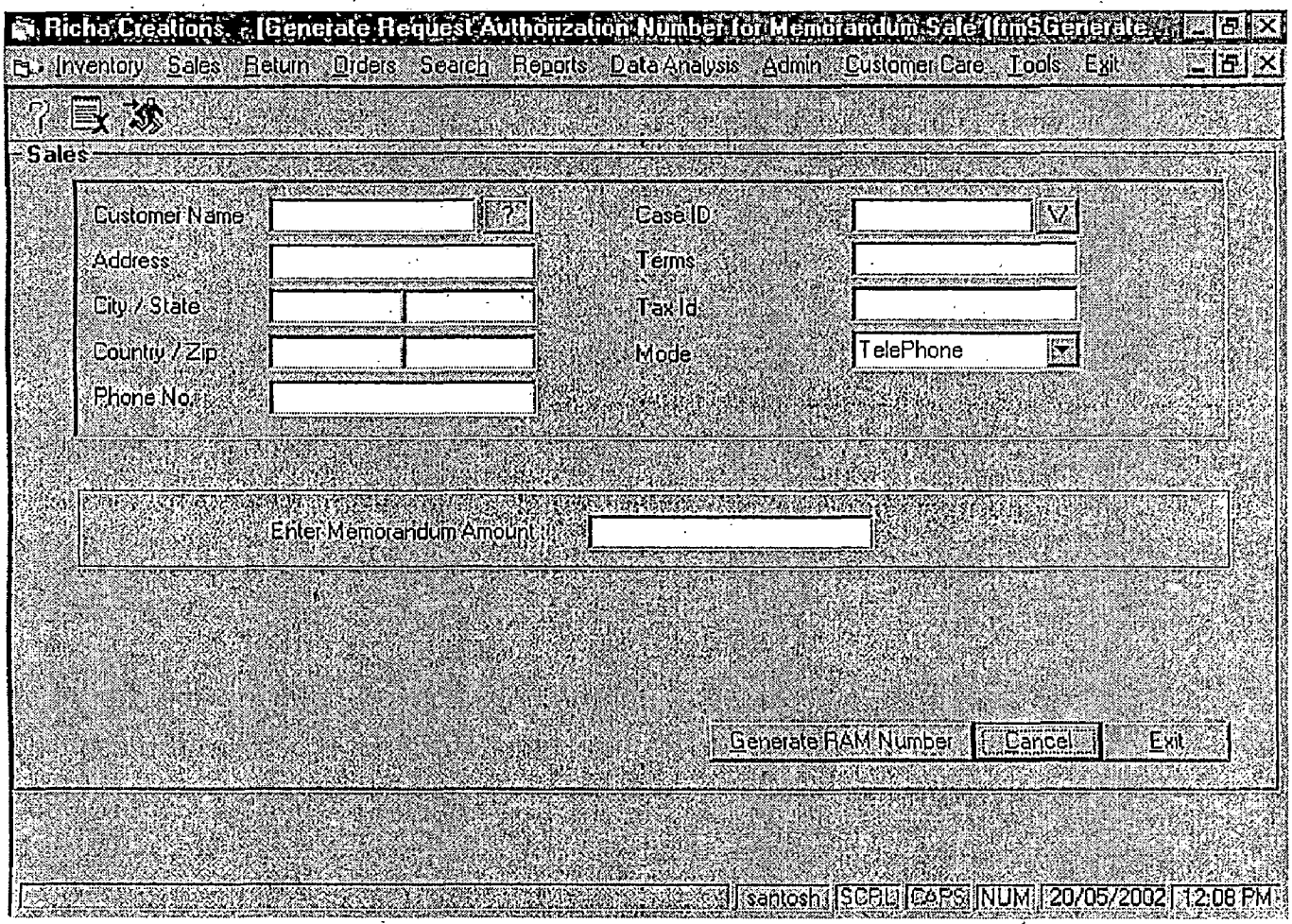


Figure [3.8] Form Interface to generate RAM Number.

This RAM Number (Request Authorization number for Memorandum sale) is generated by HOME. When REP does a memorandum type sale then he requires RAM Number. REP asks for RAM Number and then HOME gives him by Telephone or mail then REP does validation of that number. Without doing validation REP can't make memorandum type sale. REP cannot validation any randomly chosen number because by using complex algorithm this RAM Number is generated by HOME and reverse process is done at REP side. The form interface to generate the RAM Number is shown in Figure [3.8].

- Customer Name, address, city/state, country/zip, phone number, terms and tax id are populated by customer search and customer detail process. All these fields are mandatory.
- Proper Case Id and Mode are selected from drop down list.
- New Amount is entered in the Memorandum Amount.
- Generate RAM Number: To generate RAM Number this button is clicked.
- Exit: Exits out of this module.
- Cancel: Refreshes all the fields.

3.9 Generating RAT Number

This RAT Number (Request Authorization number for Term Id) is generated by HOME. When REP wants to change Term then he requires RAT Number. REP asks for RAT Number and then HOME gives him by Telephone or mail then REP does validation of that number. Without doing validation REP can't change Term. REP cannot validation any randomly chosen number because by using complex algorithm this RAT Number is generated by HOME and reverse process is done at REP side. The form interface to generate the RAT Number is shown in Figure [3.9].

- Customer Name, city/state, country/zip, phone number, terms, tax id are populated by customer search and customer detail process. All these fields are mandatory.
- Proper Case Id, Scheme Abbreviation and Mode are selected from drop down list.
- New Term Id is selected from drop down list.
- Generate RAT Number: To generate RAT Number this button is clicked.
- Exit: Exits out of this module.

- Cancel: Refreshes all the fields.

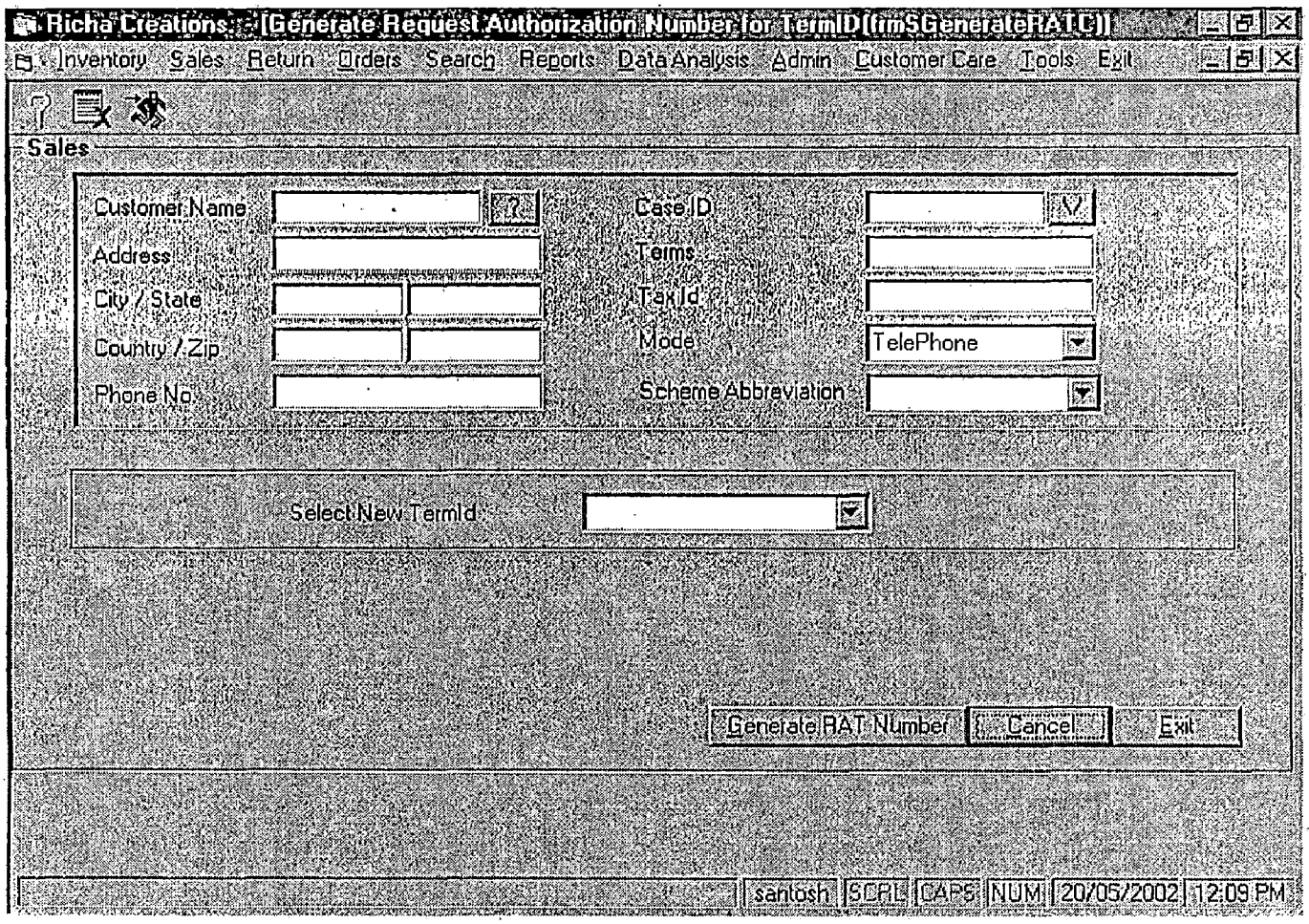


Figure [3.9] Form Interface to generate RAT Number.

3.10 Generating RAD Number

This RAD Number (Request Authorization number for Discount) is generated by HOME. When REP does a sale with scheme that requires RAD Number authorization to allow new discount then this number is needed. REP asks for RAD Number and then HOME gives him by Telephone or mail then REP does validation of that number. Without doing validation REP cannot give new discount to customer. REP cannot validation any randomly chosen number because by using complex algorithm this RAD Number is generated by HOME and reverse process is done at REP side. The form interface to generate the RAD Number is shown in Figure [3.10].

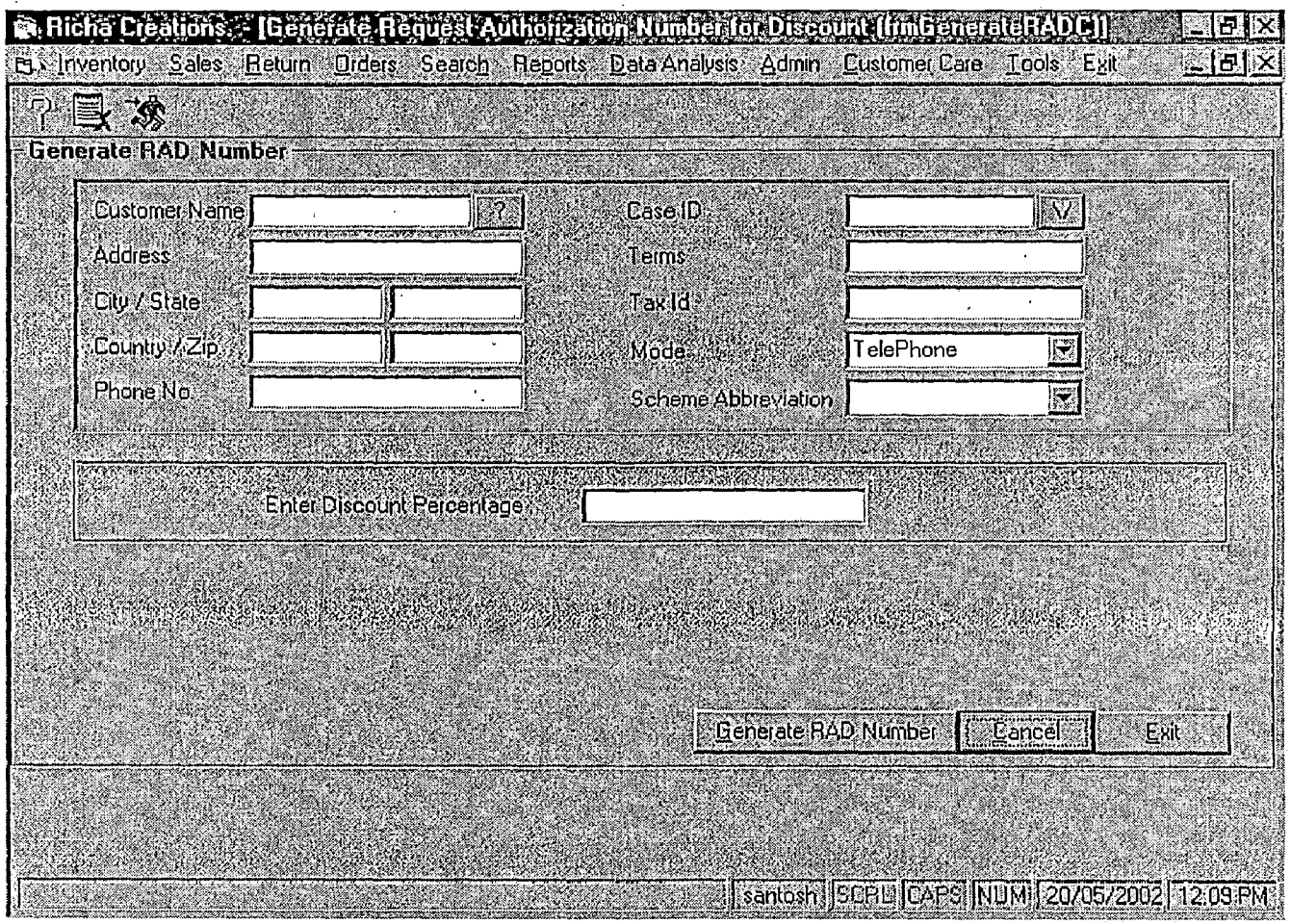


Figure [3.10] Form Interface to generate RAD Number.

- Customer Name, city/state, country/zip, phone number, term and tax id are populated by customer search and customer detail process. All these fields are mandatory.
- Proper Case Id and Mode are selected from drop down list.
- New Discount Percentage is entered in the Enter Discount Percentage field.
- Generate RAD Number: To generate RAD Number this button is clicked.
- Exit: Exits out of this module.
- Cancel: Clears all the fields.

4.1 What is Bar Code?

A bar code [4] consists of a series of parallel, adjacent bars and spaces. Predefined bar and space patterns or "symbologies" are used to encode small strings of character data into a printed symbol. Bar codes can be thought of as a printed type of the Morse code with narrow bars (and spaces) representing dots, and wide bars representing dashes. A bar code reader decodes a bar code by scanning a light source across the bar code and measuring the intensity of light reflected back by the white spaces. The pattern of reflected light is detected with a photodiode, which produces an electronic signal that exactly matches the printed bar code pattern. This signal is then decoded back to the original data by inexpensive electronic circuits. Due to the design of most bar code symbologies it does not make any difference if you scan a bar code from right to left or from left to right.

The basic structure of a bar code consists of a leading and trailing quiet zone, a start pattern, one or more data characters, optionally one or two check characters and a stop pattern.

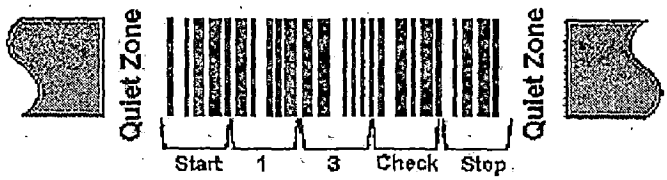


Figure [4.1] Basic Structure of a bar code.

There are a variety of different types of bar code encoding schemes or "symbologies", each of which were originally developed to fulfill a specific need in a specific industry. Several of these symbologies have matured into de-facto standards that are used universally today throughout most industries.

The different symbologies have different capabilities for encoding data. For example the UPC symbology used to identify retail products always contains 12 numeric digits whereas the general purpose Code 39 or Code 128 bar code symbologies can encode variable length alphanumeric data up to about 30 characters in length. These types of bar codes are called "linear symbologies" because they are made up of a series of lines of

different widths. Most commercially available bar code scanners are able to read all of the different linear bar code symbologies therefore we do not need different readers for different types of bar codes.

The primary purpose of a bar code is to identify something by labeling the item with a bar code containing a unique number or character string. Bar codes are typically used with a database application where the data encoded in the bar codes is used as an index to a record in the database that contains more detailed information about the item that is being scanned. For example, when a checkout clerk scans a bar code on a product in a grocery store, the bar code data is fed to a computer that looks up the information in a central database and returns more detailed information about the item that was scanned including possibly a description of the item and a price.

Bar codes also provide a quick and error free means for inputting the data into an application running on a computer. By using bar codes, the potential for errors from manual data input is eliminated. Another typical application for bar codes is therefore for inputting data without having to type. For example you could encode name or address data in a bar code on an ID badge and then scan the ID badges to input a persons name into a computer program instead of typing the information.

The different bar code symbologies support different types and amounts of data therefore we normally choose a particular symbology based on the type and amount of data that we want to encode in our bar codes. We are generally free to use any type of bar code that we like and encode whatever data we like for applications.

4.2 Bar Code Symbologies

In bar code different bar and space patterns are used to represent different characters. Sets of these patterns are grouped together to form a "symbology". There are many types of bar code symbologies each having their own special characteristics and features. Most symbologies were designed to meet the needs of a specific application or industry. For example the UPC symbology was designed for identifying retail and grocery items and PostNET was designed to encode Zip Codes for the US Postal Service.

The following is a detailed description of the most commonly used bar code symbologies.

4.2.1 CODE 39

Figure [4.2] Code 39 of Data CODE 39

The Normal CODE 39 is a variable length symbology that can encode the following 44 characters:

1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ-. *$/+%. Code 39 is the most popular symbology in the non-retail world and is used extensively in manufacturing, military, and health applications. Each Code 39 bar code is framed by a start/stop character represented by an asterisk (*). The Asterisk is reserved for this purpose and may not be used in the body of a message.

Code 39 optionally allows for a (modulo 43) check character in cases where data security is important. The health care industry has adopted the use of this check character for health care applications.

The FULL ASCII version of Code 39 is a modification of the NORMAL (standard) version that can encode the complete 128 ASCII character set (including asterisks). The Full ASCII version is implemented by using the four characters: $/+% . as shift characters to change the meanings of the rest of the characters in the Normal Code 39 character set. Because the Full ASCII version uses shift characters in combination with other standard characters to represent data not in the Normal Code 39 character set, each non-standard character requires twice the width of a standard character in a printed symbol.

Because all of the characters used to implement Full ASCII Code 39 are part of the Normal Code 39 character set, readers that do not support Full ASCII Code 39 will still read Full ASCII Code 39 symbols. The reader will output shifted characters as if they were normal Code 39 characters.

4.2.2 UPC-A, UPC-E and UPC Supplemental

UPC-A is a 12 digit, numeric symbology used in retail applications. UPC-A symbols consist of 11 data digits and one check digit. The first digit is a number system digit that normally represents the type of product being identified. The following 5 digits are a manufacturers code and the next 5 digits are used to identify a specific product.

The Uniform Code Council (UCC) assigns UPC numbers to specific products and manufacturers. Both UPC-A and UPC-E allow for a supplemental two or five digit number to be appended to the main bar code symbol. This supplemental message was designed for use on publications and periodicals. If you enter a supplemental message, it must consist of either two or five numeric digits. The supplemental is simply a small additional bar code that is added onto the right side of a standard UPC symbol.

UPC-E is a smaller seven-digit UPC symbology. It is often used for small retail items.

Differences between Type A and Type E

The main difference between a UPC-A symbol and a UPC-E symbol is the size. Below is a UPC-A bar code on the left and the same data encoded as a UPC-E symbol on the right.

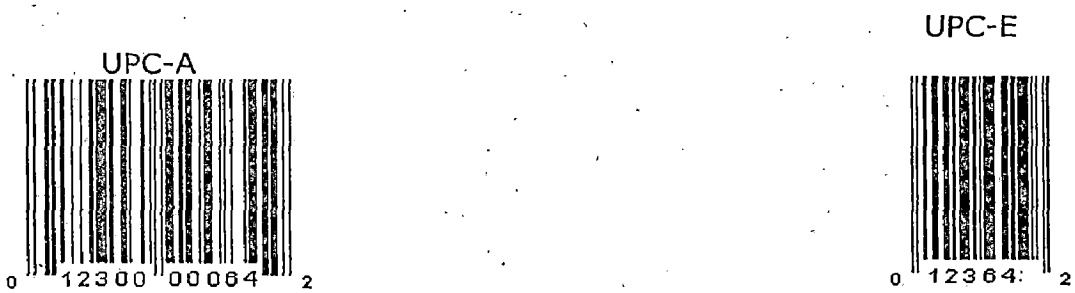


Figure [4.3] These two bar codes are equivalent.

4.2.3 EAN-8 EAN-13 with supplemental (ISBN Version)

EAN or European Article Numbering system (also called JAN in Japan) is a European version of UPC. It uses the same size requirements and a similar encoding scheme as for UPC codes.

EAN-8 encodes 8 numeric digits consisting of two country code digits, five data digits and one check digit.

EAN-13 is the Euro version of UPC-A. The difference between EAN-13 and UPC-A is that EAN-13 encodes a 13th digit into the parity pattern of the left six digits of a UPC-A symbol. This 13th digit, combined with the 12th digit, usually represents a country code.

Both EAN-8 and EAN-13 support a supplemental two or five digit number to be appended to the main bar code symbol. The supplemental is designed for use on publications and periodicals. Supplemental messages must consist of either two or five numeric digits and will appear as a small additional bar code on the right side of a standard EAN symbol.

EAN bar code numbers are assigned to specific products and manufacturers by an organization called ICOF located in Brussels, Belgium.

EAN-13 has been adopted as the standard in the publishing industry for encoding ISBN numbers on books. An **ISBN** or Book Land bar code is simply an EAN-13 symbol consisting of the first 9 digits of the ISBN number preceded by the digits 978. The supplemental in an ISBN bar code is the retail price of the book preceded by the digit 5. For example, if ISBN number is 1-56276-008-4 and the price of the book is $29.95 then we would enter 978156276008 as the bar code message and 52995 for the supplemental.

4.2.4 CODABAR

CodaBar is a variable length symbology that allows encoding of the following 20 characters: 0123456789-$:/.+ABCD. CodaBar is commonly used in libraries, blood banks, and the air parcel business. CodaBar uses the characters A B C and D only as start and stop characters. Thus, the first and last digits of a CodaBar message must be A B C or D and the body of the message should not contain these characters.

4.2.5 INTERLEAVED 2 OF 5

Figure [4.4] Example of I 2 of 5

Interleaved 2 of 5 is a high density variable length numeric symbology that encodes digit pairs in an interleaved manner. The odd position digits are encoded in the bars and the even position digits are encoded in the spaces. Because of this, I 2 of 5 bar codes must

consist of an even number of digits. Also, because partial scans of I 2 of 5 bar codes have a slight chance of being decoded as a valid (but shorter) bar code, readers are usually set to read a fixed (even) number of digits when reading I 2 of 5 symbols. The numbers of digits are usually pre-defined for a particular application and all readers used in the application are programmed to only accept I 2 of 5 bar codes of the chosen length. Shorter data can be left padded with zeros to fit the proper length.

4.2.6 DISCRETE 2 OF 5

Figure [4.5] Example of Discrete 2 of 5

Discrete 2 of 5 is a variable length numeric symbology very similar to Interleaved 2 of 5 except that instead of encoding data in both the bars and the spaces, data is only encoded in the bars. Because of this, discrete 2 of 5 is not as compact as Interleaved 2 of 5 and also, odd numbers of digits may be encoded. Use of Discrete 2 of 5 is not very common and few bar code readers support this symbology.

4.2.7 CODE 93

Figure [4.6] Example of CODE 93

CODE 93 is a variable length symbology that can encode the complete 128 ASCII character set. Code 93 was developed as an enhancement to the CODE 39 symbology by providing a slightly higher character density than CODE 39. CODE 93 also incorporates two check digits as an added measure of security. Although CODE 93 is considered more robust than CODE 39, it has never achieved the same popularity as Code 39.

4.2.8 CODE 128

Figure [4.7] Example CODE 128

This bar code consists of the full ASCII set of 128 characters and number of digits is variable in length. Checksum is performed through the modulus 103 calculations and added to the end of digit.

Code subset A is a bar code which consists of standard upper-case alphabetic characters, numeric, keyboard characters, control characters and special characters.

Code subset B is a bar code, which consists of standard upper and lower-case alphabetic characters, numeric, keyboard characters, control characters and special characters.

Code subset C is a bar code, which consists of special characters and 2 digits from 00 to 99. The numeric digit number gets coded twice.

4.2.9 Data Matrix

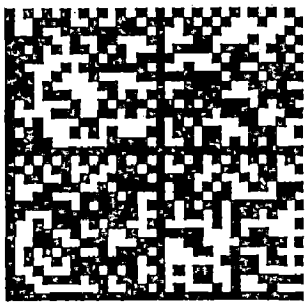


Figure [4.8] Example of Data Matrix.

Data Matrix is a high density 2 dimensional matrix style bar code symbology that can encode up to 3116 characters from the entire 256 byte ASCII character set. The symbol is built on a square grid arranged with a finder pattern around the perimeter of the bar code symbol.

There are two types of Data Matrix symbols each using a different error checking and correction scheme (ECC). The newest version of Data Matrix is called ECC 200 and is

recommended for all new Data Matrix applications. The ECC 200 version of Data Matrix uses a much more efficient algorithm for encoding data in a symbol as well as an advanced error checking and correction scheme.

4.2.10 POSTNET

Figure [4.9] Example of POSTNET.

POSTNET (POSTal Numeric Encoding Technique) is a 5, 9 or 11 digit numeric only bar code symbology used by the U.S. Postal Service to encode ZIP Code information for automatic mail sorting by zip code. The bar code may represent a five digit ZIP Code (32 bars), a nine digit ZIP + 4 code (52 bars) or an eleven digit Delivery Point code (62 bars).

POSTNET is unlike other bar codes because data is encoded in the height of the bars instead of in the widths of the bars and spaces. Most standard bar code readers cannot decode POSTNET. This symbology was chosen by the Postal Service mainly because it is extremely easy to print on almost any type of printer. POSTNET is a fixed dimension symbology meaning that the height, width and spacing of all bars must fit within exact tolerances.

4.3 Accuracy of Different Symbologies

It's commonly known that the best-trained data entry operator will make a keystroke entry error once every 300 keystrokes. Each of these keystroke errors represents an error in decision-making data. This leads to wasted time, misappropriated capital and ultimately, lost revenues.

In studies conducted by the University of Ohio, common bar code symbologies were tested to determine real life accuracy. The worst bar code for data accuracy in the test proved to be one of the most common - the UPC. The UPC had a worst-case error rate of 1 error in 394K characters. The best-tested symbology was the DataMatrix, with a worst-case error rate of 1 error in 10.5M characters. All the results from the University of Ohio study are listed below.

Taken from Lincoln's Log Vol 1-00

Symbology	Worst Case	Best Case
DataMatrix	1 error in 10.5M	1 error in 612.9M
Code 128	1 error in 2.8M	1 error in 37M
Code 39	1 error in 1.7M	1 error in 4.5M
UPC	1 error in 394K	1 error in 800K
Data Entry Operator	1 error in 300 Key Strokes	-

Table [4.1] Comparison of Accuracy of Different Symbologies.

4.4 How a Bar Code Reader Works

There are currently four different types of bar code readers available. Each uses a slightly different technology for reading and decoding a bar code. There are pen type readers (i.e. bar code wands), laser scanners, CCD readers and camera-based readers.

1. Pen Type Readers and Laser Scanners

Pen type readers consist of a light source and a photo diode that are placed next to each other in the tip of a pen or wand. To read a bar code, you drag the tip of the pen across all the bars in a steady even motion. The photo diode measures the intensity of the light reflected back from the light source and generates a waveform that is used to measure the widths of the bars and spaces in the bar code. Dark bars in the bar code absorb light and white spaces reflect light so that the voltage waveform generated by the photo diode is an exact duplicate of the bar and space pattern in the bar code. Scanner decodes waveform to generate bar code value.

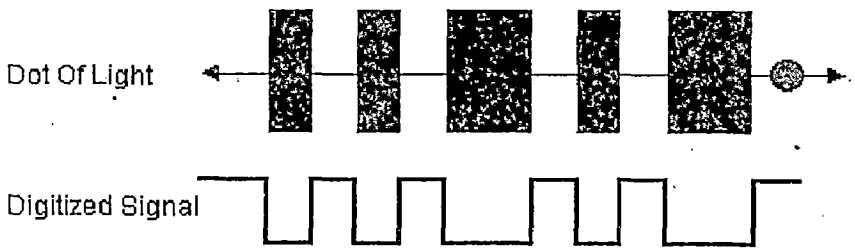


Figure [4.10] Reading a Bar Code with Pen Type Reader.

Laser scanners work the same way as pen type readers except that they use a laser beam as the light source and typically employ either a reciprocating mirror

or a rotating prism to scan the laser beam back and forth across the bar code. Just the same as with the pen type reader, a photo diode is used to measure the intensity of the light reflected back from the bar code. In both pen readers and laser scanners, the light emitted by the reader is tuned to a specific frequency and the photo diode is designed to detect only this same frequency light.

2. CCD Readers

CCD (Charge Coupled Device) readers use an array of hundreds of tiny light sensors lined up in a row in the head of the reader. Each sensor can be thought of as a single photo diode that measures the intensity of the light immediately in front of it. Each individual light sensor in the CCD reader is extremely small and because there are hundreds of sensors lined up in a row, a voltage pattern identical to the pattern in a bar code is generated in the reader by sequentially measuring the voltages across each sensor in the row. The important difference between a CCD reader and a pen or laser scanner is that the CCD reader is measuring emitted ambient light from the bar code whereas pen or laser scanners are measuring reflected light of a specific frequency originating from the scanner itself.

3. Camera Based Readers

The fourth and newest type of bar code reader currently available are camera based readers that use a small video camera to capture an image of a bar code. The reader then uses sophisticated digital image processing techniques to decode the bar code. Video cameras use the same CCD technology as in a CCD bar code reader except that instead of having a single row of sensors, a video camera has hundreds of rows of sensors arranged in a two dimensional array so that they can generate an image.

The factors that make a bar code readable are: an adequate print contrast between the light and dark bars and having all bar and space dimensions within the tolerances for the symbology. It is also helpful to have sharp bar edges, few or no spots or voids, a smooth surface and clear margins or "quiet zones" at either end of the printed symbol.

4.5 Interfacing a bar code reader to a PC

All application programs support bar code reading as long as we have the right equipment. Bar code readers are available with two types of output - either "keyboard wedge" output or RS232 output. The bar code readers with keyboard wedge output plug directly into the keyboard port on your PC and they also provide a pigtail connector so that you can plug in your keyboard at the same time. When you scan a bar code with the keyboard wedge bar code reader, the data goes into the computer just as if it were typed in on the keyboard. This makes it extremely easy to interface the bar code reader to any application that is written to accept keyboard data.

The keyboard wedge interface is extremely simple however it has a few drawbacks. If we swipe a bar code, the cursor has to be in the correct input field in the correct application otherwise we end up reading bar code data into whatever application has the focus. This can cause all sorts of potential problems. The keyboard output also is limited in that we cannot modify the data in any way before sending it into the program that is to receive the data. For example, if we needed to parse a bar code message into multiple pieces or remove some of a bar code message or add in a date or time stamps we would not be able to with a normal keyboard wedge reader.

The other possible output option is to get a bar code reader with an RS232 or "Serial" interface. With these types of bar code readers, we connect the reader to an available serial port on the back of PC. We would then need a program to take the data from the bar code reader and feed it to the application where we want the data to go. The disadvantage to this approach is that it is a little more complex however we gain much more control over how and where our data ends up when we read a bar code.

4.6 Printing Bar Code From Any Printer

A number of different print technologies have been used to print bar codes. These technologies roughly break down into two categories:

1. Impact printing techniques.
2. Non-impact printing techniques.

Impact printing includes dot matrix and Non-impact printing includes thermal direct, thermal transfer, laser printing and ink jet printers.

Printing 100% compliant barcodes from any printer to any label or document will ensure perfect readability. With good quality label printers with built-in barcode support this is usually possible as long as we carefully follow the manufacturer's quality control directions. But good thermal printers can cost several thousand dollars and are only designed for label printing. What about printing from general-purpose printers, such as laser printers? In this case the software used is critical to guarantee perfect barcodes.

4.6.1 Thermal Transfer Bar Code Label Printers

Along with electrostatic or laser printers, thermal transfer printers are becoming the choice of a majority of bar code users since the printers produce labels of outstanding quality. Ink is transferred to the paper by means of a heating element striking a ribbon; the ink dries on contact and is not subject to darkening or fading from sunlight or heat. Bar codes printed by thermal transfer printers are durable and won't distort. Printers are also quiet and suitable for environments where noise may be a factor.

These are commonly used printers for barcode labeling. It is most important with this type of printer to make sure that we follow the manufacturer's recommendations for setting up the printer and testing the bar codes. Direct thermal and thermal transfer printers require specific settings for best results depending on the combination of label and ribbon materials. The manufacturer will supply the directions for the correct adjustment for printer.

To maintain the quality of printed bar codes the manufacturer's directions for cleaning the print head and guide surfaces should be followed. It will also be necessary to replace the thermal print head eventually as these wear out. When this happens the bar codes will no longer be readable as one or more of the dot elements will not heat properly.

It is so important to verify printed symbols on a regular basis that some thermal printers can be equipped with on-line verifiers.

With thermal printers the quality of the label design software we use will not affect the quality of the printed bar codes. This is because the software is just sending a command

to print a bar code - a command that triggers the printer's internal software to actually generate and print the correct bar code.

4.6.2 Thermal Printers

Thermal printers (also known as thermal direct and heat stamp) are inexpensive and produce high quality labels. They are also very quiet in their operation and thus are ideal for offices, libraries or hospitals. They have two disadvantages, however. They print at a relatively low speed and use special, heat-sensitive paper, which is more expensive than paper stock used for other printers.

4.6.3 General Purpose Printers

General-purpose printers, especially laser printers, are excellent for producing bar code labels on sheets of pre-cut labels or on continuous feed labels. Obviously, for anyone needing extremely high quantities of bar codes on a daily basis thermal transfer printers with their high speed would be better, but for many users general-purpose printers, especially laser printers, are preferably.

Laser printers are also perfect for creating bar codes on documents, such as medical and legal records, coded "mail-merge" letters, etc.

With general-purpose printers, including laser, desktop, ink jet, and dot matrix, the software used is critical to ensuring readable symbols.

There is a wide variety of software packages for creating symbols using general purpose printers. Unfortunately, many of these packages are capable of producing symbols with totally unacceptable quality.

4.7 Bar Code Graphic Type

There are 3 main types of bar code software products.

1. Bar Code Fonts
2. Bit-map (Raster) Bar Code Generators
3. Vector Bar Code Generators

1. Bar Code Fonts

The Uniform Code Council does not recommend the use of fonts. The user cannot specify the module (bar) width exactly in printer dots. Fonts do not support Bar Width Reduction. Nor do they support many of the features specific to bar code symbols such as Bearer Bars, Quiet Zones, independent symbol height and module (bar) width specification and symbol rotation. In addition, most fonts do not automatically calculate and add check digits and other security features to bar codes.

2. Bit-map Bar Code Generators

A bit map image is device dependent. This means a bitmap of a one-inch square to be printed at 300 dpi would be 300 pixels across by 300 pixels down. If a bar code designed to print one-inch square on a 300 dpi printer were printed instead on a 600 dpi printer the resulting bar code would be ½ inch x ½ inch. This is because there would only be 300x300 pixels in the image instead of 600x600.

The other problem with bit-maps images is their size. Printing to a 200 dpi printer a one-inch square would contain 200 x 200=40,000 bits of data. This is bad enough but to a higher resolution printer, such as a common 600 dpi printer it would be 360,000 bits of data. These huge files not only use large amounts of computer memory they also print extremely slowly.

3. Vector Bar Code Generators

Vector graphics are perfect for defining bar code images. A vector image is a set of drawing commands that precisely defines the edges of each bar and specifies how to fill in the area created within the defined edges.

Vector images are completely device independent as they are a set of precise commands instead of a collection of dots. A 1" x 1" box will print 1" by 1" on a 100, 600 or 2540 dpi printer! When a vector image is sent to a general printer the printer's software converts it to a raster image so that it can print it. The Raster *Image Processor in the Printer converts the precise path commands in the vector* graphic into a series of pixels.

4.8 Advantages of Bar Coding

- **SPEED:** A bar code label of twelve characters can be wanded in approximately the time it takes a keyboard operator to make two keystrokes.

- **ACCURACY:** For every 1,000 characters typed by a keyboard operator, there are an average of ten keying errors. For an Optical Character Reader (OCR), there is one error in every 10,000 reads. With wands, a bar coding systems approach one error in every 3,000,000 characters and with laser technology; they approach one error in 70 million entries.

- **DATA INTEGRITY:** Probable number of substitution errors for every 3,400,000 characters entered:

Keyboard Entry	10,000
OCR Scanning	300
Bar Coding (Code 39)	1

- **EASE OF IMPLEMENTATION:** Operators of bar code scanners can learn to use the equipment effectively in less than 15 minutes. System costs are lower than other means of data entry because of the existence of interfacing hardware and software.

- COST EFFECTIVENESS: Greater versatility of scanning equipment such as reading wands, slot readers, scanners and portable readers. A bar code wand, for example, can be bought for as little as $140.
- Bar code labels can be used in harsh environments where there are extreme temperatures and dirt.

4.9 FACTORS AFFECTING BAR CODE PERFORMANCE

1. BAR HEIGHT

For hand-held readers, bar height should be at least one-quarter of an inch or 15 percent of the entire code's length, whichever is greater.

2. "X" DIMENSION

The "X" dimension is the width of the narrowest element of the bar code. Other elements of the code are multiples of the "X" dimension.

3. CHARACTER DENSITY

Density refers to the number of characters, which can be encoded in a given unit of length and is vitally important to the eventual application of a bar code.

4. CONTINUOUS/DISCRETE

Bar code symbologies are either continuous or discrete. Continuous symbologies use the inter character gap as a character, whereas discrete symbologies do not.

5. FIRST PASS READ RATE

The first read rate is the ratio of the number of successful reads to the number of attempted reads.

6. ADHESIVES

Adhesives vary with each application. Some require labels to permanently affixed to an item, such as a piece of capital equipment.

4.10 BAR CODING APPLICATIONS IN OPERATIONS

- **RECEIVING**

Whether the receiving dock gets shipments with bar codes already applied by suppliers or whether the receiver must print and apply labels for their own internal use, the applications of bar coding in the receiving area are the same. Labels identify the item, purchase order number, supplier, lot number, date of delivery, etc. This information is then used to create a receiving and purchasing

record, which will be part of the material's history throughout its cycle. This ensures lot trace ability.

- **INVENTORY**

The most valuable characteristic of bar codes in inventory control is keeping track of where the hundreds, even thousands, of parts, which arrive at a company, are stored or delivered.

- **FIXED ASSETS**

The application of bar coding to fixed assets was one of this technology's first uses. By attaching a label to capital equipment, office furniture, shelving and other permanent objects, companies are able to keep an account of what they own, to note the age of assets are and to conduct physical inventories.

- **SAVINGS AND BENEFITS**

There are many benefits from the application of bar coding. From the receiving to the shipping door, bar coding has a direct impact on labor productivity, production control, operation costs, customer services, space requirements and inventory control.

Chapter 5 Implementation of Bar Code Module

The thermal barcode printer that has been used for barcode label printing is C.Itoh S4. This printer is the product of [5] Itochu Electronics Co Ltd and manufactured by Citizen where C.Itoh is the brand name and S4 is the model number Figure [5.1].

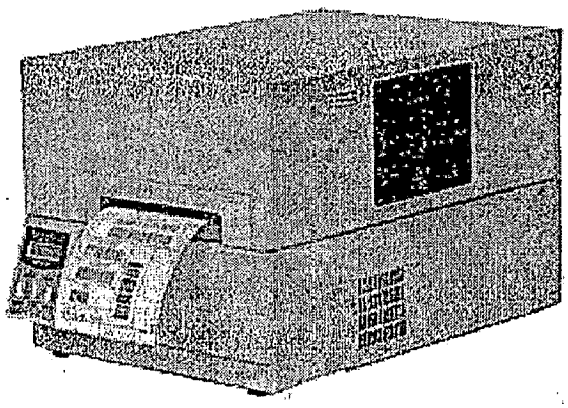


Figure [5.1] C.Itoh S4 Printer.

5.1 Command System and Command Interpreter

Generally, [7] when labels of graphics and bar codes are printed by a line printer, print data is converted into bit map data in the computer and transmitted to the printer for printing. In this process the host computer has to generate the bit map data and send it to the printer, this reduces throughput leading to slow printing and host computer operation.

This printer incorporate many functions such as a variety of fonts, bar code generators, and graphic commands along with high speed processing, so high-quality labels can be printed easily at high speeds when simple commands are transmitted from the host computer. The computer processing in generating labels is reduced enabling it to undertake more processing.

5.1.1 Command System

Commands for this printer consist of a string of ASCII characters and end with a "CR" (decimal: 13, hex: 0D). Generally, commands are classified into two types, that is, system level commands and label format commands.

System level commands are used in system level operations, including printer output, sensor selection and memory card maintenance. Label format commands are used in the definition of printing contents such as character data, bar code data, printing speed, and print density.

System level commands start with ASCII "SOH" ([01] H) or ASCII "STX" ([02] H). Commands, which start with "SOH", are requested for the real time execution. When received, they are executed immediately even during printing. Commands, which start with "STX", enter the buffer area and are executed in the order of data reception.

Label format commands follow the system level commands' "STX" + "L" and end with a "CR."

5.1.2 Command Interpreter

Two types of interpreters are used for this printer; system level and label format interpreters. When power is turned on, the system level interpreter is selected and the data received is processed in the system level interpreter and system level commands are executed.

Changing to the label format interpreter to start generating label data is executed with system level commands. When the system level interpreter receives the system level command "STX" + "L," it changes to the label format interpreter. The commands after this are regarded as label format commands and label format starts.

The label format interpreter does not need headers such as "SOH" and "STX." The data for printing data format is delimited by a "CR" and then transmitted. Changing to the system level interpreter from the label format interpreter is executed by the label format command "E" or "X." When label format ends with "E," defined data is printed and the system level interpreter is started. When label format ends with "X," the system level interpreter is started without printing.

5.2 Label Format Data

Two types of memory are available for this printer.

1. On-board RAM (standard): volatile storage memory
2. Memory card: (optional) non-volatile storage memory

This printer prints label format data by using memory space of the following size.

1. Reception buffer area: 32K bytes
2. Field register data area: 10K bytes
3. Global register area: 2K bytes
4. Bit map area: 676K bytes

1. Reception buffer area

The reception buffer area is a ring data buffer area (software FIFO) under software control. Basically, all commands and data transmitted from the host computer are buffered once into this area and then executed in the order of buffering to complete communications from the host computer in the shortest possible time. However, some system level commands (such as those starting with "SOH" for print halt), which require real time execution, are executed immediately after being received.

2. Field register area

A string of character data and bar code data is regarded as one field, which includes information such as type, print position, and size. The field register area is an area, which encloses the label format field. The label format interpreter analyzes the format data received, stores it once, and then generates a bit map. If there are any problems in the data received, the data, which is being analyzed, is discarded without being stored in the field register area. The field data is given a control number for every field (1, to 200) when the data is stored. This printer has a field register area of 10,000 characters and can print a maximum of 200 different types of field data per label.

3. Global register area

The global register area is an area, which stores field data that is repeatedly used. A part of the data (character string and bar code data) in the field register is stored in the global register area and used as requested. The data stored in the global register area is given a control number starting with A (A, B, ... P). The data set in the global register is stored while formatting for one label is performed (until the label format interpreter returns control to the system level interpreter) so it can be reused for data definition within the same label.

4. Bit map area

The bit map area is a buffering area for output data. The data in this area is generated by a rasterizer according to the data in the field data area and corresponds to individual dots, which are generated on the label during printing. The data of the bit map area is printed on the label with high quality and at high speed by means of the printer control program and exclusive thermal control circuit.

5.3 Label Printing Method

This printer has two label printing methods, one is that all label format data received is printed, and the other is that format data which has been received beforehand, is printed or partially modified and printed.

1. All data received method

i. ASCII code "STX" + "L" sets the printer to label format mode. The printer clears the field register area and control is transferred from the system level interpreter to the label format interpreter. At this time, use of label format commands is enabled.

ii. Printing data such as characters, bar codes and graphics is transmitted. Each data set has a special field structure, which includes information such as print position and size. The printing data received is checked by the label format interpreter, stored in the field register area, and generation of bit map data is then started. Powerful commands such as specification of the number of copies, characters strings and automatic incrementing or decrementing of bar code data

are included in the label format commands. In addition, the format data stored in the global register area can be read out and used.

iii. After completing label formatting, an ASCII "E" is transmitted. The printer prints the labels specified by the data in the field register area and control is then returned from the label format interpreter to the system level interpreter.

2. Using formatted data

i. In this mode, fixed format labels are printed. While label data formatting is completed, an ASCII "X" is transmitted instead of "E." The printer forms the field register area and completes formatting without printing and control is returned to the system level command processor. From this point on, the system level command processor allows the printer to print fixed format labels by using the format data in the field register area.

ii. If ASCII "STX" + "G" is transmitted to the system level command processor; the labels are printed according to the contents of the specified field register. In addition, change of data and number of copies is provided. (Only data can be changed. Format Information such as print position and size cannot be changed.) Printing with the "STX" + "G" command can be performed repeatedly.

5.4 Control Code Specification

- This printer can be connected to the computer via a serial interface or via parallel interface and prints characters and bar codes at the requested print position on the label. It also prints graphic data by using an optional IC card.

- The printer has a data area of 10,000 characters. This character data can be stored in up to 200 different fields. Each field stores attributes such as print position, rotation angle, font specification, and expansion factor (called attribute information). Machine control commands for print density or printing speed setting are used in addition to printing data control commands.

5.4.1 System Level Immediate Execution Commands

These commands are executed as soon as the printer receives them. They begin with "SOH," i.e. [01] H.

Command reset	[01] #
Pause	[01] B
Stop/cancel	[01] C
SOH command shutdown	[01] D
Printer status transmission request (1-byte packet)	[01] F

5.4.2 System Level Occasional Execution Commands

These commands are executed as soon as the printer receives them. They begin with "STX," i.e. [02]. Some commonly used commands are described below.

- **Setting paper length for continuous paper**

Code: [02] cnnnn
Unit: 0.01 inch
Setting: nnnn 4-digit data Initialization value: 0000
Inch system 0001 to 9999 (0.01 inch to 99.99 inches)
Metric system 0001 to 9999 (0.1 mm to 99.9 mm)

Function: Sets label length for continuous paper. Length of label format is specified with this command. Label is cut in the length with this setting when using auto-cutter. When using label paper, 0000 must be set.

Example: Example of input data below represents paper length of 2.5-inch setting.

Input data:

[02] n	Sets units to inch system
[02] c0250	Sets paper length to 2.5 inches for continuous paper
[02] L	Starts label format mode

D11	Sets pixel size
1F330600050005049012345678 9	Sets EAN13 bar code for data "490123456789"
E	Ends label format mode and prints

- **Changing units from inch to metric system**

Code: [02] m

Function: Changes units for all-distance-specified-command-parameters from inch to 0.1 mm. With reset, units are set to metric system.

Example: Input data below represents that data "ABC" is specified with units of metric.

Input data:

[02] m	Sets units to metric
[02] L	Starts label format mode
D22	Sets pixel size
190001001000050ABC	Sets character data "ABC" with smooth font 48pt
E	Ends label format mode and prints

- **Changing units from metric to inch system**

Code: [02] c

Function: Changes units for all-distance-specified-command-parameters from mm to 0.01 inch. With reset, units are set to inch system.

- **Clearing all memory module contents**

Code: [02] Q

Function: Clears all data in flash and on-board memory modules.

Caution: Clearing of all memory module contents takes about 30 seconds.

- **Clearing memory module contents**

Code: [02] qn

Setting: n Memory module specification parameter A, B, C
A Memory module A (on-board memory)
B Memory module B (flash memory card)
C Default module

Function: Clears all data in memory module.

- **Printing quality test pattern**

Code: [02] T

Function: After receiving this command, printer prints out the quality test pattern to check whether printer is in good condition and not involved in troubles such as head disconnection.

- **Printing printer status**

Code: [02] Z

Function: Performs test printing for indicating printer status.

- **Rewriting specified format register contents**

Code: [02] Unnaa..

Setting: nn 2-digit format register numbers 01 to 99 Printer sets format register numbers in sequence when label format is executed.

aa: Input character string data ending with CR code (Dh) instead of old data. Basically the number of characters must be the same as the old data. But if it is small, rewriting can be executed.

Function: Changes some part of the formerly printed label format or formatted label contents and prints it again.

Example: Input data below represents that data "0001" and "ABC" are set on the fields No. 01 and No. 02 respectively and label is printed and then with this command, the data contents of the fields No. 01 and No. 02 are changed to "9999" and "GHIJK" and printed.

Input data:

[02] n	Sets units to inch
[02] L	Starts label format mode
D11	Sets pixel size
1611000000000500001	Sets data "0001" on fields No. 01
161100000300050ABCDE	Sets data "ABCDE" on fields No. 02
E	Ends label format mode and prints
[02] U019999	Changes data on fields No. 01 to "9999"
[02] U02GHIJK	Changes data on fields No. 02 to "GHIJK"
[02]G	Prints edited format

- **Specifying printing contents setting start**

Code: [02] L

Function: With this command input, printer enters the label format mode and waits for input of printing contents definition and label format commands until it receives "E" or "X" code.

Example: Input data below represents that label format command input is started, data "ABC" is defined as printing contents, and label format

command input is completed, and label-printing command "E" is entered.

Input data:

[02] n	Sets units to inch
[02] L	Starts label format mode
D22	Sets pixel size
190001001000050ABC	Sets character data "ABC" with smooth font 48pt
E	Ends label format mode and prints

5.4.3 Label Format Commands

The following commands will be valid if the label format command interpreter is turned on with "STX" + "L," i.e. [02] L. Some commonly used label format commands are described below.

- **Setting offset in direction of column**

Code: Cnnnn

Units: 0.01 inch or 0.1 mm

Setting: nnnn 4-digit data Initialization value: 0000
Inch system 0000 to 9999 (0.00 inch to 99.99 inches)
Metric system 0000 to 9999 (0.0 mm to 99.9 mm)

Function: Sets offset value for printing start position in direction of column (paper left and right) to adjust the position of the total printing contents.

Example: Shifts label printing position 1.0 inch (C0100) rightward and prints.

Input data:

[02] n	Sets units to inch
[02] L	Starts label format mode
D11	Sets pixel size
C0100	Sets column offset to 1.0 inch
190000501150090ABC	Sets character data "ABC" with smooth font 14pt
1F330600050005049012345678 9	Sets EAN13 bar code for data "490123456789"
E	Ends label format mode and prints

- **Setting pixel size in horizontal and vertical direction**

Code: Dhv

Units: One dot 0.127 mm or 0.005 inch

Setting: h Dot size in horizontal direction 1, 2 Initialization value: 2
v Dot size in vertical direction 1, 2, 3 Initialization value: 2

Function: Specifies pixel size (dot formation units) in the range of 1 dot x 1 dot to 2 dots x 3 dots. If dot size in vertical direction is doubled or tripled, the maximum printing length (printing range) will be doubled or tripled.

Example: The following is the dot size in horizontal direction is set to 1 and dot size in vertical direction is set to 1.

Input data:

[02] n	Sets units to inch
[02] L	Starts label format mode
D11	Sets pixel size
1F330600050005049012345678 9	Sets EAN13 bar code for data "490123456789"
E	Ends label format mode and prints

- **Entering previous-defined field character string into global table**

Code: G

Function: Enters a previous-defined label format character string into global table, allocating one character in alphabetical order from A to P. Reading of character string stored in the global table is executed with command "S."

Example: Input data below represents that data of character string "ABC" is stored in the global table with font 9.

Input data:

[02] n	Sets units to inch
[02] L	Starts label format mode
D22	Sets pixel size
190000500800120ABC	Sets character data "ABC" with smooth font 14pt
G	Stores character string "ABC" in global table
150000502800120[02] SA	
E	Ends label format mode and prints

- **Setting print density (head heat factor)**

Code: Hnn

Setting: nn 00 to 30 Initialization value: 10

Function: Sets print density (heat energy is applied to print head).

- **Setting number of prints**

Code: Qnnnn

Setting: nnnn 0001 to 9999 Initialization value: 0001

Function: Sets number of sheets to be printed.

Example: Input data below represents that the same contents of labels are printed ten sheets.

Input data:

[02] n	Sets units to inch
[02] L	Starts label format mode
D11	Sets pixel size
Q0010	Setting 10 sheets for same label printing
1F330600050005049012345678 9	Sets EAN13 bar code for data "490123456789"
E	Ends label format mode and Prints

- **Setting offset in direction of row**

Code: Rnnnn

Units: 0.01 inch or 0.1 mm

Setting: nnnn 4-digit data Initialization value: 0000
Inch system 0000 to 9999 (0.00 inch to 99.99 inches)
Metric system 0000 to 9999 (0.0 mm to 99.9 mm)

Function: Sets offset value for printing start position in direction of row (paper top and bottom) to adjust the position of the total printing contents.

Example: Shifts label printing position 1.0 inch (R0100) upwards and prints.

Input data:

[02] n	Sets units to inch
[02] L	Starts label format mode
D11	Sets pixel size
R0100	Sets row offset to 1.0 inch
190000501150090ABC	Sets character data "ABC" with smooth font 14pt

1F33060005000504901234567 89	Sets EAN13 bar code for data "490123456789"
E	Ends label format mode and prints

- **Storing label format**

Code: snaa.aa

Setting: n A Memory module A (on-board memory)
B Memory module B (flash memory card)
C Default module
aa..a Maximum 16 characters of format name ending with CR code

Function: Stores label format into memory module and ends label format.

Example: Input data below represents that label format data is stored in the on-board memory by using format name "NAME."

Input data:

[02] n	Sets units to inch
[02] L	Starts label format mode
D11	Sets pixel size
190000700500050PRINTER	Sets data "PRINTER" with smooth font 24pt
sANAME	Stores label format name "NAME" in on-board RAM and ends label format mode

- **Calling label format**

Code: raa..a

Setting: aa..a Maximum 16 characters of format name ending with CR code
Function: Calls label format stored in the memory module. Storing of label format into memory module is executed with "s" command.

Example: Input data below represents that label format data is stored in the RAM by using "NAME" and label format is ended once, then put in again and file name "NAME" in the RAM is called and those data together with the current label format data are printed.

Input data:

[02] n	Sets units to inch
[02] L	Starts label format mode
D11	Sets pixel size
1900007005000 50PRINTER	Sets data "PRINTER" with smooth font 24pt
sANAME	Stores label format name "NAME" in on-board RAM and ends label format mode
[02] L	Starts label format mode
rNAME	Reads out label format "NAME" data
D11	Sets pixel size
1900007008000 50LABEL	Sets data "LABEL" with smooth font 24pt
E	Ends label format mode and prints

- **Character field definition**

Code: rotate, font, hexp, vexp, point, row, column, d1, d2,.....

Setting: rotate: Sets rotation direction for character data 1, 2, 3 or 4

1: 0 Degrees rotation
2: 90 Degrees rotation
3: 180 Degrees rotation
4: 270 Degrees rotation

Font: Sets type of character (see table [5.1])
Hexp: Sets expansion rate in horizontal direction 1 to 9, A to O (A to O corresponding to 10 to 24)
Vexp: Sets expansion rate in vertical direction 1 to 9, A to O (A to O corresponding to 10 to 24)

Point: Sets size of smooth font A06 to A48
(Corresponding to 6pt to 48pt). Setting of this item is valid only when font is set to 9.

Row: Row address 0000 to 9999 Unit: 0.01 inch

Column: Column address 0000 to 0410 Unit: 0.01 inch

d1, d2,..: Printing character data Character data ending with CR By setting [02] Sn (n is a calling character string parameter specifying A to P), character string data stored in the global register is picked out and printed.

Function: Prints characters for contents data entered with items such as rotation, vertical and horizontal expansion rate, type of font and printing position.

The following table [5.1] shows the system fonts available for the printer C.Itoh S4:

Font	Horizontal (in Dots)	Spacing (In Dots)	Vertical (In Dots)
0	5	1	7
1	7	2	13
2	10	2	18
3	14	2	27
4	18	3	36
5	18	3	52
6	32	4	64
7	15	5	32
8	15	5	28
9(Smooth Font)	From 6 points to 48 points. (6,8,10,12,14,18,24,30,36,48)		

Table [5.1] Fonts supported by C.Itoh S4 Printer.

- **Bar code field definition**

Code: rotate, font, thick, narrow, hight, row, column, d1, d2,.....

Setting: rotate: Sets rotation direction for bar code 1, 2, 3 or 4

1: 0 Degrees rotation
2: 90 Degrees rotation
3: 180 Degrees rotation
4: 270 Degrees rotation

Font: Sets type of bar code (see table [5.2])
Thick: Sets thick bar width in 1-dot units (0.005 inch) 1 to 9, A to O (A to O corresponding to 10 to 24)
narrow: Sets narrow bar width in 1-dot units (0.005 inch) 1 to 9, A to O (A to O corresponding to 10 to 24)
height: Sets height of bar code data by using 3-digit numeric 001 to 999 Unit: 0.01 inch
Row: Row address 0000 to 9999 Unit: 0.01 inch
Column: Column address 0000 to 0410 Unit: 0.01 inch
d1, d2,..: Bar code data

Function: Encodes contents data specified with items such as rotation, size of bar code data and printing position into bar code and prints.

Example: Input data below is prepared with the following setting and printed. Font EAN-13 bar ratio (thick: narrow) 3:3 height 0.6 inch row, column 0.5 inch, 0.5 inch d1, d2,.. 490123456789

Input data:

[02] n	Sets units to inch
[02] L	Starts label format mode
D11	Sets pixel size
1F3306000500050490123456789	Sets EAN13 bar code for data "490123456789"
E	Ends label format mode and prints

The following table shows the Bar Code fonts used for various symbologies for C.Itoh S4 printer:

Symbology	Bar Code Font
Code 3 of 9	A
UPC-A	B
UPC-E	C
Interleaved 2 of 5	D
Code 128	E
EAN-13 (JAN-13)	F
EAN-8 (JAN-8)	G
HIBC	H
CODABAR	I
Interleaved 2 of 5 W/BARS	J
PLESSEY	K
CASE CODE	L
UPC 2DIG ADD	M
UPC 5DIG ADD	N
Code 93	O
ZIP	p
UCC/EAN-128	Q
UCC/EAN-128 (for K-MART)	R
UCC/EAN-128 Random Weight	S
Telepen	T
UPS MaxiCode	u
FIM	v
PDF-417	z

Table [5.2] Bar Code Fonts supported by C.Itoh S4 Printer.

5.5 Bar Code Module

It is not easy to print things such as Inventory Id, Description of Inventory, Price (Retail Price, Whole Sale Price), Vendor and Style and Bar Code of Inventory Id on a label (e.g. of dimension 12 mm X 54 mm). To print all the things in desired format effectively on the label requires deep study of all printer commands (Some of that have been explained above) and other factors, which affect speed of printing.

Following are figures of some Tag.

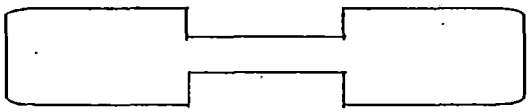


Figure [5.2] Tag 1

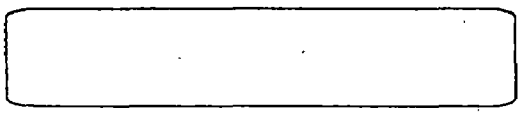


Figure [5.3] Tag 2

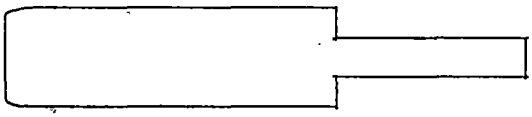


Figure [5.4] Tag 3

The Tag type that has been used for Bar Code Printing in ISMA application is Tag1 type. Its height is 12 mm and width is 54 mm. These labels are sticked on a continuous paper (like a sticker) that can be removed from the paper and again fixed with item.

5.5.1 Providing Interface To User

For printing perfect Bar Code with full flexibility that is Bar Code, Description, Price, Style and Vendor should start from desired position, in desired font, in desired orientation, in desired character expansion, all the input parameters should be entered by user when saving setting of tag.

But setting of desired location by new user will take many hit and trial and also there will be wastage of paper, because the input parameters are difficult to understand. Therefore Tag has been divided in five parts according to requirements of Richa Creations (Shown in Figure [5.5]).

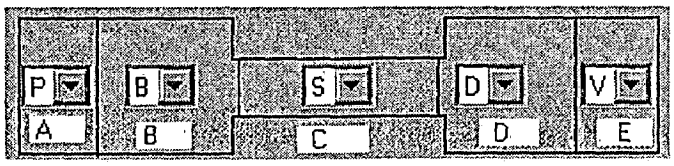


Figure [5.5] Partition of Tag in Five Parts

In Part A user can choose to print any one from Price, Vendor and Style. In Part B user can choose to print any one from Bar Code and Description. In Part C user can choose to print any one from Price, Vendor and Style. In Part D user can choose to print any one from Bar Code and Description. In part E user can choose to print any one from Price, Vendor and Style. All the selected fields will be different. User cannot select same value in more than one Part.

In the figure [5.5] above, the meaning of characters in the combo box is as follows:

- P – Price
- B – Bar Code
- S – Style
- D – Description
- V – Vendor

In Part A and Part E what will be chosen its orientation will be 90 degree or 270 degree. When User chooses from these combo boxes, he/she is asked for orientation that is 90 degree or 270 degree.

Suitable default settings are for every possible combination according to the requirements of Richa Creations. But also there is advanced setting for advance users where user can change every setting he/she wants that will be discussed later in this chapter.

Tag format is again divided in four formats as shown in Figure [5.6].

- Tag format 1
- Tag format 2
- Tag format 3
- Tag format 4

In Tag format1 everything (Bar Code, Description, Style, Price and Vendor) will be in different parts. In Tag format2 Price will be below of Inventory Bar Code either in Part B or in Part D and not in any other Part. In Tag format3 Style will be below of Inventory Bar Code either in Part B or in Part D and not in any other Part. In Tag format4 Style will be below of Inventory Bar Code either in Part B or in Part D and not in any other Part.

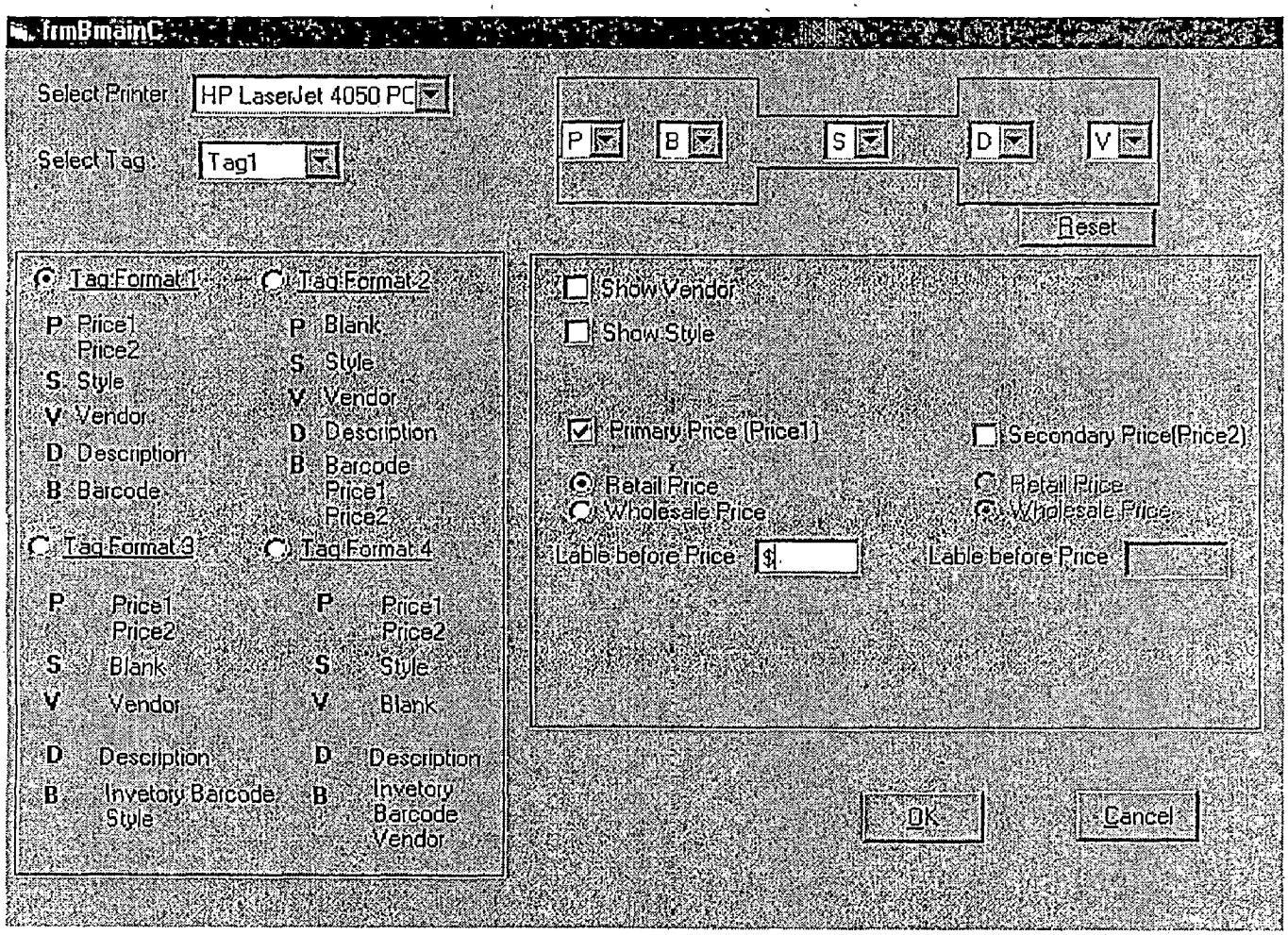


Figure [5.6] User interface for selecting Tag Format

There is two prices for each item, retail price and wholesale price. For printing flexibility price has been divided in two categories Primary Price (Price1) and Secondary Price (Price2). If user selects retail price in Price1/Price2 then automatically wholesale price will be selected in Price2/Price1.

User can set if he/she wants to print Vendor and Style or not by checking and un-checking Show Vendor check box and Show Style check box.

5.5.2 Providing Advance Setting

In advance setting user can set any input parameters required for Bar Code field and Character field definitions. There are input fields for every parameter for all Vendor, Style, Description, Price and Bar Code.

The User Interface for advance setting is shown in Figure [5.7].

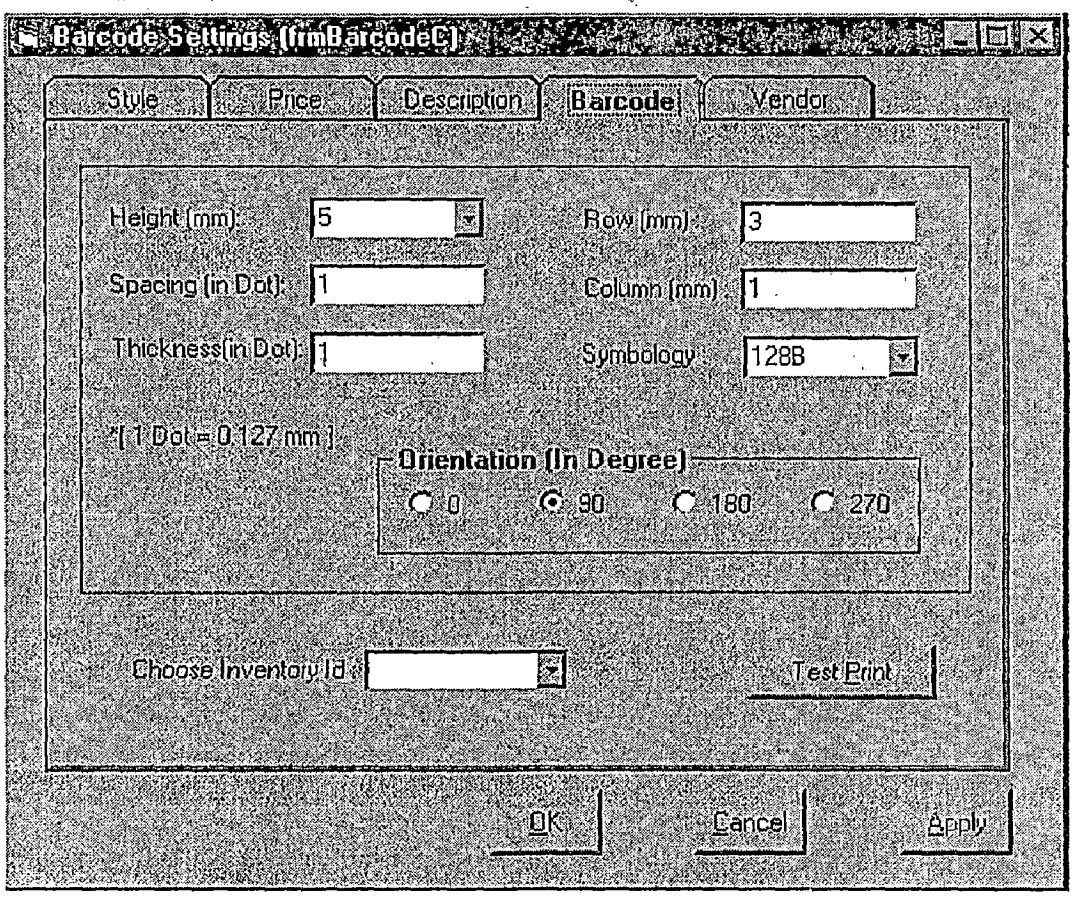


Figure [5.7] User Interface for Advance Setting.

5.5.3 How to Save Settings

All the settings made by user are saved in .INI file (bar.ini). The structure of bar.ini file is shown below.

```
[TAGPARAMETERS]
TAG_LENGTH=54
TAG_WIDTH=13
TAG_TYPE=1
SHOW_VENDOR=1
SHOW_STYLE=1
PRICE1=1
PRICE2=0
TAG_FORMAT=1

[TAG_FORMAT]
PARTA=P
PARTB=B
PARTC=S
PARTD=D
PARTE=V
PARTC_LENGTH=16

[PRINTER]
PRINTER_NAME=C.ITOH S4

[PRICE]
PRICE1_LABEL=$
PRICE1_TYPE=r
PRICE2_LABEL=
PRICE2_TYPE=
PRICE1_ORIGIN_X=10
PRICE1_ORIGIN_Y=0.5
PRICE2_ORIGIN_X=
PRICE2_ORIGIN_Y=
PRICE_ORIENTATION=90
```

[STYLE]
STYLE_ORIGIN_X=5.5
STYLE_ORIGIN_Y=19.5
STYLE_ORIENTATION=0

[VENDOR]

VENDOR_ORIGIN_X=10
VENDOR_ORIGIN_Y=51
VENDOR_ORIENTATION=90

[DESCRIPTION]

DESCRIPTION_ORIGIN_X=11
DESCRIPTION_ORIGIN_Y=36
DESCRIPTION_ORIENTATION=0
DESCRIPTION_LENGTH=15
DESCRIPTION_HEIGHT=12
DESCRIPTION_LINE_HEIGHT=1
DESCRIPTION_LINE_SPACE=0.3
DESCRIPTION_CHAR_WIDTH=1.15

[BARCODE]

BARCODE_ORIGIN_X=2.5
BARCODE_ORIGIN_Y=2
BARCODE_HEIGHT=4
BARCODE_NARROW=0.127
BARCODE_THICKNESS=0.127
BARCODE_SYMBOLOGY=128
BARCODE_ORIENTATION=0

[CHARACTER_FIELD_PARAMETER]

FONT_TYPE=Font 1
FONT_POINT=0
FONT_HEXP=0.127
FONT_VEXP=0.127
CHAR_SPACING=0.127

The following API's are used to set and retrieve the data from bar.ini file.

- GetPrivateProfileString()
- WritePrivateProfileString()

The declaration of these API's functions in Visual Basic is as follows:

Private Declare Function GetPrivateProfileString Lib "kernel32" Alias "GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Long, ByVal lpFileName As String) As Long

Private Declare Function WritePrivateProfileString Lib "kernel32" Alias "WritePrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpString As Any, ByVal lpFileName As String) As Long

5.5.4 How to Send Printer Commands from Application

If printer commands are sent to printer as other documents are sent for printing then commands are not executed by printer instead printer prints them as a simple text. To get executed the system commands and label format commands by printer, these commands should be directly written to printer port. The following windows APIs have been used to send the commands directly to printer.

- StartPage ()
- OpenPrinter ()
- ClosePrinter ()
- StartDocPrinter ()
- StartPagePrinter ()
- EndDocPrinter ()
- EndPagePrinter ()
- WritePrinter ()

FinalCommandString = Chr(2) & "n" & Chr(13) & Chr(2) & "c0200" & Chr(13) & Chr(2) & "L" & Chr(13) & "D11" & Chr(13) & "C0050" & Chr(13) & "1B33060005000504901 23" & Chr(13) & "E"

Above FinalCommandString is a typical command string generated in Visual Basic which changes unit to inch, sets continuous paper length to 2 inches, sets pixel size 1X1 dots, sets column offset 0.5 inch, sets 128B Bar Code for data 490123 and prints.

What is an API?

API stands for *application programming interface*. An API is simply a set of functions that can be used to work with a component, application, or operating system. Typically an API consists of one or more DLLs that provide some specific functionality.

DLLs are files that contain functions that can be called from any application running in Windows. At run time, a function in a DLL is *dynamically linked* into an application that calls it. No matter how many applications call a function in a DLL, that function exists in only a single file on the disk, and the DLL is created only once in memory.

The Windows API, which includes the DLLs that make up the Windows operating system. Every Windows application interacts with the Windows API directly or indirectly. The Windows API ensures that all applications running under Windows will behave in a consistent manner.

As the Windows operating system has evolved; several versions of the Windows API have been published. Windows 3.1 uses the Win16 API. The Windows NT, Windows 95, and Windows 98 platforms use the Win32 API.

There are other published APIs available in addition to the Windows API. For example, the Mail Application Programming Interface (MAPI) is a set of DLLs that can be used to write e-mail applications.

APIs are traditionally written for C and C++ programmers, who are building Windows applications, but other programming languages, including VBA, can call the functions in a DLL. Because most DLLs are written and documented primarily for C/C++ programmers, calling a DLL function may differ somewhat from calling a VBA function.

Calling the Windows API and other DLL functions can be hazardous to the health of application. When calling a DLL function directly from code, we are bypassing some of the safety mechanisms that VBA normally provides. Any mistake in defining or calling a DLL function (as all programmers eventually do), cause an application error.

API Resources

In order to call functions in the Windows API, or any API for that matter, there is need of documentation describing the available functions, how to declare them in VBA, and how to call them. The Win32API.txt file contains VBA **Declare** statements for most of the functions in the Windows API. The API Viewer application that's included with Microsoft Visual Basic can be used to locate and copy the **Declare** statement that is needed.

5.6 Design Constraints

- The application best runs at 640X480 resolutions.
- Each screen has drop down menu feature, which are again supported with drop down features.
- On the system, there must be a provision of taking backup on the network.
- The administrator id and password based security will ensure protection from unauthorized access/tampering of the data for the Administrator section of the application.
- A periodic backup will be taken manually to ensure restoration of the system with last backup in case of data lost due to system crashes.
- The application will be loaded on IBM compatible Personal Computer, Pentium or higher series of computers under Windows 98 operating system.
- This application has been designed with three-tier architecture. Both the database server and the web server may be installed on the same or different computer systems. All the interacting computers should be connected through TCP-IP protocol.

Chapter – 6 Conclusion and Suggestions for Future Work

6.1 Conclusion

The sales module and bar code printing module have been developed and integrated successfully with ISMA (Inventory Sales-Management and Analysis system). All the sub-modules of sales module works properly. Bill of sale sub module successfully generates bill of sale. Generate invoice, Sales schemes and generate authorization number sub modules works properly.

Bar code printing module generates perfect bar code and bar code reader correctly reads it. Bar code module provides user interface for bar code and character field setting where user can set height, bar code spacing, symbology, bar code thickness, bar code orientation, row and column position for bar code fields and font orientation, font, horizontal expansion, vertical expansion and row and column position for character fields.

6.2 Suggestions for Future Work

1. The Application best runs at 640X480 resolutions. It can be made resolution free.

2. The application developed for generating Auth number, RAM number, RAT number and RAD number can be made more complex.

3. Different algorithm can be developed for different authorization number.

4. Bar code printing module only works for C.Itoh thermal printers.

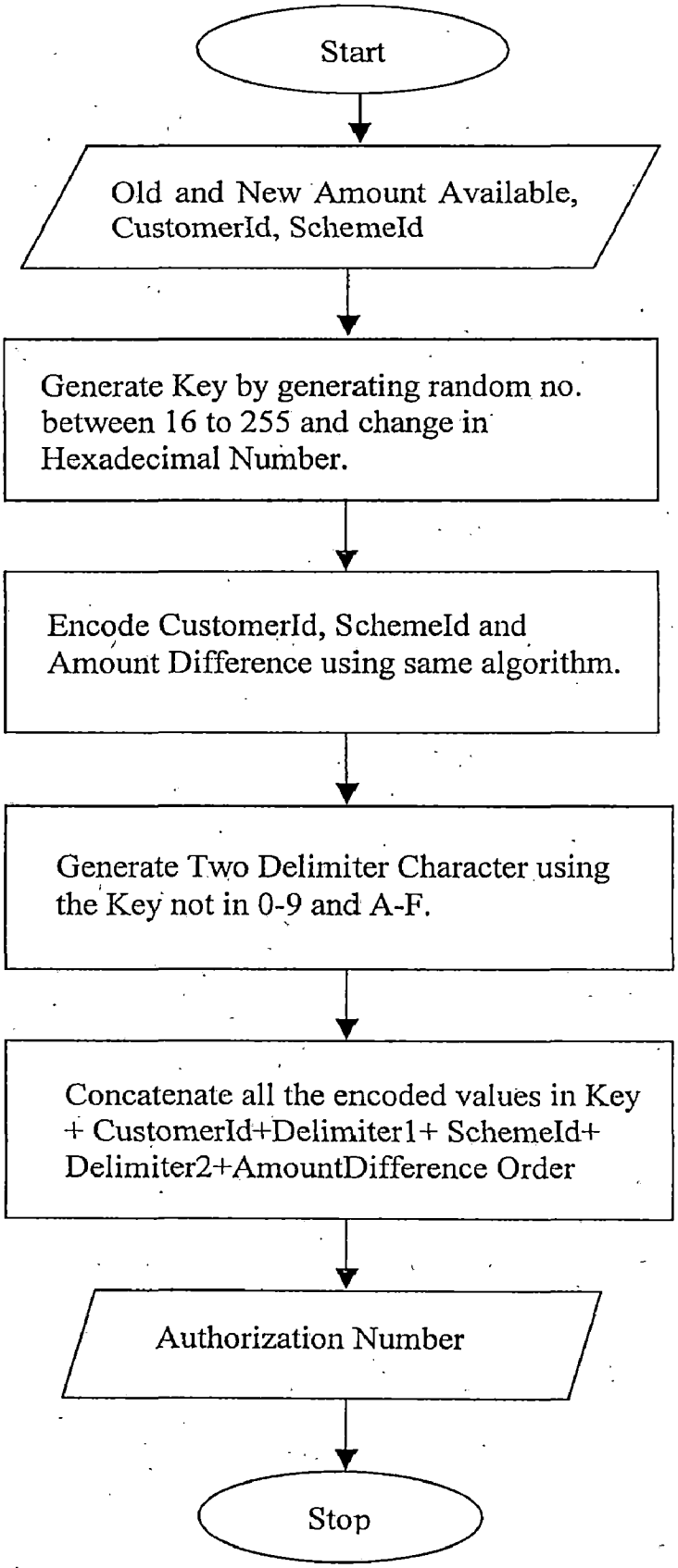


Figure [A.1] Flow Chart of Generating Authorization Number

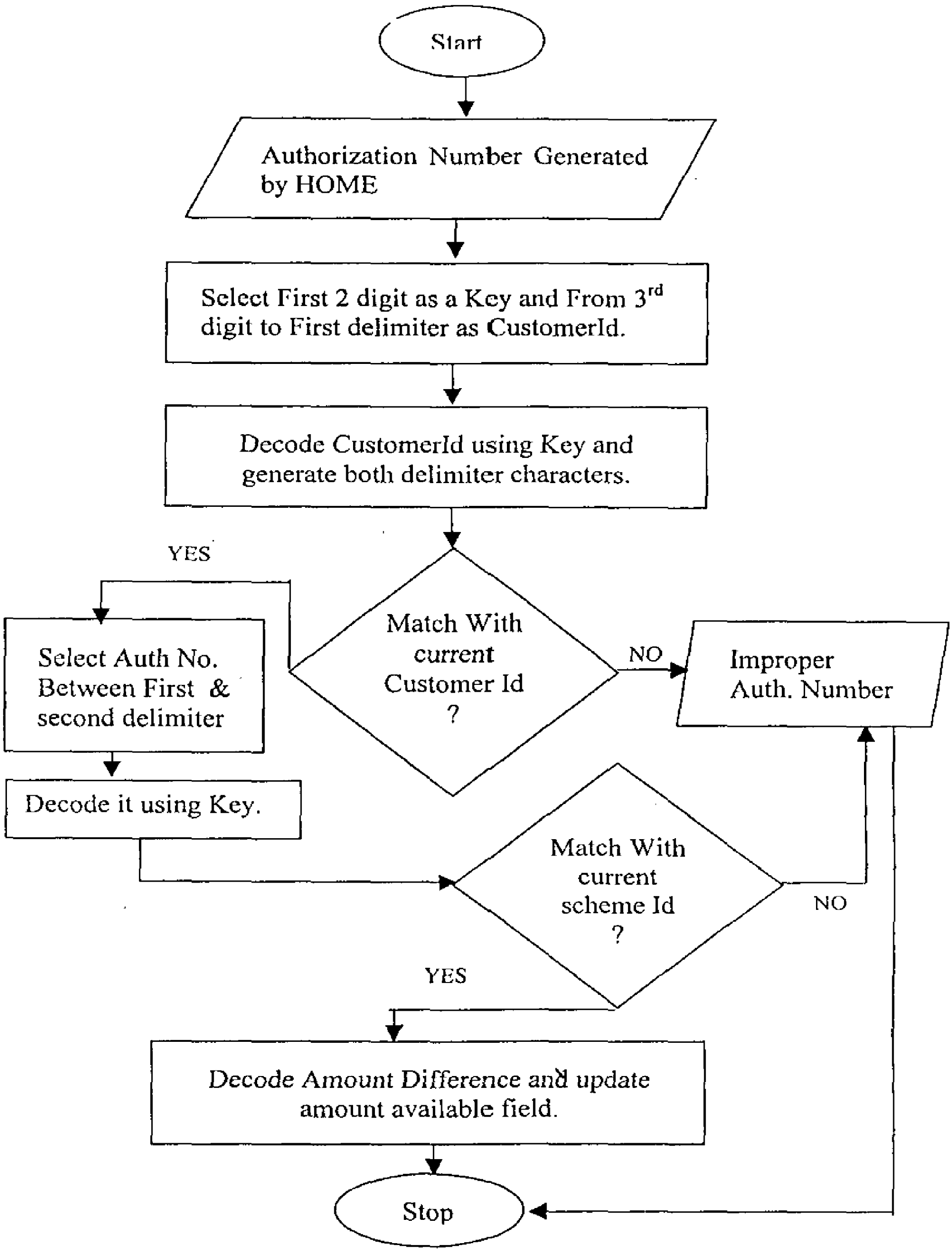


Figure [A.2] Flow Chart of Validating Authorization Number

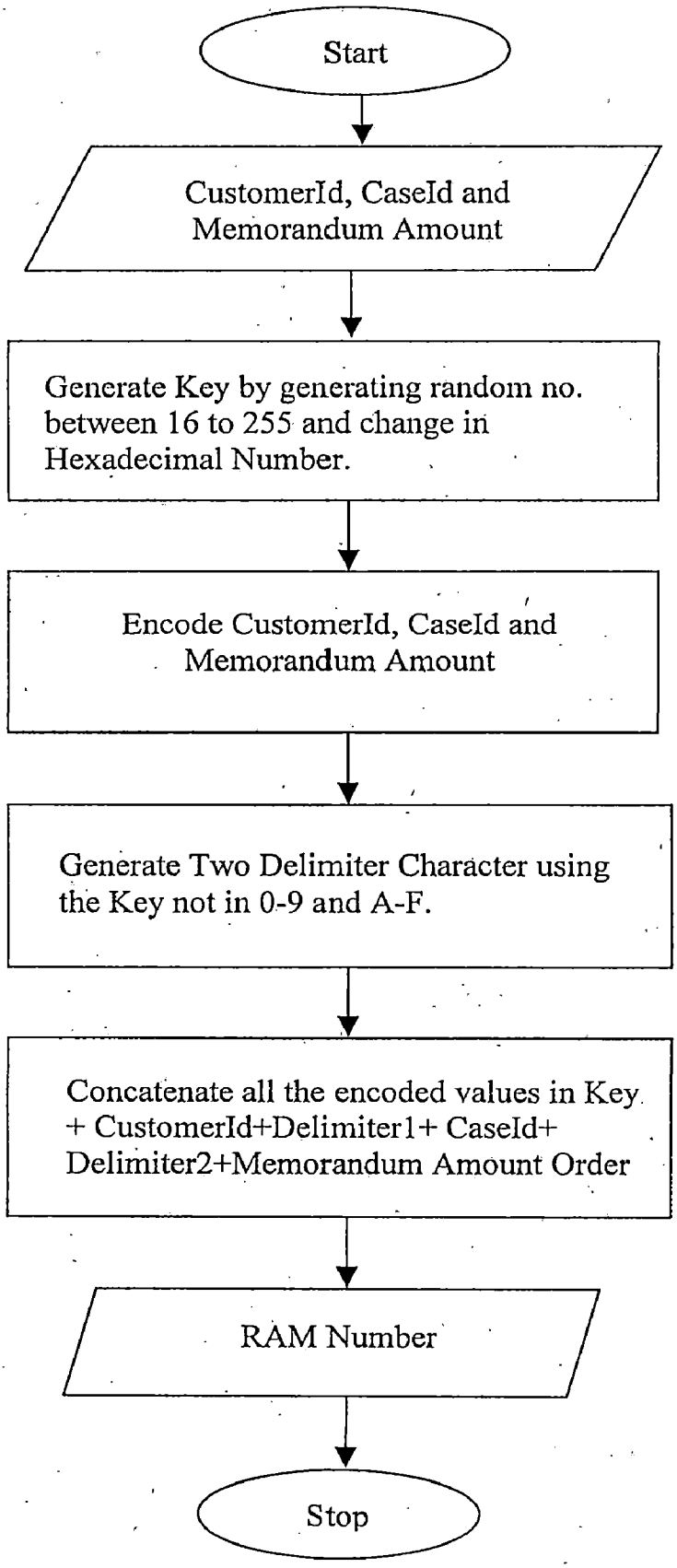


Figure [A.3] Flow Chart of Generating RAM Number

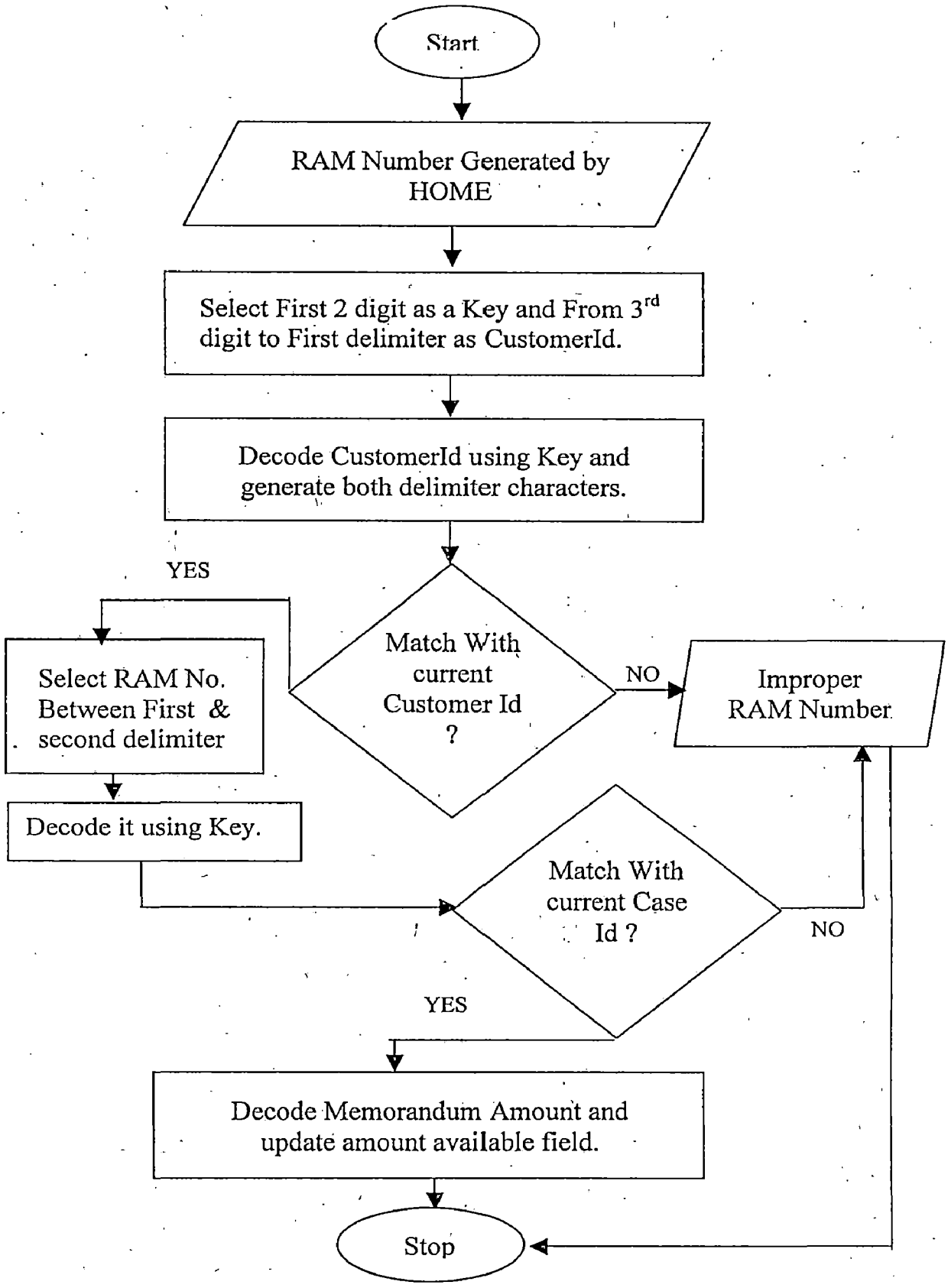


Figure [A.4] Flow Chart of Validating RAM Number

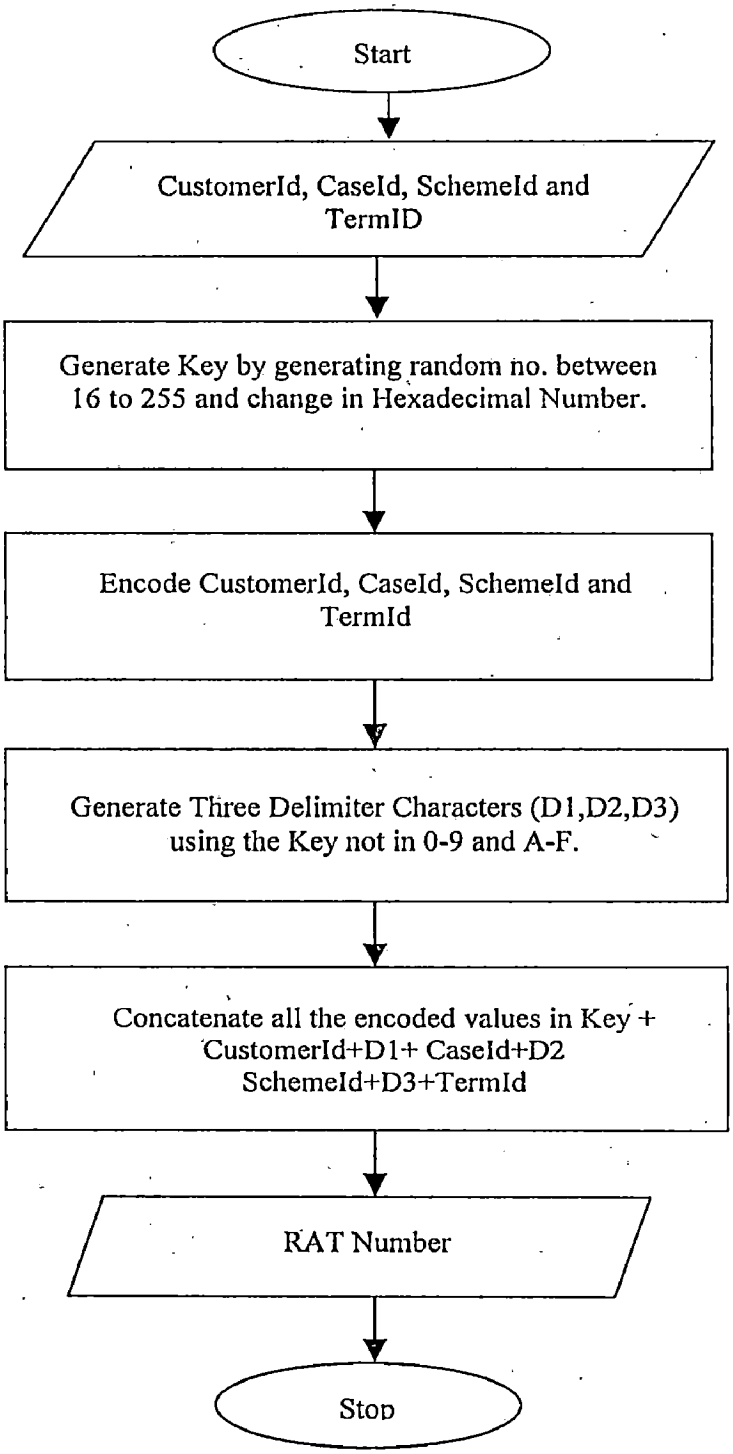


Figure [A.5] Flow Chart of Generating RAT Number

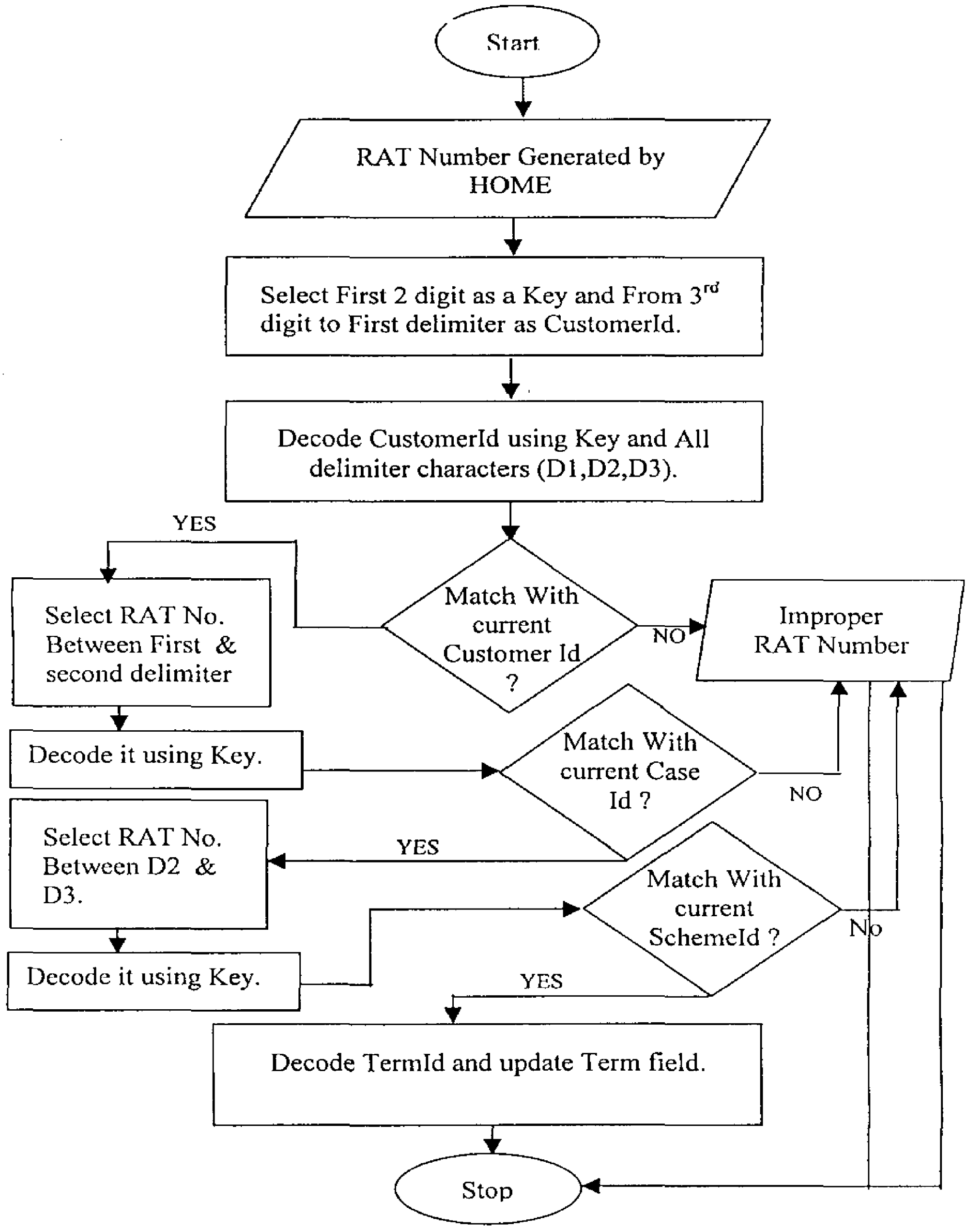


Figure [A.6] Flow Chart of Validating RAT Number

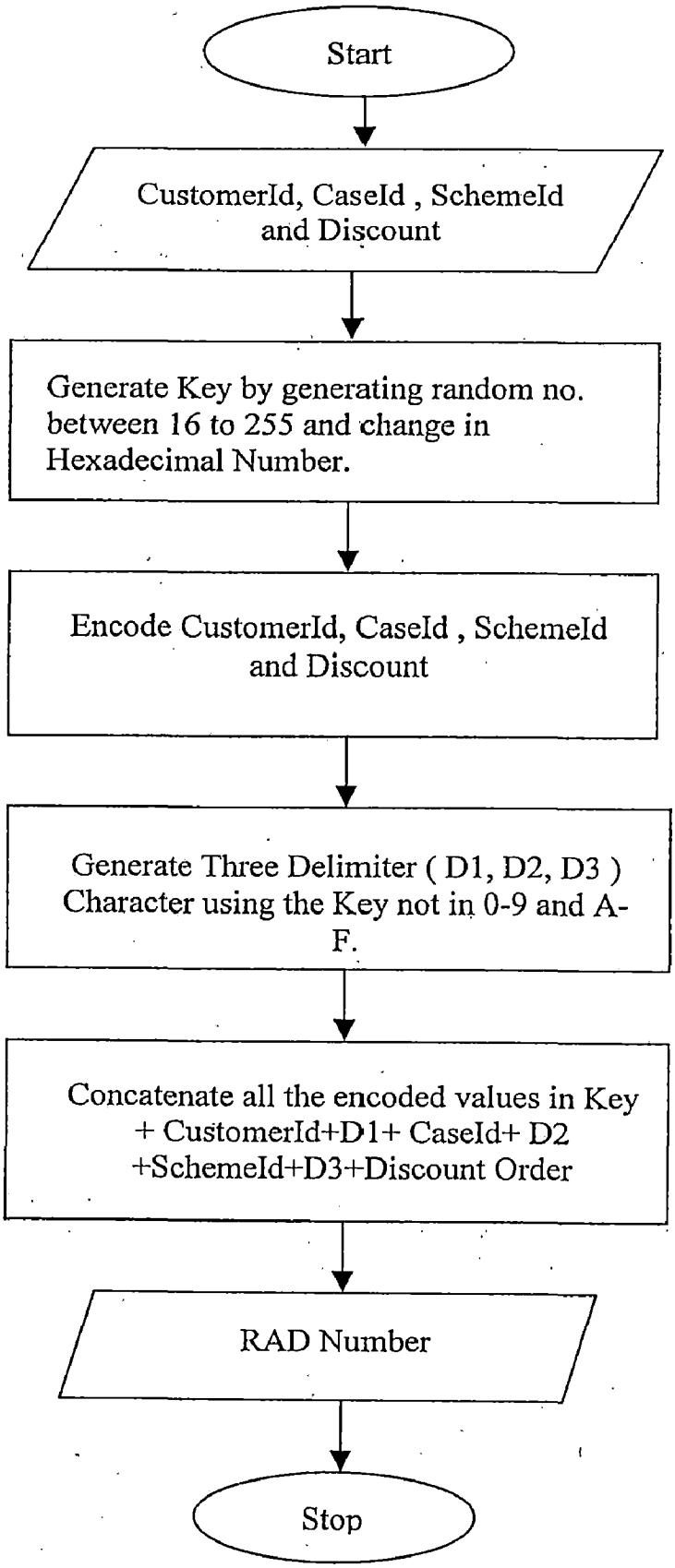


Figure [A.7] Flow Chart of Generating RAD Number

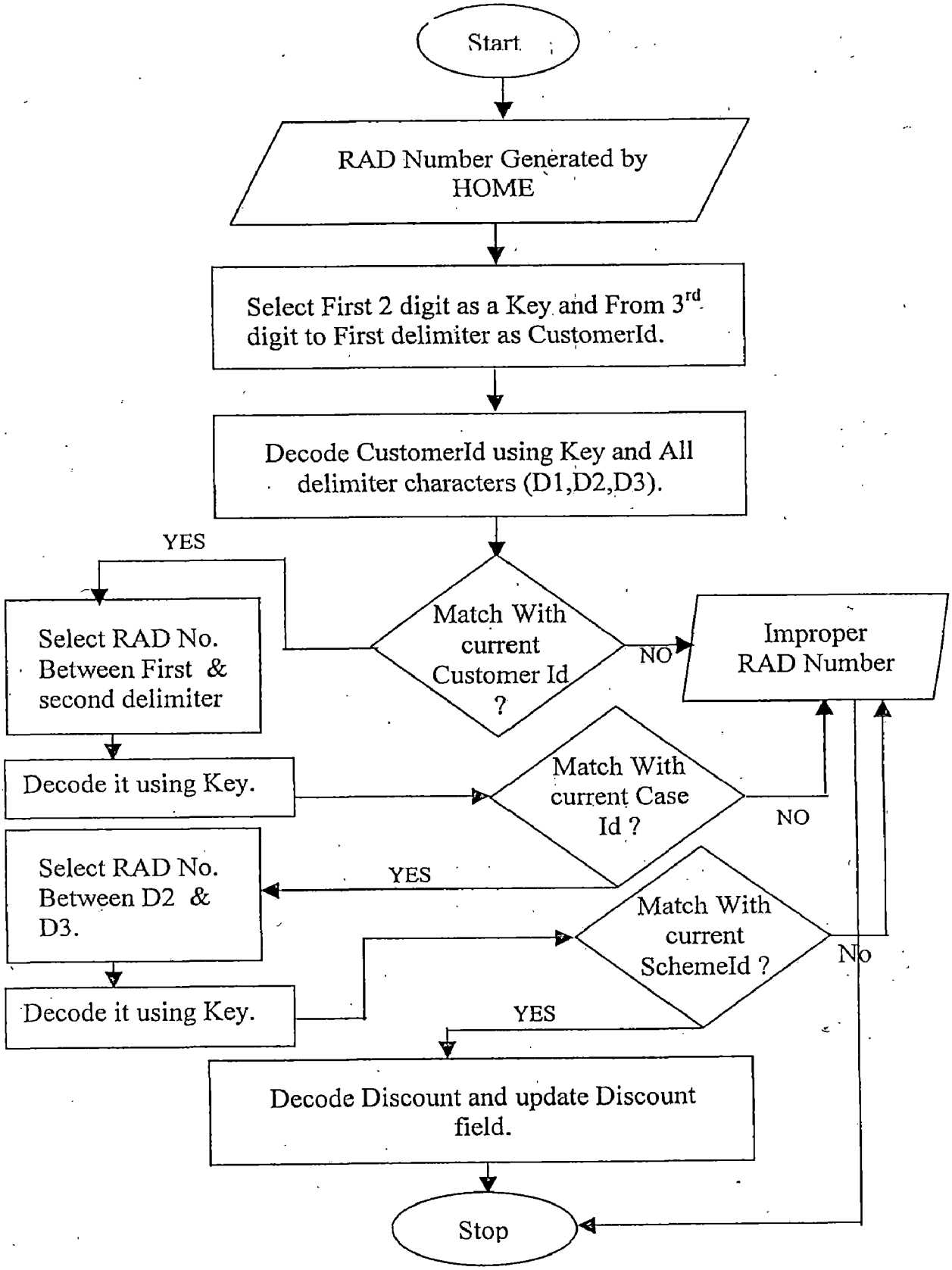


Figure [A.8] Flow Chart of Validating RAD Number

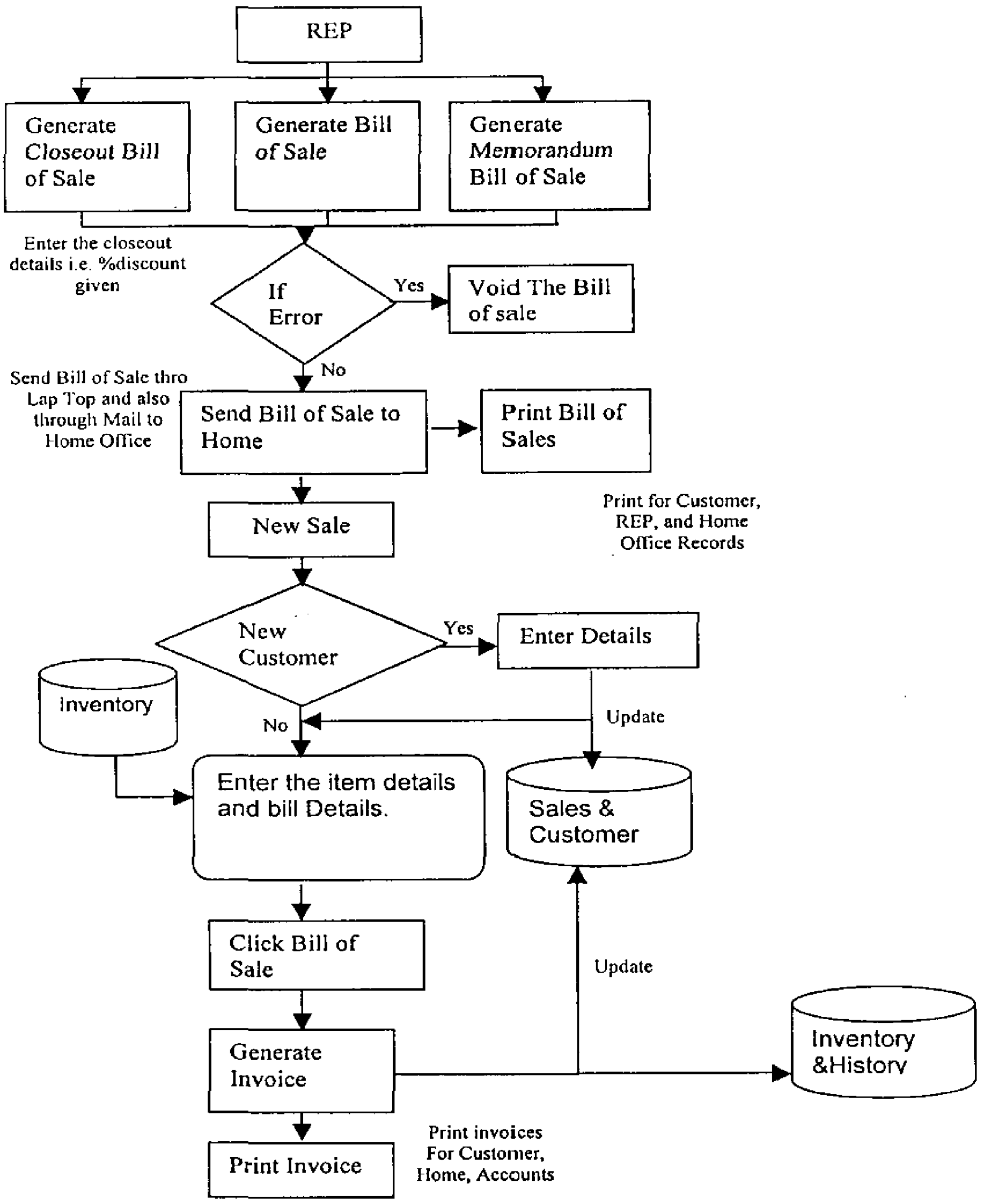


Figure [A.9] Flow Chart for the sub modules: Bill of Sale and Generate Invoice.

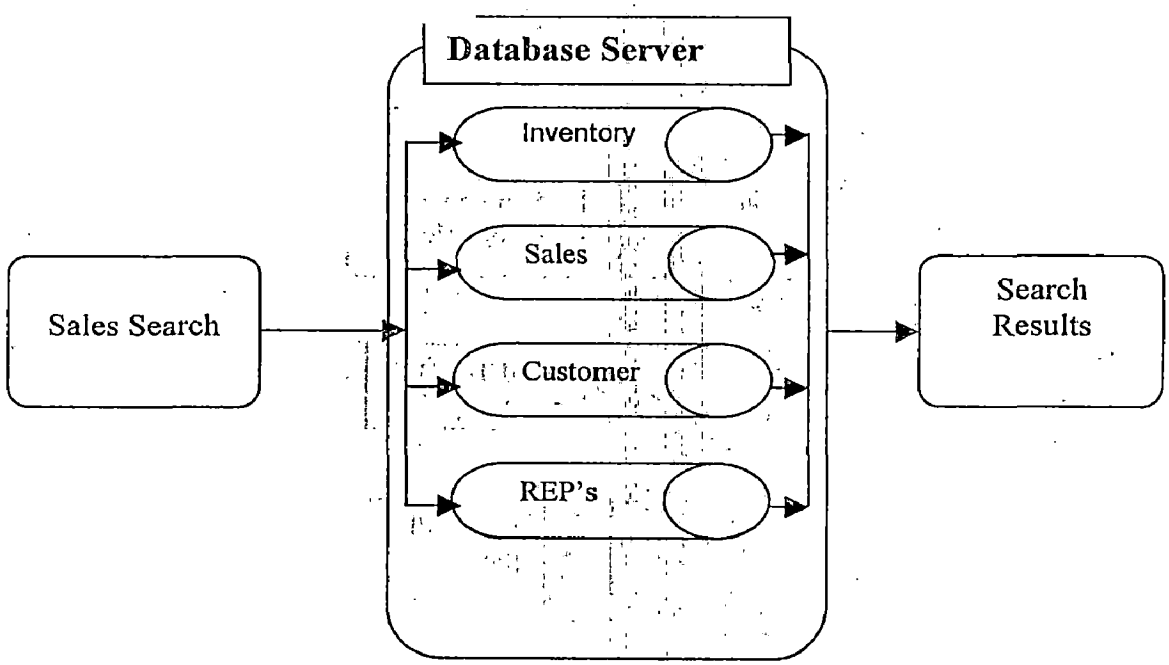


Figure [A.10] High-level data flow diagram for Search sub module.

REFERENCES

[1] Software Requirement Specifications of ISMA, Document Number:D08010

[2] Microsoft Developers Network (MSDN) January 2002.

[3] SQL Server Books Online.

[4] http://www.taltech.com

[5] http://www.citizen.co.uk

[6] http://www.citoh.co.uk

[7] C.Itoh Thermal Printer Commands Reference Manual.

[8] Evangelos Petroutsos ," Mastering Visual Basic 6 " First Edition 1998.