FURNITURE SHOP MANAGEMENT SYSTEM

ARCADE BUSINESS COLLEGE, PATNA

(AFFILIATED TO PATLIPUTRA UNIVERSITY)

Rajendra Nagar, Patna – 800016

A PROJECT ON FURNITURE SHOP MANAGEMENT SYSTEM

- PROJECT ID : PRJ2133N1
- SESSION : 2019-2022
- COURSE : BCA
- BATCH : (3–3)

ARCADE BUSINESS COLLEGE, PATNA

(AFFILIATED TO PATLIPUTRA UNIVERSITY)

Rajendra Nagar, Patna – 800016

GUIDED BY : -

ANUPAM SINGH

(H.O.D)

SUBMMITED BY:-

1. AMIT KUMAR

(TEAM LEADER)

2. ALOK KUMAR

3.VISHAL KUMAR

4. AKASH KUMAR

5.KUNDAN KUMAR

GROUP MEMBER'S DETAILS

NAME	COLLEGE ROLL NO	REGISTRATION NO.	MOBILE NO .
AMIT KR. (TEAM LEADER)	19175	194391114008051	8434706679
ALOK KR.	19198	194391314008010	7352090225
VISHAL KR.	19184	194391614008003	7700818001
AKASH KR.	19183	194391114008033	9798762640
KUNDAN KR.	19181	194391114008003	8825354784

SYNOPSIS

SYNOPSIS OF THE PROJECT

THIS PROJECT CAN MANAGE ALL INFORMATION ABOUT STOCK, PRICE, STOCK AVAILIBILITY, CUSTOMER RECOREDS, SUPLIER RECORD LIKE CONTACT NO OF SUPLIER, IT ALSO MAINTAIN PURCHASE AND SALES DETAILS AND GENERATE ELECTRONIC BILL SO THAT WE CAN

SEARCH RECORDS QUICKLY AND GENERATE ELECTRONIC REPORT WITHIN A SECOND

WE CAN USE THIS PROJECT IN ANY SHOP AND MAINTAIN THEIR TRANSECTION WITH THE HELP OF COMPUTER AND ELECTRONICLLY IT ALSO REDUCE HUMAN EFFORT AND PAPER WORK.

I WOULD LIKE TO EXTEND MY SINCERE THANKS TO DIRECTOR OF “COLLEGE” WHO MADE AVAILABLE ALL POSSIBLE RESOURCES NECESSARY FOR MY PROJECT. FURTHER I WOULD LIKE TO GIVE REGARDS TO WHO HAD GIVEN THEIR KIND TIME TOWARDS MY PROJECT. FINALLY I WOULD GIVE MY WHOLE HEARTED THANKS FOR PROVIDING ALL NECESSARY REQUIRED DATA FOR THE PROJECT.

MAIN REPORT

ACKNOWLEDGEMENT

First and foremost,I would like to thanks supreme being for making me triumphed in completing this project. I am really thankfull to them. Secondly I would also like to thanks my my group members who helped a lot in finalizing this project in the limited time frame.

Thank to my HOD Mam

(MRS.ANUPAM SINGH) for guiding me to do this unique opportunity to do this subline project and for guiding me at every point. She had been very cooperative and helped me very much.

she is our honourable Project Guide Prof. Anupam Singh,

Assistant Professor in "Arcade Business College".

Thank You!

DECLARATION

We hereby declare to the project work entitled “FURNITURE SHOP MANAGEMENT SYSTEM” submitted to the 6 TH semester **ARCADE BUSINESS COLLEGE**,Rajendra Nagar,Patna(bihar) is a record of an original work done by our team under the guidance of **(MRS.ANUPAM SINGH)** HOD of my My College and this project is submitted in the partial fulfill of the requirement for the award of the degree in BCA.

This software developing under (visual basic 6.0) and (oracle 10g) with prof Khurshid Allam. the result embodied in the thesis have not been submitted to any other university or institute for the awards of any degree.

Thank You!

INDEX

Main Report

INTRODUCTION

This is Furniture shop in Small Town. MR.Sunil Kumar is the owner of the shop. It is situated in District - Seikhpura , Town - Barbigha, Pin- 811101. This Shop is Started in 14/Nov/2020.This Shop is Provides Multiple type of product Related to Furniture .

Furniture Showroom is the official platform independent, and with some of the furniture centers, building mutual assistance and cooperation relations, offline sales.This furniture site supplies facilities to buy offline basic , durable home and office furniture. This software for is created to help increase its sales as well as to acquire more customers in the furniture market. As well as;to treat every supplier, employee, and customer with honesty, dignity and respect,improve all aspects of service delivery to our customers, our employees and our community and to provide a safe and convenient environment to shop.

OBJECTIVE OF THE PROJECT

There are so many disadvantages in the existing system because they want to a log of paper work in the system and many people engage to maintain of the system. But the technology is growing fatly and researches a new technology.

So the cost of the control is very less and we also find previous information in a few seconds on the computerization by this Software. It improve efficiency of their bill according to the customer satisfaction.

The major objective for implementing a computerized Sofware in an organization is:-

- **To manage the resources of the Stock efficiently.**
- **To reduce the losses incurred due to wrong entries.**
- **To manage the Customer/member data in a proper manner.**
- **Also, manage the asset.**
- **To maintain the daily expense in efficient manner.**
- **Preparation of various analysis reports.**
- **Generation of report that help management for making effective and timely decisions.**

Tracking: Tracking the progress of your projects and programs is important for the long term success of your business. A project management software system will give you the visibility to see if

a project is running to its predetermined time frame, what has been done and what

still needs to be complete. The software reduces the need for paper or electronic updates from the project team members to their project manager. **Resourcing**: The project management system ensures that the

optimal resources are working on the right projects by examining the level of skills and abilities needed to successfully complete the project. Assigning resources to tasks or projects through the project management system will ensure that each resource knows all the information needed for them to complete the project.

Communication: A project management system makes communication channels regarding tasks, projects and programs easier through the uploading and sharing of files, collaborating on an online space and emailing updates through the system. Communication channels with your customers/ clients can be made simpler also as business documents e.g. invoices can be sent directly from the project management system to its required destination and information on those customers/ clients can be kept on a virtual database to help with the follow up process that takes place.

Financial Control: A critical item in the management of projects is the budget and the way in which the budget is managed could ultimately decide the future of the business. The project management software gives the project manager a visual on the project costs, project budget (actual Vs planned), project cost data etc to ensure that projects and programs run to their assigned budgets, money is ultimately saved and not lost and generating a clear Return on Investment (ROI) on all projects.

Where are we now?: A PM software solution can easily tell an organization if they are in line with their initial requirements by taking a snapshot of the project at that point in time and examining some of the following:

Who is working on each task or project

Is there any time delays

Is the project keeping to its budget etc

Decision Making: When making a decision in project management all aspects of the project need to be taken into consideration. By using a project management system the decision making process is improved, as the information is all in clearly presented in one place.

The Edge: By having a project management system in place it could ultimately give you a competitive edge over your

competitors. If you can complete a project more efficiently than the rest you it can give you that extra push ahead of your competitors.

Risk management: Out of nowhere a risk can throw a spanner in the works and cause severe problems resulting in the project failing. Project management software system can give you a platform by which risks can be flagged, tracked and correctly resolved to ensure that the risk has the lowest possible negative effect on the project.

Quality of information: The quality of the information is increased as only the information that's needed is gathered and shown.

Keeping the boss happy: Ok this won't provide the company with benefits however it will give you some peace and reassurance at your next meeting with the boss. The project management system provides you with all the tools to help your project come in on time and too budget keeping all involved happy.

.**USER INFORMATION:**

- The project is designed in such a manner that it can provide all the detail information of USERS

- **DATABASE INFORMATION**
  - This project is designed in such a manner that it can provide all the detail information of record of database.

- **REPORT GENERATION:**
  - This project is designed to provide us various Summary and detailed report of stored data we can also filter our report using various filter.

SCOPE OF THE PROJECT

The scope of the project includes that what all future enhancements can be done in this system to make it more feasible to use :

- Databases for different products range and storage can be provided.
- Multilingual support can be provided so that it can be understandable by the person of any language.
- More graphics can be added to make it more user- friendly and understandable.
- Manage and backup versions of documents online.

Scope creep management is important for effective project management. Projects are expected to meet strict deadlines with resource restraints, and an unvetted and unapproved change in the scope can affect the success of the project. Scope creep sometimes causes cost overrun.

Scope creep is a term which refers to the incremental expansion of the scope of a project, which may include and introduce more requirements that may not have been a part of the initial planning of the project, while nevertheless failing to adjust schedule and budget. There are two distinct ways to separate **scope creep management**. The first is **business scope creep**, and the second is called **features (also technology) scope creep**. The type of scope creep management is always dependent on the people who create the changes.

Business scope creep occurs when decisions that are made with reference to a project are designed to solve or meet the requirements and needs of the business. Business scope creep changes may be a result of poor requirements definition early in development, or the failure to include the

users of the project until the later stage of the systems development life cycle. Management system. Items deemed out of scope go directly through the change control process and are not automatically added to the project work items. The Project Scope Management plan is included in as one of the sections in the overall Project Management plan. It can be very detailed and formal or loosely framed and informal depending on the communication needs of the project.

Features (Technology) scope creep occurs when the scope creep is introduced by technologists adding features not originally contemplated. *Customer-pleasing scope creep* occurs when the desire to please the customer through additional product features adds more work to the current project rather than to a new project proposal. *Gold-plating scope creep* occurs when technologists augment the original requirements because of a bias toward "technical perfectionism" or because the initial requirements were insufficiently clear or detailed.

PROJECT PLANING

PHASES	MEMBERS	TOTAL DAYS
1.ANALYSIS ➢ DATA GATHERING ➢ FEASIBILITY STUDY ➢ COST BENEFIT ANALYSIS ➢ PROJECT PROPOSAL	➢ AMIT KUMAR ➢ AMIT KUMAR ➢ ALOK KUMAR ➢ AMIT KUMAR	DAYS * 20
2.DESIGN	➢ ALOK KUMAR ➢ VISHAL KUMAR	DAYS *10
3.CODING	➢ VISHAL KUMAR ➢ AKASH KUMAR	DAYS *20
4.TESTING	➢ KUNDAN KUMAR ➢ ALOK KUMAR	DAYS *2
5.IMPLEMENTETION	➢ AKASH KUMAR ➢ VISHAL KUMAR	DAYS *8
6.DOCUMENTATION	➢ ALOK KUMAR ➢ KUNDAN KUMAR	DAYS*5

MODULO DESCRIPTION

- **Authentication:** In this module we provide security to the owner by providing username and password.

- **Stock**: This module is used to add the new items arrived to the Stock.

- **Sales:** This module describes the sale of the item. It also tells numbers of the sales item and amount specification for the item and details of the customer.

- **Billing**: This module is used to prepare bill to the user by specifying the details of the item like product type ,product details,customer details,address and contact number.

THEORETICAL BACKGROUND OF THE PROJECT

FRONT END (VB 6.0)

Visual basic 6.0 is the popular version of the programming language. This environment is to develop robust. Attend alone application and utilities is less time then it usually takes in other language. Project development has been all easier what with the language supporting the oops concept.

VISUAL BASIC is a VISUAL and **Event-driven Programming Language**. These are the main divergence from the old BASIC. In BASIC, programming is done in a text-only environment and the program is executed sequentially. In VB6, programming is done in a graphical environment. In the old BASIC, you have to write program code for each graphical object you wish to display it on screen, including its position and its color. However, In VB6 , you just need to drag and drop any graphical object anywhere on the form, and you can change its properties using the properties window.

In addition, Visual Basic 6 is **Event-driven** because we need to write code in order to perform some tasks in response to certain events. The events usually comprises but not limited to the user's inputs. Some of the events are load, click, double click, drag and drop, pressing the keys and more. We will learn more about events in later lessons. Therefore, a VB6 Program is made up of many subprograms, each has its own program code, and each can be executed independently and at the same time each can be linked together in one way or another.

The Integrated Development Environment :-

One of the most significant changes in Visual Basic 6.0 is the Integrated Development Environment (IDE). IDE is a term commonly used in the programming world to describe the interface and environment that we use to create our applications. It is called integrated because we can access virtually the entire development tool that we need from one screen called an interface. The IDE is also commonly referred to as the design environment, or the program. Integrated development is one in which we can develop, run, test and debug your applications.

The visual basic IDE is made up of a number of components:

1) Menu Bar
2) Tool Bar
3) Project Explorer
4) Properties window
5) Form Layout Window
6) Toolbox
7) Form Designers
8) Code Window

Menu Bar –The menu bar presents the Visual Basic menus. Here is a list of those menus and what they do.

- File – File handling and printing; also used to make EXE files.
- Edit – Standard editing functions, undo, searches.
- View – Displays or hides windows and toolbars.
-

- Project – Sets project properties, adds/removes forms and modules, and Adds/removes references and components.
- Format – Aligns or sizes controls.
- Debug – Starts/stops debugging and stepping through programs.
- Run – Starts a program, or compiles and starts it.
- Tools – Adds procedures, starts the Menu Editor, sets IDE options.
- Add-Ins – Add-in manager lists add-ins like Application Wizard and API Viewer.
- Window – Arranges or selects open windows.
- Help – Handles Help and the about box Toolbar – The main Visual Basic toolbar contains buttons matching popular menu items. Clicking the button is the same as selecting a menu item and can save some time. Besides the main toolbar, we can also display other dock able toolbars in Visual Basic: the Debug, Edit, and Form Editor toolbars. To display one of these toolbars, just select it using the Toolbars item in the View menu; the toolbar appears.

Project Explorer –Project Explorer Window allows us to coordinate the parts of our program into folders for easy manipulation. It gives us a valuable overview of our entire project, which is really very useful when a project gets large and contains many components.

Property Window –The properties window is docked under the project explorer window. It describes about the various characteristics of selected objects such as caption for label, text for text boxes. Here we can view and change the property of selected object.

Forms Layout Window – The form layout window is used to determine the initial positions of the forms in our application. This window is useful in applications that use multiple forms.

Toolbox – The toolbox contains a set of controls such as – text boxes, labels, list boxes, option buttons and much more. We can add these controls to our forms.

Form Designer –Form designers are really just windows in which a particular form appears. We can place controls into a form simply by drawing them after clicking the corresponding control's tool in the toolbox.

Code Window – We just place the code we want to attach to an object in the code window (to open an object's code in the code window, just double-click that object). There is two drop-down list boxes at the top of the code window: the left list lets us select the object to add code to, and the right list lets us select the procedure to add (all the methods the object supports appear in this list).

Programming in Visual Basic 6.0 –
Programming in VB 6.0 is a combination of visually arranging components or controls on a form, specifying attributes and

actions for those components, and writing additional lines of co de for more functionality.

Visual Basic supports several project types. Some of them are :-

- Standard Window EXE programs
- ActiveX EXE files
- ActiveX DLLs
- Program written by the Visual Basic Application Wizard
- Data Projects
- IIS(the Microsoft Internet Information Server) applications
- Visual Basic add-ins
- ActiveX document DLLs
- ActiveX document EXE files
- DHTML applications
- VB Enterprise Edition Controls

This list of project types indicates that there is a vast scope for different types of application development in VB 6.0, but it is primarily used to develop Windows applications and to interface database systems. Forms are created in VB 6.0 using drag-and-drop techniques. A tool is used to place controls (e.g., text boxes, buttons, etc.) on the form (window). Controls have attributes and event handlers associated with them. Default values are provided when the control is created, but may be changed by the programmer.

Many attribute values can be modified during run time based on user actions or changes in the environment, providing a dynamic application. When the user selects an element, an event handler

is called that executes code that the programmer created to perform the action for that list item.

For example, code can be inserted into the form resize event handler to reposition a control so that it remains centered on the form, expands to fill up the form, etc. By inserting code into the event handler for a key press in a text box, the program can automatically translate the case of the text being entered, or even prevent certain characters from being inserted.

Visual Basic 6.0 provides user with a large library of utility objects, and the language provides basic support for object-oriented programming. Unlike many other programming languages, Visual Basic is generally not case- sensitive—though it transforms keywords into a standard case configuration and forces the case of variable names to conform to the case of the entry in the symbol table. String comparisons are case sensitive by default.

The Parts of a Visual Basic Project –

Forms –Forms are the templates we base windows on. Besides standard forms, Visual Basic also supports Multiple Document Interface (MDI) forms, as well as a whole number of predefined form.

Modules –Modules are collections of code and data that function something like objects in object-oriented programming (OOP), but without defining OOP characteristics like inheritance, polymorphism, and so on. The point behind modules is to enclose procedures and data in a way that hides

them from the rest of the program. Breaking a large program into smaller, self-contained modules can be invaluable for creating and maintaining code.

Global Items –Global items are accessible to all modules and forms in a project, and we declare them with the Public keyword.

Control statements –Control statements are programming statements that affect program execution, based on the outcome of some logical comparison. There are two types of control statements. They are –

Decision Structures – These structures enable the program to make decisions on how it will operate based on test conditions. The decision structure can be of three types. They are – If-Then, If-Then-Else, Select case.

Looping Structures – These structures enable the program to execute a block of statement for a number of times based on a condition. They are while-wend, do-while loop, for-next. In addition to all these, Visual Basic 6.0 also supports the concept of different types of operators such as arithmetic operator, logical operators etc. It also supports the concept of array . Overall, Visual Basic 6.0 provides a programmer with a healthy environment to develop almost all kind of application.

BACK END (ORACLE 10g)

- Developer: Oracle Corporation
- Initial Release: 1979; 42-years ago.

A database that is a multi-model DataBase Management System (DBMS) produced and marketed by Oracle Corporation and Oracle 10g was released in 2003 which had support till 2006. Oracle database 10g furthermore reduces the time, cost and complexity of managing the database through the introduction of self-managing features such as the Automated Database, Diagnostic monitor, automated shared memory tuning, automated storage management and automated disk-based backup and recovery which essential in major power failures and sudden disconnection of data/internet outage etc. Oracle provides an application server and business applications, including the Ebusiness suite and the Oracle Collaboration suite. This edition (10g) of Oracle was for Grid Computing hence its name version contains a letter 'g'. Oracle database 10g was introduced as grid computing in 2003. It enabled organizations to virtualize computing resources by building a grid infrastructure based on low-cost commodity servers which are known as cloud servers accessed by internet across any country.

Every business enterprise maintains a large volume of data for its operations. With more and more people accessing these data for their work, the need to maintain its integrity and relevance increases. Normally, with the traditional methods of storing data and information in files, the chances that the data loses its

integrity, consistency and data redundancy are very high. So, there is always a need of more secured and computerized way of storing, accessing and manipulating data. DBMS serves this purpose.

DBMS stands for Database Management System. It is application software. It is collection of hardware and software. Hardware refers storage media (i.e. Hard Disk) for storage of Database. Whereas, software defines set of programs required to provide access to Database and their management.

Types of DBMS –

a. File Oriented DB
b. Database Management System (DBMS)
c. Relational Database Management System (RDBMS)
d. Object Oriented Database Management System (OODBMS)
e. Distributed Database Management System (DDBMS)
f. Knowledge Based Database Management System (KDBMS)

Models of DBMS –

1. 1st Model – Hierarchical Model
2. 2nd Model – Network Model
3. 3rd Model – ER Model
4. 4th Model – Relational Model
5. 5th Model – Object oriented Model

Oracle was originally developed by Lawrence Ellison (Larry Elision) and his two friends and former co-worker in 1977. Oracle DB runs on the most major platforms like Windows, UNIX, Linux and Mac OS. Oracle 10g is an Object Oriented Database Management System (OODBMS) having both the features of Object Oriented Programming an RDBMS. Oracle

Database XE provides an organized mechanism for storing, managing, and retrieving information.

Tables are the basic storage structure for holding data. The Oracle Database offers a wide range of options and features in the areas of Availability, Scalability, Analytics, Performance, Security, Management, Developers and Integration. These aim to enhance and complement existing database functionality to meet customer-specific requirements.

Oracle products are based on a concept known as the 'client/server technology'. This concept involves segregating the processing of an application between two systems. One performs all activities related to the database (server) and the other performs activities that help the user to interact with the application (clients).

A client or front-end database application also interacts with the database by requesting and receiving information from the database server. It acts as an interface between the user and the database. Further, it also checks for the validation against the data entered by the user. Oracle 10g can perform wide array of tasks. It acts as a transparent interface between the physical storage and the logical presentation of data. It provides a set of flexible and sophisticated tools for handling information.

Facilities provided by Oracle are:

- Defining Database.
- Creating Table in Database Query Processing.
- Add, edit and delete data from.

- Modify the structure of the database.
- Secure data from public access.
- Communicates with in the network, export and import data.

Oracle uses SQL (Structured Query Language) as its native language to enable users to instruct DBMS components.
SQL is a 4th Generation Language. It contains command based language which is known as Query. Each query contains its executable program.

The name SQL stands for Structural Query Language. SQL is a data access language, like any other language, it is used for communication. SQL communicates with database manager. The database manager could be Oracle, Informix, DB2 and SQL database. SQL is easy to learn. Despite the fact that SQL is a computer programming language, it is much simpler than traditional programming language like COBOL, BASIC, FORTRAN or API.
This is due to the fact that SQL is a nonprocedural language.

A database management system requires a query language to enable users to access data. Structured Query Language (SQL – pronounced 'sequel') is the language used by most relational database systems. IBM developed the SQL language in a prototype relational database management system –System R – in the mid-1970s. In 1979, Oracle Corporation introduced the first commercially available implementation of SQL.

Features of SQL –

SQL is a non-procedural language. You specify what information you require, not how to get it. In other words, SQL

does not require you to specify the access method to the data. All SQL statements use the query optimizer – a part of the RDBMS – to determine the fastest means of retrieving the specified data. This feature makes it easier for you to concentrate on obtaining the desired result.

SQL processes sets of records rather than a single record at a time. The most common form of a set of records is a table.

A range of user including DBA's, application programmers, management personnel, and many other types of end users can use SQL.

SQL provides commands for a variety of tasks including:

- Querying data.
- Inserting, updating and deleting rows in a table.
- Creating, modifying and deleting database objects • Controlling access to the database and database objects
- Guaranteeing database consistency.

All statements or query of SQL is divided into 4 sub languages according to access operation –

Data Definition Language – Creates, Alter, Drop, and Commands.

Data Manipulation Language – Insert, Select, Delete and Update. Transaction Control Language – Commit, Save point and rollback Data Control Language – Grant and Revoke commands.

Oracle greatly supports RDBMS features. Also it supports high security to the data and faster accessing capability. It can be run on a variety of platforms and operating systems. One can develop an application easily by providing user-friendly environment.

The features of oracle are portability and compatibility.

DEFINITION OF PROBLEM

As we know manual system are quite tedious, time consuming and less efficient and accurate in comparison to the computerized system. So following are some advantages of the old system:

1. Time consuming
2. Less accurate
3. Less efficient
4. Lots of paper work
5. Slow data processing
6. Not user friendly environment
7. Difficult to keep old records

SYSTEM ANALYSIS

AND

DESIGN

SYSTEM ANALYSIS AND DESIGN

FEASIBILITY STUDY:-

An initial investigation terminates in a proposal that determines whether an alternative system is feasible. Feasibility study can be categorized into three major parts: -

(1) TECHNICAL FEASIBILITY:-

- The proposed system has technical capacity of required to hold the data.

- This project is efficient and responds quickly for various enquires regardless
- Of number of locations.
- The system proposed could be expanded easily and Efficiency, whenever required.

(2) OPERATING FEASIBILITY STUDY:-

The management of the organization has a fully supported us to bring up the project and the data security in this project provided by setting up the password procedure so that only the authorized user can access the system.

(3) OUR PROJECT IS ECONOMICAL FEASIBLE AS:-

* It has computerized paper works and also is reduced to large extent
* With the help of this project single person is now available to do the tasks of 5 to 7 persons.
* Due to processing speed of then Computer, we can extract desired information's in a fraction of second.

HARDWARE AND SOFTWARE USED

HARDWARE USED:-

- **PROCESSOR -** PENTIUM-IV 945 GHZ
- **RAM –** 1 GB
- **HARD DISK –** 160 GB
- **MONITOR –** 17” SAMSUNG COLOR MONITOR
- **DVD WRITER SONY**
- **KEYBOARD –** 1.5 KEY MULTIMEDIA KEYBOARD
- **PRINTER –** HP LASER JET 6L PRINTER
- **MOUSE –** OPTICAL MOUSE IBALL

SOFTWARE USED:-

- **OPERATING SYSTEM –** XP HOME
- **DATABASE –** ORACLE(10g)
- **FRONT-END PACKAGE –** VISUAL BASIC 6.0

COST AND BENEFIT ANALYSIS

Cost and Benefit Categories

In developing cost estimates for a system, we need to consider several cost elements. Among them is hardware, personnel, facility, operating, and supply cost.

1 **Hardware cost** relate to actual purchase or lease of the computer and peripherals. Determining the actual cost of hardware is generally more difficult when the system is shared by various users than for a dedicated stand-alone system. In some cases, the best way to control for this cost is to treat it is an operating cost.

2 **Personnel cost** include EDP staff salaries and benefits as well as pay for those involved in developing the system.

3 **Facility costs** are expenses incurred in the preparation of the physical site where the application or the computer will be in operation. This includes wiring, flooring, acoustics, lighting, and air conditioning. These cost are treated as one time cost and incorporated into the overall cost estimate of the candidate system.

4 **Operating cost** include all cost associated with the day to day operation of the system, the amount depends on the number of shifts, the nature of the application, and the caliber of the operating staff. There are various ways of covering operating cost one approach is to treat operating cost as overhead. Another approach is to charge each authorized user for the amount of processing they request from the system. The amount charged is based on computer

time, staff time, and volume of output produced. In any case, some accounting is necessary to determine how operating cost should be handled.

5 **Supply costs** are variable costs that increase with increased use of paper, ribbons, disks, and the like. They should be estimated and included in the overall cost of the system.

DETAILED LIFE CYCLE OF THE PROJECT

All stages of a software process status quo, problem definition, technical development, and solution integration coexist simultaneously at some level of detail. design and analysis. Information engineering encompasses requirements gathering at the strategic business level and at the business area level.

Software requirements analysis. The requirements gathering process is intensified and focused specifically on software. To understand the nature of the program(s) to be built, the software engineer ("analyst") must understand the information domain for the software, as well as required function, behavior, performance, and interface. Requirements for both the system and the software are documented and reviewed with the customer.

Design. Software design is actually a multiuse process that focuses on four distinct attributes of a program: data structure, software architecture, interface representations, and procedural (algorithmic) detail. The design process translates requirements into a representation of the software that can be assessed for quality before coding begins. Like requirements, the design is documented and becomes part of the software configuration.

Code generation. The design must be translated into a machine-readable form.The code generation step performs this task. If design is performed in a detailed manner,code generation can be accomplished mechanistically.

Testing. Once code has been generated, program testing begins. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested, and on the functional externals; that is, conducting tests to uncover errors and ensure that defined input will produce actual results that agree with required results.

Implementation. Software will undoubtedly undergo change after it is delivered to the customer (a possible exception is embedded software). Change will occur because errors have been encountered, because the software must be adapted to accommodate changes in its external environment (e.g., a change required because of a new operating system or peripheral device), or because the customer requires functional or performance enhancements. Software support/maintenance reapplies each of the preceding phases to an existing program rather than a new one.

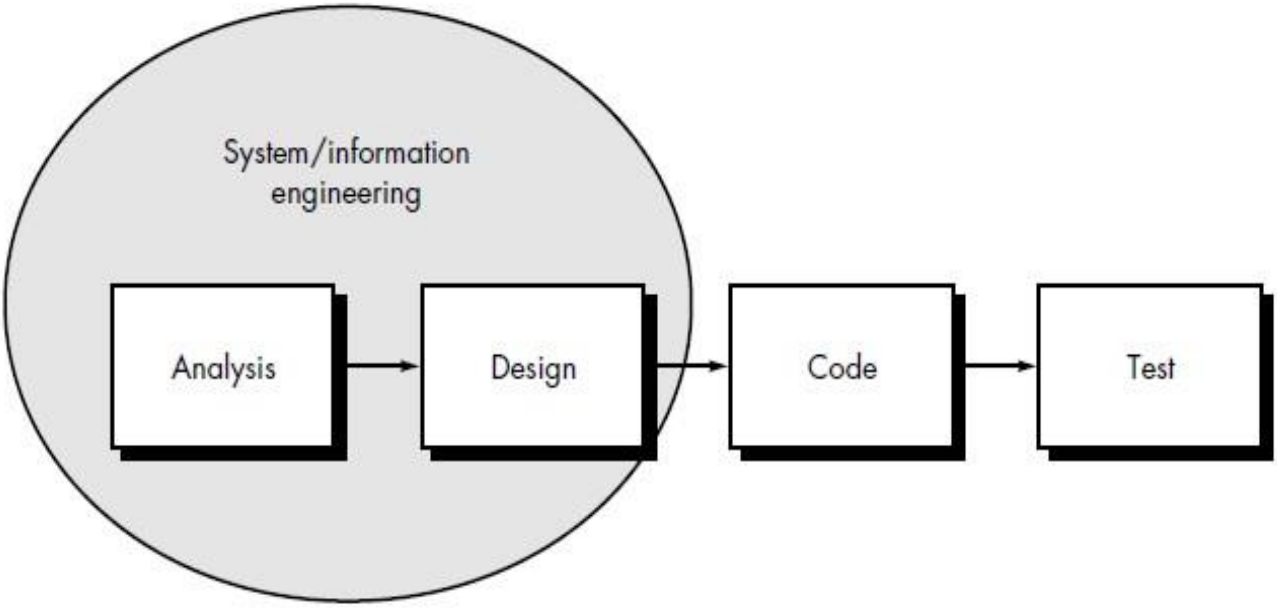


FIG. SDLC PHASE

E R DIAGRAM

E R DIAGRAM

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

History of ER models

Peter Chen (a.k.a. Peter Pin-Shan Chen), currently a faculty member at Carnegie-Mellon University in Pittsburgh, is credited with developing ER modeling for database design in the 1970s. While serving as an assistant professor at MIT’s Sloan School of Management, he published a seminal paper in 1976 titled “The Entity-Relationship Model: Toward a Unified View of Data.”

In a broader sense, the depiction of the interconnectedness of things dates back to least ancient Greece, with the works of Aristotle, Socrates and Plato. It’s seen more recently in the 19th and 20th Century works of philosopher-logicians like Charles Sanders Peirce and Gottlob Frege.

By the 1960s and 1970s, Charles Bachman (above) and A.P.G. Brown were working with close predecessors of Chen’s approach. Bachman

developed a type of Data Structure Diagram, named after him as the Bachman Diagram. Brown published works on real-world systems modeling. James Martin added ERD refinements. The work of Chen, Bachman, Brown, Martin and others also contributed to the development of Unified Modeling Language (UML), widely used in software design.

Uses of entity relationship diagrams

- Database design: ER diagrams are used to model and design relational databases, in terms of logic and business rules (in a logical data model) and in terms of the specific technology to be implemented (in a physical data model.) In software engineering, an ER diagram is often an initial step in determining requirements for an information systems project. It's also later used to model a particular database or databases. A relational database has an equivalent relational table and can potentially be expressed that way as needed.
- Database troubleshooting: ER diagrams are used to analyze existing databases to find and resolve problems in logic or deployment. Drawing the diagram should reveal where it's going wrong.
- Business information systems: The diagrams are used to design or analyze relational databases used in business processes. Any business process that uses fielded data involving entities, actions and interplay can potentially benefit from a relational database. It can streamline processes, uncover information more easily and improve results.
- Business process re-engineering (BPR): ER diagrams help in analyzing databases used in business process re-engineering and in modeling a new database setup.
- Education: Databases are today's method of storing relational information for educational purposes and later retrieval, so ER Diagrams can be valuable in planning those data structures.

- Research: Since so much research focuses on structured data, ER diagrams can play a key role in setting up useful databases to analyze the data.

The components and features of an ER diagram

ER Diagrams are composed of entities, relationships and attributes. They also depict cardinality, which defines relationships in terms of numbers. Here's a glossary:

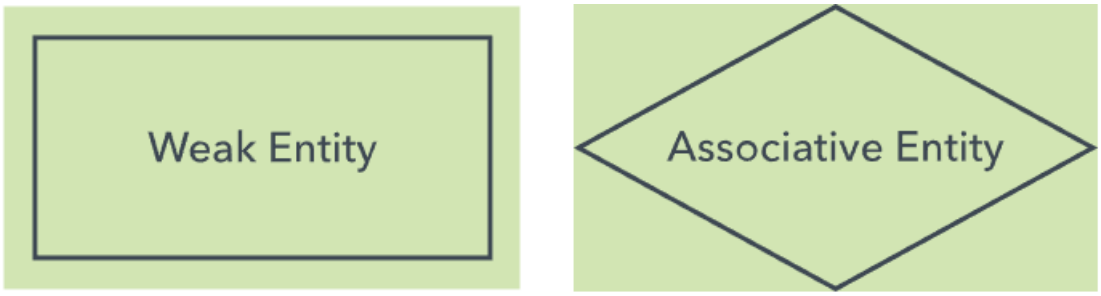
Entity

A definable thing—such as a person, object, concept or event—that can have data stored about it. Think of entities as nouns. Examples: a customer, student, car or product. Typically shown as a rectangle.

Entity type: A group of definable things, such as students or athletes, whereas the entity would be the specific student or athlete. Other examples: customers, cars or products.

Entity set: Same as an entity type, but defined at a particular point in time, such as students enrolled in a class on the first day. Other examples: Customers who purchased last month, cars currently registered in Florida. A related term is instance, in which the specific person or car would be an instance of the entity set.

Entity categories: Entities are categorized as strong, weak or associative. A strong entity can be defined solely by its own attributes, while a weak entity cannot. An associative entity associates entities (or elements) within an entity set.



Entity keys: Refers to an attribute that uniquely defines an entity in an entity set. Entity keys can be super, candidate or primary. Super key: A set of attributes (one or more) that together define an entity in an entity set. Candidate key: A minimal super key, meaning it has the least possible number of attributes to still be a super key. An entity set may have more than one candidate key. Primary key: A candidate key chosen by the database designer to uniquely identify the entity set. Foreign key: Identifies the relationship between entities.

Relationship

How entities act upon each other or are associated with each other. Think of relationships as verbs. For example, the named student might register for a course. The two entities would be the student and the course, and the relationship depicted is the act of enrolling, connecting the two entities in that way. Relationships are typically shown as diamonds or labels directly on the connecting lines.

Recursive relationship: The same entity participates more than once in the relationship.

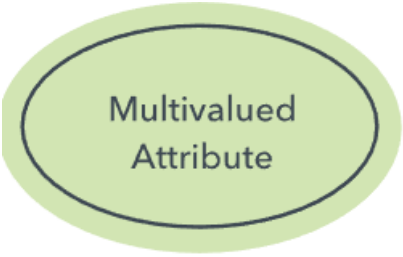
Attribute

A property or characteristic of an entity. Often shown as an oval or circle.

Descriptive attribute: A property or characteristic of a relationship (versus of an entity.)

Attribute categories: Attributes are categorized as simple, composite, derived, as well as single-value or multi-value. Simple: Means the attribute value is atomic and can't be further divided, such as a phone number. Composite: Sub-attributes spring from an attribute. Derived: Attributed is calculated or otherwise derived from another attribute, such as age from a birthdate.

Multi-value: More than one attribute value is denoted, such as multiple phone numbers for a person.



Single-value: Just one attribute value. The types can be combined, such as: simple single-value attributes or composite multi-value attributes.

Cardinality

Defines the numerical attributes of the relationship between two entities or entity sets. The three main cardinal relationships are one-to-one, one-to-many, and many-many. A one-to-one example would be one student associated with one mailing address. A one-to-many example (or many-to-one, depending on the relationship direction): One student registers for multiple courses, but all those courses have a single line back to that one student. Many-to-many example: Students as a group are associated with multiple faculty members, and faculty members in turn are associated with multiple students.

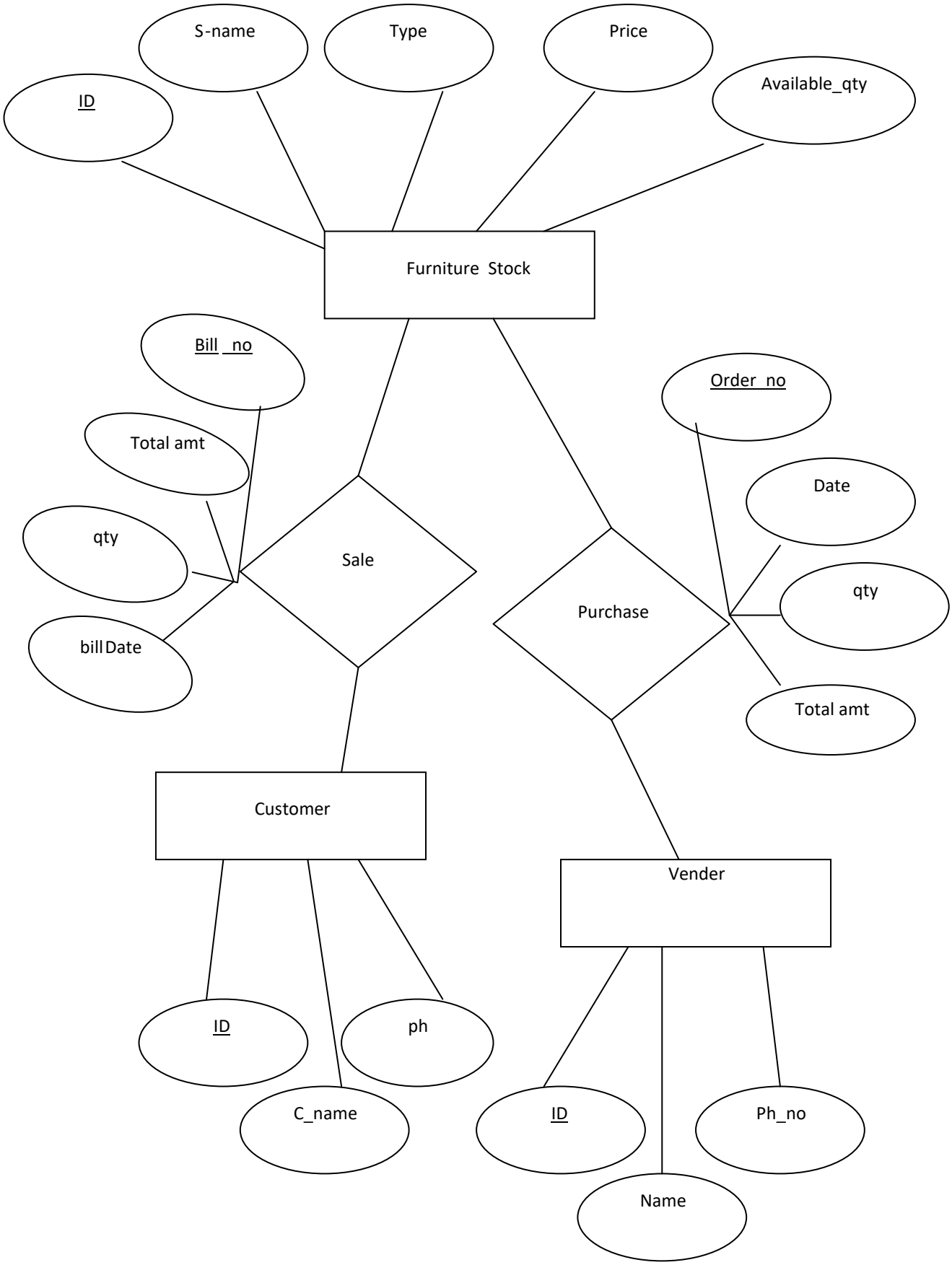

S-name
Type
Price
ID
Available_qty
Furniture Stock
Bill _no
Order_no
Total amt
Date
qty
Sale
Purchase
qty
billDate
Total amt
Customer
Vender
ID
ph
ID
Ph_no
C_name
Name

TABLE DESCRIPTION

LOG IN FROMS

Field name	Data type	Size	Constraints	Description
User_id	Varchar	20	Foreign key	User id
Password	Varchar	20	Foreign key	Password of user

EMP DETAILSH TABLE

Field name	Data type	Size	Constraints	Description
EMP_NO	Varchar2	5	Primary key	Employee number
EMP_NAME	Varchar2	20	NOT NULL	Employee name
ADDRESS	Varchar2	30	NOT NULL	Employee address
SALARY	number	10	NOT NULL	Employee salary
PHONE_NO	NUMBER	10	NOT NULL	EMPLOYEE NUMBER
DESIGNATION	Varchar2	10	NOT NULL	Employee designSSSSation
D_O_B	date	-	NOT NULL	Employee D_O_B
D_O_J	DATE	-	NOT NULL	Employee D_O_J
SEX	Varchar2	5	NOT NULL	Employee sex

PURCHASE DETAILS

Field name	Data type	Size	Constraints	Description
PURCHASE_ORDER_NO	number	5	Primary key	Purchase order number
P_DATE	date			Purchase date
CUSTOMER	Varchar2	25	NOT NULL	customer
CATEGORY	Varchar2	25	NOT NULL	Product category
ITEM	Varchar2	20	NOT NULL	Category item
AMT	number	8	NOT NULL	Amount of item
QTY	number	8	NOT NULL	Quantity of item
TAMT	number	8	NOT NULL	Total amount

MAIN SERVICE DETAILS

Field name	Data type	Size	Constraints	Description
SERVICE_NAME	Varchar	20	NOT NULL	Service name

SUB SERVICE DETAILS

Field name	Data type	Size	Constraints	Description
SERVICE NAME	Varchar2	20	Foreign key	Service name
SUB_SERVICE_NAME	Varchar2	20	PRIMARY KEY	Sub service name
MEASURED_IN	number	5	Not null	Measurement of product

EMPLOYEE DETAILS

Field name	Data type	Size	Constraints	Description
EMP_ID	Varchar2	8	Primary key	Employee id
EMP_NAME	Varchar2	25	Not null	Employee name
EMP_SEX	Varchar2	5	Not null	Employee sex
ADATE	Date	-		Date of joining
EMPATT	Varchar2	10	Not null	Emp attendence

DATA FLOW DIAGRAM

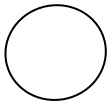
DFD is a graphical representation of data process and files used in a support system.Data Flow Diagrams are useful tools for analyzing existing systems. Data Flow Diagram is a network that describes flows of data and the processes that changes or transforms the data throughout a system.

Data flow diagrams can be expanded to show successive levels of details sufficient.

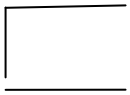
Expansion should be performed during the initial investigation to be certain that both the analyst and user personnel share a common understanding of the existing system and its data flow.

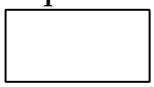
The different symbols used in the data flow diagram are:-

1. Circle — It is used form transfer the input to output. Indicate process that transfer data flow.

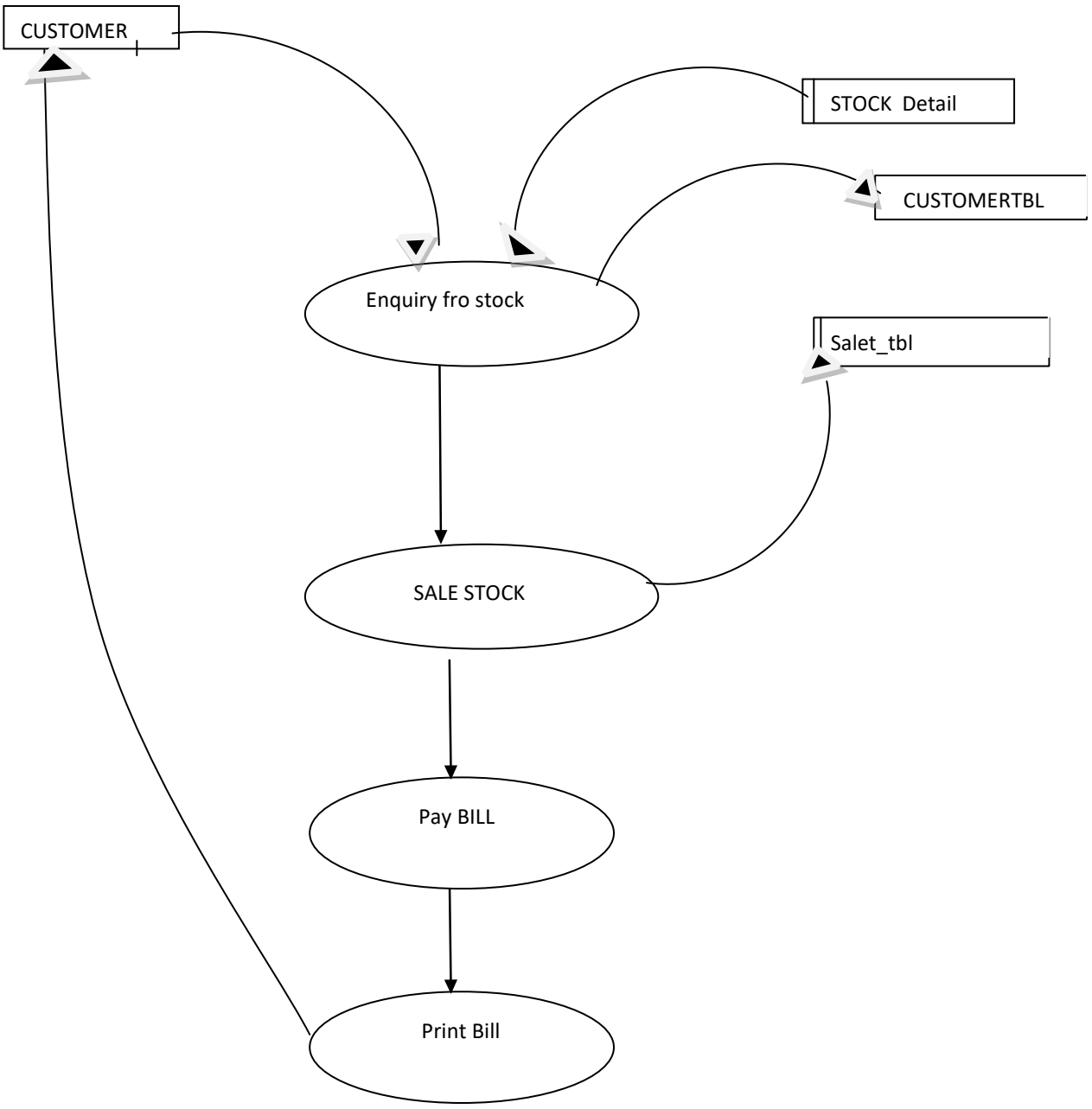


4. Arrows — Arrows are sued to show flow of path from where it is



3. Square  Squares indicates the source of destination .

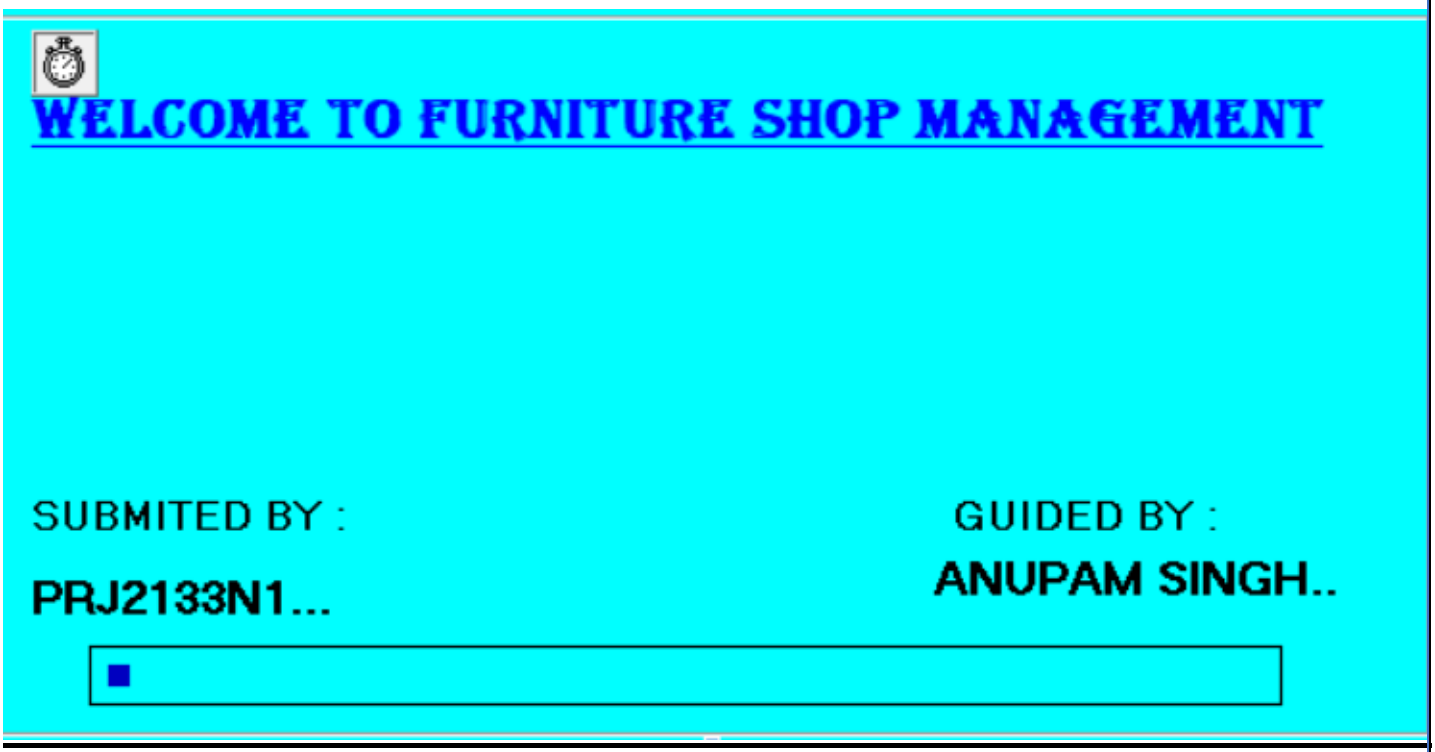
coming & going.


CUSTOMER
STOCK Detail
CUSTOMERTBL
Enquiry fro stock
Salet_tbl
SALE STOCK
Pay BILL
Print Bill

INPUT AND OUTPUT SCREEN DESIGN

Login Form

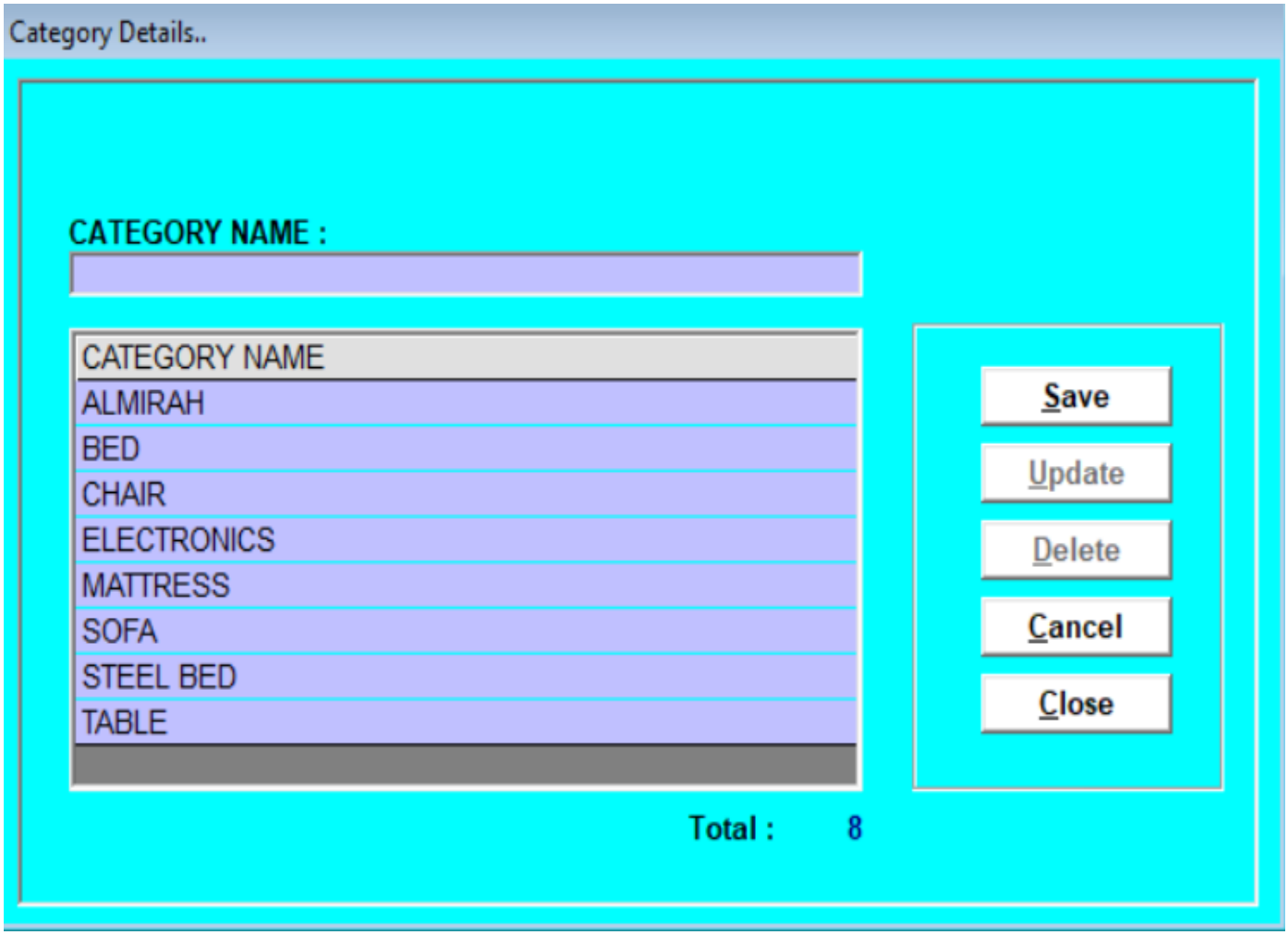
SPLASH FORM


WELCOME TO FURNITURE SHOP MANAGEMENT
SUBMITED BY :
PRJ2133N1...
GUIDED BY :
ANUPAM SINGH..

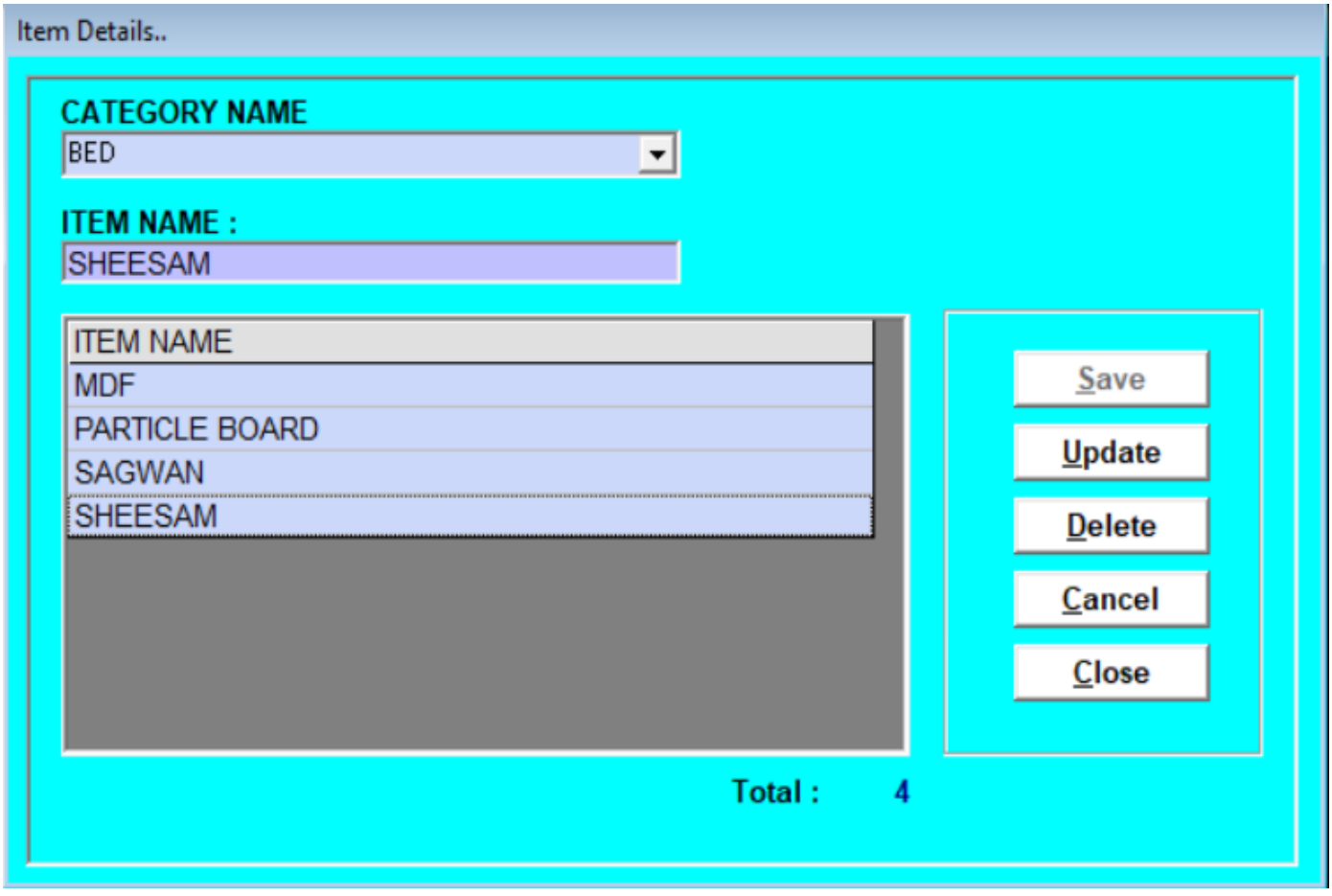
MDI FORM


FURNITURE SHOP MANAGEMENT.....
MASTER DETAILS
TRANSACTION
VIEW REPORTS
ABOUT US
EXIT
M S
FURNITURE
WORLD

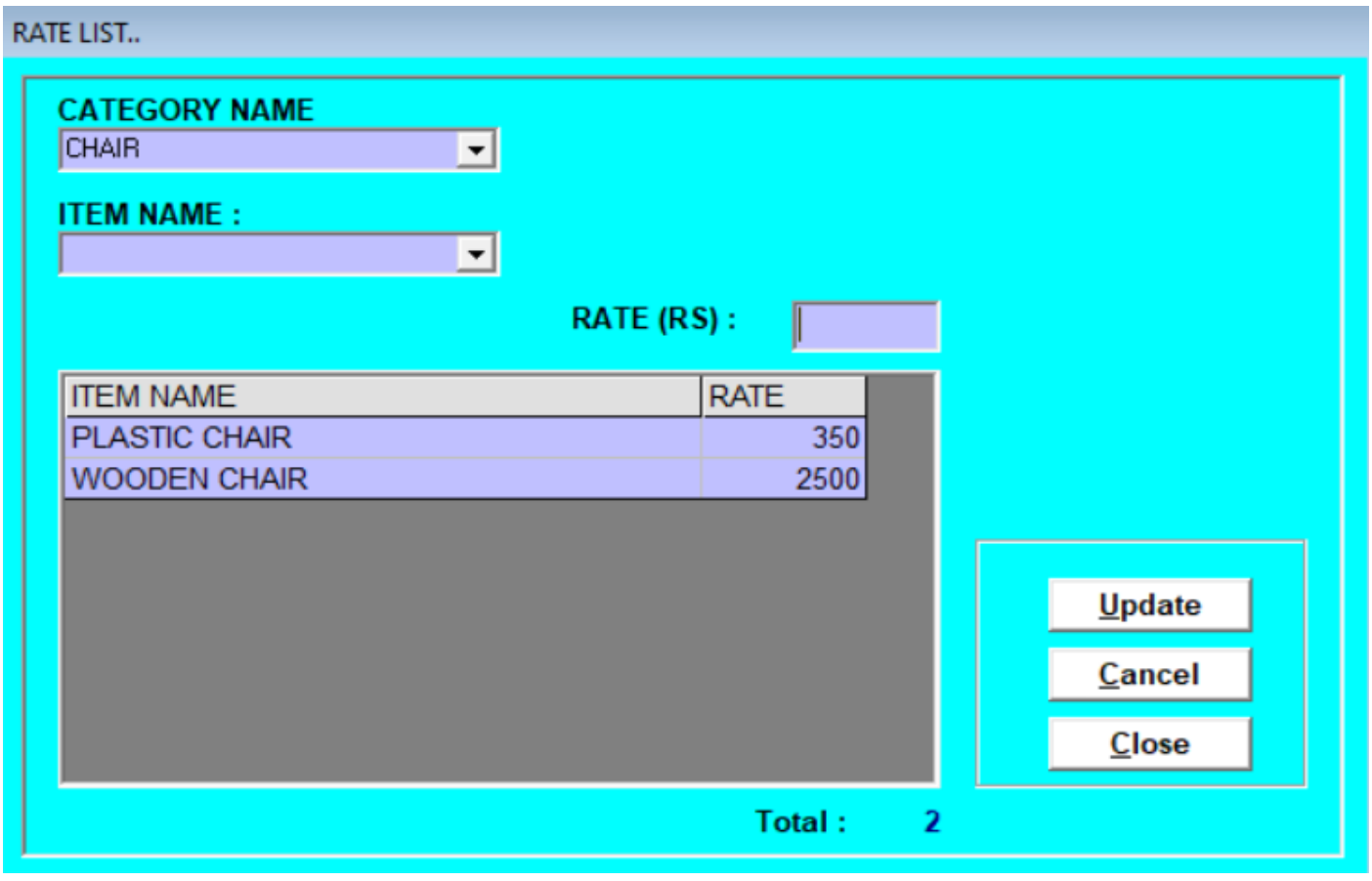
CATEGORY DETAILS ENTRY FORM



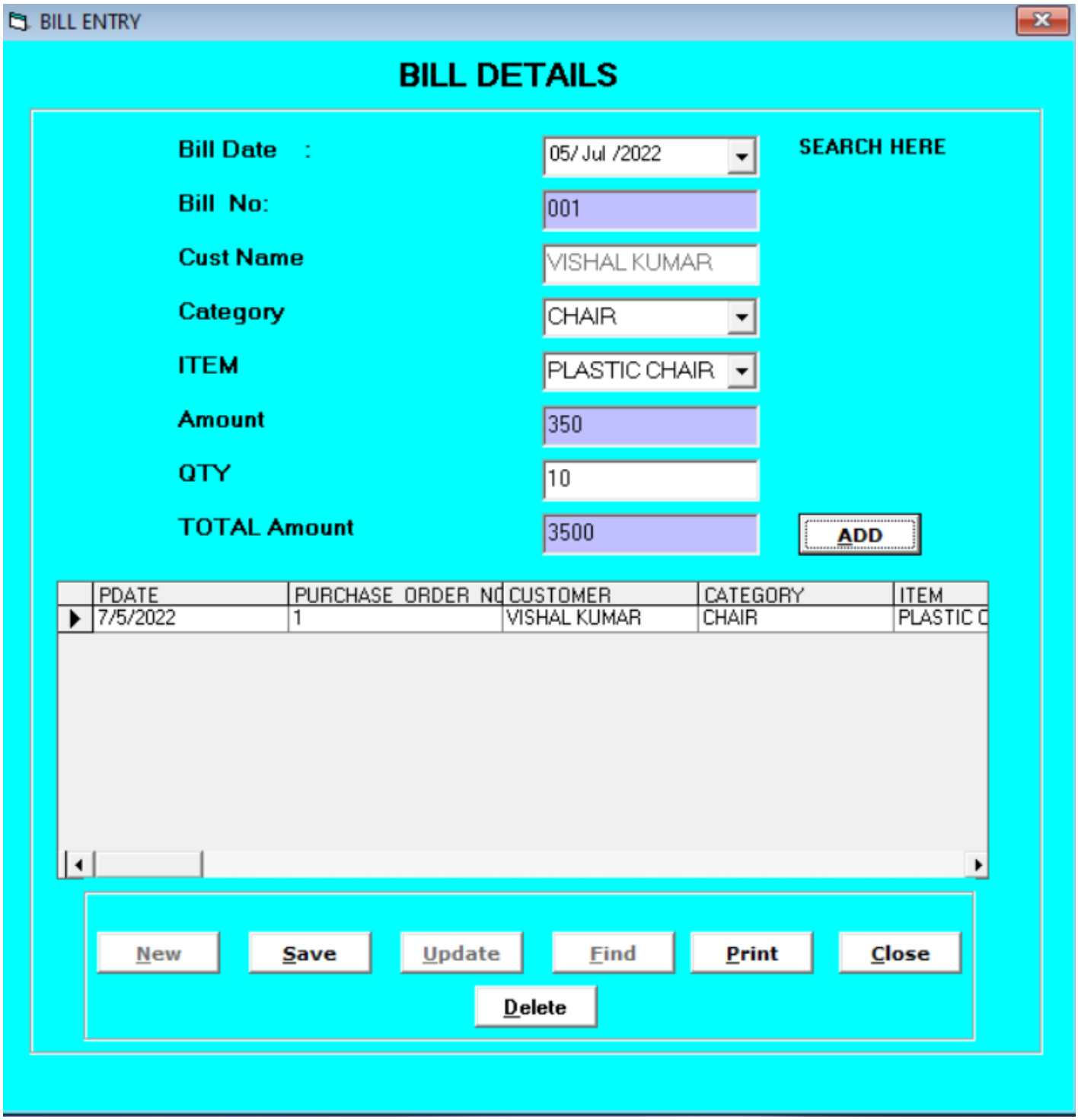
ITEM DETAILSH ENTRY FORM



RATE LIST ENTRY FORM



BILL DETAILSH FORM



RATE LIST REPORT FROM

RATE LIST REPORT

CATEGORY	ITEM NAME	AMT
PIZZA	CHEESWE PIZZA	20
SANWITCH	CORNSANWITCH	30
ALMIRAH	MDF	
ALMIRAH	PARTICLE BOARD	
ALMIRAH	WOOD	
BED	MDF	
BED	PARTICLE BOARD	
BED	SAGWAN	
BED	SHEESAM	
STEEL BED	STEEL BED	
SOFA	WOOD	
TABLE	DRESSING TABLE	
TABLE	CENTER TABLE	
TABLE	DINING TABLE	
TABLE	STUDY TABLE	
TABLE	OFFICE TABLE	
CHAIR	PLASTIC CHAIR	
CHAIR	WOODEN CHAIR	
MATTRESS	MATTRESS	
ELECTRONICS	FREEZER(WHIRPOOL)	
ELECTRONICS	FREEZER(LG)	
ELECTRONICS	IRON(USHA)	
ELECTRONICS	COOLER(SYMPHONY)	
ELECTRONICS	COOLER(ORIENT)	
ELECTRONICS	T.V. (L.E.D)	
ELECTRONICS	FAN (SELLING FAN)	

BILL REPORT FORM

BILL REPORT

BILL	Date	customer	From	Qty	PRICE	Amount
1	7/5/20	VISHAL	PLASTIC	10	350	3500
					TOTAL:	3500

CUSTOMER REPORT

FURNITURE SHOP

DATE : 7/5/2022

BILL NO: 1

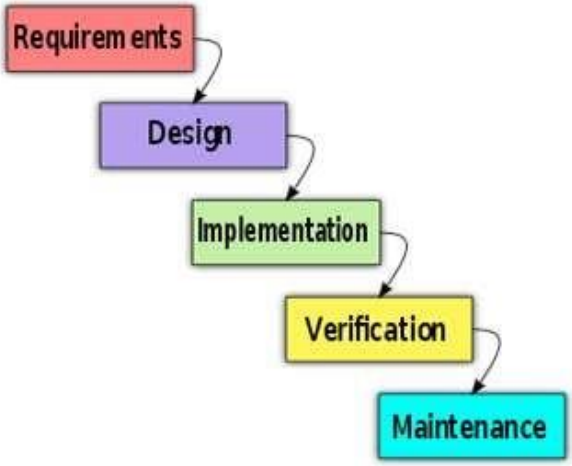
NAME : VISHAL KUMAR

PLASTIC CHAIR	10	350	3500.00

TOTAL: **3500.00**

PROCESS INVOLVED

WaterFall Model: The stages of "The Waterfall Model" are:



Requirement Gathering & Analysis:-This is the starting phase of the SDLC in which all possible system requirements are captured & analyzed. Software requirements specification includes the complete information about how actual end users are expecting from the system. This document covers all the necessary requirements for the development of project. Finally after completion of requirement gathering & analysis (validation of requirement against the user needs), a Requirement Specification document is created which give out as a input to the next phase of SDLC. In this model once we moved to the next phase then it won't possible to add or update the requirements.

System & Software Design: Prior to start actual coding, it is mandatory to be aware of what all features we are going to implement & how it would look like? The requirement specifications document created in the first phase is used as the input to this phase & based on this the system design specifications would be prepared. In which all hardware & system

requirements would be specified. The system design specifications document prepared in this phase is used as input for the Implementation & Testing phase model.

Implementation & Unit Testing: Upon getting the system design specifications document the actual coding would be started. Before starting the actual coding requirements are divided into the models/units. In the actual coding the initially develop the small programs called units. After implementing all units the integration would be started in the next phase. The developed unit's model functionality is tested separately in this phase to check whether the unit models are meets the specified requirements & this individual model testing is called as **Unit Testing**.

Integration & System Testing: In the previous model the system requirements are divided into models & each model is developed & tested separately. In this phase all units are integrated & done the system testing to check whether the all models/units are integrated properly or not & the system as whole doing as mention as per the system requirement document, so we call this phase as the Integration & System Testing. Upon complete testing of software is done then actual Software is successfully send to customer.

Operations & Maintenance: This phase is the never ending phase of the Waterfall Model. The problems are comes in picture after **Implementation & Unit Testing** phase. The issues found after the implementation phase i.e. not found in the period of the development life cycle. Some of the issues are not catch after testing cycle done so those are implemented in the maintenance phase, so this phase is called as Operations & Maintenance phase.

Advantages of Waterfall Model:

- This is linear simple model to implement & easy to maintain.
- In the SDLC initial phase spent time on reviewing requirements and design which saves the time later.
- Required resources are minimum in this model as compare to other.
- After every phase of the model a document in created which help & simpler to understand & design the system.
- Upon completion of coding, is done to check for implemented code or correctness of system.
- For each stage deadlines can be set which will help to develop the system on decided time frame.

Disadvantages of Waterfall Model:

- The biggest disadvantages of such system is it won't allow to go back. If the problems in the design phase which creates complication in the implementation phase.

- This model is rigid model. Not flexible & make changes is not possible in the development of system.

- Unable to start the next phase before completing the previous phase so time consuming.

- In the deployment phase all requirements are not covered or all requirements are not cover so adding such requirements may create unsuitable system.

- To make it customer happy such new requirement need to be implemented in new version of system which leads to added cost to system development.

SOURCE CODE

LOGIN FORM

```
    Private Sub cmdcancel_Click()
End
End Sub


     Private Sub CMDOK_Click()
If LCase(text1.Text) = "admin" And LCase(Text2.Text) = "admin" Then
   Me.Hide
   splash.Show
   Unload Me
Else
   MsgBox ("Invalid user name or Password"), vbInformation
   Text2.SetFocus
End If
End Sub


Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
   Text2.SetFocus
End If
```

```
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
   CMDOK.SetFocus
End If
End Sub
```

SPLASH FORM

```
Private Sub Timer1_Timer()

    If Label7.Width > 7000 Then
   Timer1.Enabled = False
   mdimain.Show
   Unload Me
Else
   Label7.Width = Label7.Width + 50

End If
End Sub
```

MDI FORM

```
Private Sub c_Click()
Load frmchanges
frmchanges.Show
End Sub


Private Sub C_M_Click()
  DataReport5.Show
End Sub


Private Sub cl_Click()
End
End Sub


Private Sub cm_Click()
Load frmcashmemo
 frmcashmemo.Show
End Sub
```

```
Private Sub E_D_Click()

 Load DataReport2
 DataReport2.Show
 End Sub


Private Sub ed_Click()
Load frmemp_detail
frmemp_detail.Show
End Sub


Private Sub MDIForm_Unload(Cancel As Integer)
End
End Sub


Private Sub mnuare_Click()
Load frmrep1
frmrep1.Show
End Sub


Private Sub MNUCAT_Click()
SERVICE_DETAILS.Show
```

```
End Sub


Private Sub MNUCHANGE_Click()
RATELIST.Show
End Sub


Private Sub mnuempa_Click()
Load FrmATT
FrmATT.Show
End Sub


Private Sub MNUITEM_Click()
SUB_SERVICE_DETAILS.Show
End Sub


Private Sub MNURATE_Click()
DataReport3.Show
End Sub


Private Sub pr_Click()
Load frmpurchase_order
```

```
frmpurchase_order.Show
End Sub


Private Sub print1_Click()
CommonDialog1.ShowPrinter
End Sub


Private Sub PUR_Click()
 Form1.Show
End Sub


Private Sub s_Click()
Load frmStock_detail
frmStock_detail.Show
End Sub


Private Sub sa_Click()
Load frmsalesdetail
frmsalesdetail.Show
End Sub


Private Sub SAL_Click()
```

```
 Load DataReport3
 DataReport3.Show
End Sub


Private Sub STO_Click()
 Load DataReport2
 DataReport2.Show
End Sub
```

CATEGORY FORM

```
Dim P_DEPT_OLD As String
Dim CpyUpdate As String


Private Sub cmdcancel_Click()
   Call Form_Load
End Sub


Private Sub cmdClose_Click()
   Unload Me
End Sub


Private Sub cmddel_Click()
   If MsgBox("IT WILL ALSO DELETE ALL METIRIAL OF THIS SERVICE" &
vbCrLf & "                    ARE YOU WANT TO DELETE.", vbYesNo) =
vbYes Then


      con.Execute "DELETE FROM MAIN_SERVICE where
SERVICE_NAME = '" & Trim(UCase(TXTDEPT.Text)) & "'"
      Call Form_Load
      TXTDEPT.SetFocus
```

```
        MsgBox "Recored Deleted", vbInformation
    End If
End Sub


Private Sub cmdSave_Click()
    If TXTDEPT.Text = "" Then
    MsgBox "Please Enter SERVICE Name", vbInformation
    TXTDEPT.SetFocus
    Exit Sub
    End If
    sql = "INSERT INTO MAIN_SERVICE (SERVICE_NAME) VALUES('" &
UCase(Trim(TXTDEPT.Text)) & "')"
    con.Execute sql
    Call Form_Load
    TXTDEPT.SetFocus
    MsgBox "Recored Saved", vbInformation
End Sub



Private Sub CMDUP_Click()
    If (TXTDEPT.Text = "") Then

  MsgBox "Please Enter SERVICE Name", vbInformation
    TXTDEPT.SetFocus
    Else
```

```
    con.Execute ("UPDATE MAIN_SERVICE SET SERVICE_NAME = '" &
UCase(Trim(TXTDEPT.Text)) & "' where SERVICE_NAME = '" &
CpyUpdate & "'")
    Form_Load
    MsgBox "Record Updated Successfully", vbInformation
    TXTDEPT.SetFocus
    End If
End Sub



Private Sub Form_Load()
    TXTDEPT.Text = ""
    If RS.State = 1 Then RS.Close
    RS.Open "SELECT * FROM MAIN_SERVICE order by
SERVICE_NAME", con
    msg1.Rows = 1
    If RS.RecordCount >= 1 Then
    While RS.EOF = False
    msg1.AddItem RS(0) & Chr(9)
    RS.MoveNext
    Wend
    End If

 cmdsave.Enabled = True
    CMDUP.Enabled = False
```

```
    CMDDEL.Enabled = False
    Lbltotal.Caption = RS.RecordCount
End Sub



Private Sub msg1_Click()
On Error GoTo Er
If Err.Description = "" Then
If Administrator_Password = False Then
    MsgBox "You Have No Permission to Change Details.Please Logon
the Administrator first.", vbInformation
    Exit Sub
Else
    If RS.State = 1 Then RS.Close
    RS.Open "SELECT * FROM MAIN_SERVICE where SERVICE_NAME ='"
& msg1.TextMatrix(msg1.RowSel, 0) & "'", con, adOpenKeyset,
adLockOptimistic, adCmdText
    If RS.RecordCount >= 1 Then
    While RS.EOF = False
    TXTDEPT.Text = RS(0)
    RS.MoveNext
Wend

    cmdsave.Enabled = False

 CMDUP.Enabled = True
```

```
        CMDDEL.Enabled = True


    Else
        MsgBox "record not found", vbInformation
    End If
    End If
    Else
Er:
    MsgBox "No Row Selected", vbInformation
    End If
    CpyUpdate = TXTDEPT.Text
    End Sub
```

SUB CATEGORY FORM

```
Dim P_DEPT_OLD As String
Dim CpyUpdate As String


Private Sub cmbser_Click()
    If RS.State = 1 Then RS.Close
    RS.Open "SELECT * FROM SUB_SERVICE where service_NAME ='" &
cmbser.Text & "'  order by SUB_SERVICE_NAME", con, adOpenKeyset,
adLockOptimistic, adCmdText

  msg1.Rows = 1
    If RS.RecordCount >= 1 Then
    While RS.EOF = False
    msg1.AddItem RS(1) & Chr(9)
    RS.MoveNext
    Wend
    End If
    Lbltotal.Caption = RS.RecordCount
End Sub


Private Sub cmbser_KeyPress(KeyAscii As Integer)

    If (Command1.Caption = "New") Then
```

```
    End If
End Sub


Private Sub cmdcancel_Click()
    Call cmbser_Click
    TXTDEPT.Text = ""
    TXTDEPT.SetFocus

    cmbser.Visible = True
    Text1.Text = ""
    Text1.Visible = False


End Sub

Private Sub cmdClose_Click()
    Unload Me
End Sub



Private Sub cmddel_Click()
    If MsgBox("IT WILL ALSO DELETE ALL RATE LIST OF THIS SUB
SERVICE" & vbCrLf & "                    ARE YOU WANT TO DELETE.",
vbYesNo) = vbYes Then
```

```
        con.Execute "DELETE FROM SUB_SERVICE where
SUB_SERVICE_NAME = '" & Trim(UCase(TXTDEPT.Text)) & "'"
        Call cmbser_Click
        TXTDEPT.Text = ""
        TXTDEPT.SetFocus
        cmdsave.Enabled = True
        CMDUP.Enabled = False
        CMDDEL.Enabled = False
        MsgBox "Recored Deleted", vbInformation
    End If
End Sub



    Private Sub cmdSave_Click()

        If TXTDEPT.Text = "" Then

     MsgBox "Please Enter SUB SERVICE Name", vbInformation
        TXTDEPT.SetFocus
    Exit Sub
    End If
        sql = "INSERT INTO SUB_SERVICE
    (SERVICE_NAME,SUB_SERVICE_NAME) VALUES('" &
    UCase(Trim(cmbser.Text)) & "','" & UCase(Trim(TXTDEPT.Text)) & "')"
        con.Execute sql
        Call cmbser_Click
```

```
    TXTDEPT.Text = ""
    TXTDEPT.SetFocus
    MsgBox "Recored Saved", vbInformation
End Sub



Private Sub CMDUP_Click()
    If (cmbser.Text = "") Then
    MsgBox "Please Select SERVICE Name", vbInformation
    cmbser.SetFocus
Exit Sub
End If
    If (TXTDEPT.Text = "") Then
    MsgBox "Please Enter SUB SERVICE Name", vbInformation

    TXTDEPT.SetFocus
Else
    con.Execute ("UPDATE SUB_SERVICE SET SUB_SERVICE_NAME = '"
& UCase(Trim(TXTDEPT.Text)) & "', SERVICE_NAME = '" &
UCase(Trim(cmbser.Text)) & "' where SUB_SERVICE_NAME = '" &
CpyUpdate & "'")
    cmbser_Click
    MsgBox "Record Updated Successfully", vbInformation
    TXTDEPT.SetFocus
    cmdsave.Enabled = True
    CMDUP.Enabled = False
```

```
    CMDDEL.Enabled = False
    TXTDEPT.Text = ""
End If
End Sub




Private Sub Command1_Click()
If Command1.Caption = "New" Then
    Text1.Visible = True
    cmbser.Visible = False
    Text1.SetFocus
    Command1.Caption = "Save"
Else
    If Text1.Text = "" Then

   MsgBox "Please Enter SERVICE Name", vbInformation
    Text1.SetFocus
Exit Sub
End If
    sql = "INSERT INTO MAIN_SERVICE (SERVICE_NAME) VALUES('" &
UCase(Trim(Text1.Text)) & "')"
    con.Execute sql
    cmbser.Visible = True
    cmbser.AddItem UCase(Trim(Text1.Text))
    cmbser.Text = UCase(Trim(Text1.Text))
```

```
    Text1.Text = ""
    Text1.Visible = False
    cmbser_Click
    TXTDEPT.SetFocus
    Command1.Caption = "New"
End If
End Sub



Private Sub Form_Load()
    TXTDEPT.Text = ""
    cmdsave.Enabled = True
    CMDUP.Enabled = False
    CMDDEL.Enabled = False
    If RS.State = 1 Then RS.Close


 RS.Open "SELECT * FROM MAIN_SERVICE order by SERVICE_NAME",
con, adOpenKeyset, adLockOptimistic, adCmdText
    cmbser.Clear
    If RS.RecordCount >= 1 Then
    While RS.EOF = False
    cmbser.AddItem RS(0)
    RS.MoveNext
Wend
End If
```

```
    RS.Close
End Sub



Private Sub msg1_Click()
On Error GoTo Er
If Err.Description = "" Then
If Administrator_Password = False Then
    MsgBox "You Have No Permission to Change Details.Please Logon
the Administrator first.", vbInformation
    Exit Sub
Else
    If RS.State = 1 Then RS.Close
    RS.Open "SELECT * FROM SUB_SERVICE where
SUB_SERVICE_NAME ='" & msg1.TextMatrix(msg1.RowSel, 0) & "'",
con, adOpenKeyset, adLockOptimistic, adCmdText
    If RS.RecordCount >= 1 Then

 While RS.EOF = False

   TXTDEPT.Text = RS(1)
    cmbser.Text = RS(0)
    RS.MoveNext
Wend

    cmdsave.Enabled = False
    CMDUP.Enabled = True
```

```
    CMDDEL.Enabled = True

Else
    MsgBox "record not found", vbInformation
End If
End If
Else
Er:
    MsgBox "No Row Selected", vbInformation
End If
    CpyUpdate = TXTDEPT.Text
End Sub
```

RATE LIST FORM

```
Dim P_DEPT_OLD As String
Dim CpyUpdate As String


Private Sub cmbser_Click()
   If RS.State = 1 Then RS.Close
   RS.Open "SELECT * FROM SUB_SERVICE where service_NAME ='" & cmbser.Text & "'  order by SUB_SERVICE_NAME", con, adOpenKeyset, adLockOptimistic, adCmdText
   msg1.Rows = 1
   If RS.RecordCount >= 1 Then
   Combo1.Clear
   While RS.EOF = False
   msg1.AddItem RS(1) & Chr(9) & RS(2)
   Combo1.AddItem RS(1)
   RS.MoveNext
Wend
End If
   Lbltotal.Caption = RS.RecordCount
End Sub
```

```
Private Sub cmbser_KeyPress(KeyAscii As Integer)

    If (Command1.Caption = "New") Then
    End If
End Sub



Private Sub cmdcancel_Click()
    Call cmbser_Click
    TXTDEPT.Text = ""
    TXTDEPT.SetFocus
    cmbser.Visible = True
    Text1.Text = ""
    Text1.Visible = False
End Sub



Private Sub cmdClose_Click()
    Unload Me
End Sub



Private Sub cmddel_Click()
```

```
    If MsgBox("IT WILL ALSO DELETE ALL RATE LIST OF THIS SUB
SERVICE" & vbCrLf & "                    ARE YOU WANT TO DELETE.",
vbYesNo) = vbYes Then
        con.Execute "DELETE FROM SUB_SERVICE where
SUB_SERVICE_NAME = '" & Trim(UCase(TXTDEPT.Text)) & "'"
        Call cmbser_Click
        TXTDEPT.Text = ""
        TXTDEPT.SetFocus

       cmdsave.Enabled = True
        CMDDEL.Enabled = False
        MsgBox "Recored Deleted", vbInformation
    End If
End Sub




Private Sub cmdSave_Click()
    If TXTDEPT.Text = "" Then
    MsgBox "Please Enter RATE", vbInformation
    TXTDEPT.SetFocus
    Exit Sub
    End If
    sql = "INSERT INTO SUB_SERVICE (Measured_in) VALUES(" &
Val(TXTDEPT.Text) & ")"
    con.Execute sql
```

```
    Call cmbser_Click
    TXTDEPT.Text = ""
    Combo1.Text = ""
    TXTDEPT.SetFocus
    MsgBox "Recored Saved", vbInformation
End Sub




Private Sub CMDUP_Click()
    If (cmbser.Text = "") Then
    MsgBox "Please Select SERVICE Name", vbInformation
    cmbser.SetFocus
Exit Sub
End If
     If (TXTDEPT.Text = "") Then
    MsgBox "Please Enter SUB SERVICE Name", vbInformation
    TXTDEPT.SetFocus
Else
    con.Execute ("UPDATE SUB_SERVICE SET Measured_in = " &
Val(TXTDEPT.Text) & " where SUB_SERVICE_NAME = '" &
Combo1.Text & "'")
    cmbser_Click
    MsgBox "Record Updated Successfully", vbInformation
    TXTDEPT.SetFocus
    cmdsave.Enabled = True
```

```
    CMDDEL.Enabled = False
    TXTDEPT.Text = ""
    End If
End Sub




Private Sub Command1_Click()
If Command1.Caption = "New" Then

   Text1.Visible = True
    cmbser.Visible = False
    Text1.SetFocus
    Command1.Caption = "Save"
Else
    If Text1.Text = "" Then
    MsgBox "Please Enter SERVICE Name", vbInformation
    Text1.SetFocus
Exit Sub
End If
    sql = "INSERT INTO MAIN_SERVICE (SERVICE_NAME) VALUES('" &
UCase(Trim(Text1.Text)) & "')"
    con.Execute sql
    cmbser.Visible = True
    cmbser.AddItem UCase(Trim(Text1.Text))
    cmbser.Text = UCase(Trim(Text1.Text))
```

```
    Text1.Text = ""
    Text1.Visible = False
    cmbser_Click
    TXTDEPT.SetFocus
    Command1.Caption = "New"
End If

End Sub




Private Sub Form_Load()
    TXTDEPT.Text = ""
    cmdsave.Enabled = True
    CMDDEL.Enabled = False
    If RS.State = 1 Then RS.Close
    RS.Open "SELECT * FROM MAIN_SERVICE order by
SERVICE_NAME", con, adOpenKeyset, adLockOptimistic, adCmdText
    cmbser.Clear
    If RS.RecordCount >= 1 Then
    While RS.EOF = False
    cmbser.AddItem RS(0)
    RS.MoveNext
    Wend
    End If
    RS.Close
```

```
End Sub



Private Sub msg1_Click()
    On Error GoTo Er
    If Err.Description = "" Then
    If Administrator_Password = False Then
    MsgBox "You Have No Permission to Change Details.Please Logon the Administrator first.", vbInformation
    Exit Sub

Else
    If RS.State = 1 Then RS.Close
    RS.Open "SELECT * FROM SUB_SERVICE where SUB_SERVICE_NAME ='" & msg1.TextMatrix(msg1.RowSel, 0) & "'", con, adOpenKeyset, adLockOptimistic, adCmdText
    If RS.RecordCount >= 1 Then
    While RS.EOF = False
    Combo1.Text = RS(1)
    cmbser.Text = RS(0)
    RS.MoveNext
Wend

    cmdsave.Enabled = False
    CMDUP.Enabled = True
    CMDDEL.Enabled = True
```

```
Else
   MsgBox "record not found", vbInformation
End If
End If
Else
Er:
   MsgBox "No Row Selected", vbInformation
End If
   CpyUpdate = Combo1.Text
End Sub
```

PURCHASE FORM

```
Dim RS As New ADODB.Recordset




Private Sub cmbf_type_click()
    If RS.State = 1 Then RS.Close
    RS.Open "SELECT * FROM SUB_SERVICE where service_NAME ='" &
text1.Text & "' and sub_service_name='" & cmbf_type.Text & "'", con,
adOpenKeyset, adLockOptimistic, adCmdText

    If RS.RecordCount >= 1 Then
    txtAmount.Text = RS(2)
    End If
Text3.Text = ""
Text4.Text = ""
End Sub




Private Sub cmdClose_Click()
Unload Me
End Sub
```

```
Private Sub cmddel_Click()
 If MsgBox("IT WILL ALSO DELETE ALL METIRIAL OF THIS SERVICE" &
vbCrLf & "                    ARE YOU WANT TO DELETE.", vbYesNo) =
vbYes Then
    If RS.State = 1 Then RS.Close
        RS.Open "DELETE FROM PURCHASE where Purchase_Order_no =
" & Val(Text2.Text), con, adOpenKeyset, adLockOptimistic, adCmdText
      MsgBox "Recored Deleted", vbInformation
    End If
End Sub



Private Sub cmdfind_Click()
List1.Visible = True
End Sub



Private Sub cmdnew_Click()
Dim sql As String
     cmdsave.Enabled = True
     cmdupdate.Enabled = False
     cmdfind.Enabled = False
```

```
    cmdnew.Enabled = False

sql = "select max(purchase_order_no)from purchase"
If RS.State = 1 Then RS.Close
RS.Open sql, con, adOpenDynamic, adLockOptimistic

If IsNull(RS(0)) Then
    Text2.Text = "001"
Else
    Text2.Text = "00" & (RS(0) + 1)
End If
   txtPur_Or_n.SetFocus
   txtPur_Or_n.Text = ""
   txtAmount.Text = ""
   text1.Text = ""
   cmbf_type.Text = ""
   Combo1.Text = "nil"
   Combo2.Text = "nil"
   Command2.Enabled = True

End Sub

Private Sub cmdSave_Click()
Command1_Click
  Form_Load
   cmdnew.Enabled = True
   cmdsave.Enabled = False
```

```
    cmdfind.Enabled = True
End Sub



Private Sub cmdupdate_Click()
Dim sql As String
If RS.State = 1 Then RS.Close
RS.Open "select * from purchase where Purchase_Order_no=" &
Val(Text2.Text), con, adOpenKeyset, adLockOptimistic, adCmdText

    RS(0) = DTd_o_p.Value
    RS(2) = txtPur_Or_n.Text
    RS(3) = text1.Text
    RS(4) = cmbf_type.Text
    RS(5) = Combo1.Text
    RS(6) = Combo2.Text
    RS(7) = Val(txtAmount.Text)
   RS.Update
MsgBox ("record update"), vbInformation
End Sub



Private Sub Combo2_Click()
Select Case Combo2.Text
    Case "10 gm":
```

```
        txtAmount.Text = "40.00"
    Case "20 gm":
        txtAmount.Text = "90.00"


 Case "50 gm":
        txtAmount.Text = "150.00"
    Case "100 gm":
        txtAmount.Text = "320.00"
    Case "250 gm":
        txtAmount.Text = "500.00"
    Case "500 gm":
        txtAmount.Text = "700.00"
    Case "1000 gm":
        txtAmount.Text = "850.00"
    Case "MORE":
        txtAmount.Text = "950.00"

End Select
End Sub



Private Sub Command1_Click()
If DataEnvironment1.rsCommand5_Grouping.State = 1 Then
DataEnvironment1.rsCommand5_Grouping.Close
```

```
DataEnvironment1.rsCommand5_Grouping.Filter =
"PURCHASE_ORDER_NO = " & Val(Text2.Text)
DataReport5.Show
End Sub



Private Sub Command2_Click()
If cmbf_type.Text = "" Then
        MsgBox "Please enter Fuel Type", vbInformation
        cmbf_type.SetFocus
    Exit Sub
End If
If Trim(DTd_o_p) = "" Then
   DTd_o_p.SetFocus
   MsgBox "enter suitable date", vbInformation
   Exit Sub
   End If
   If Trim(txtPur_Or_n) = "" Then
   txtPur_Or_n.SetFocus
   MsgBox "enter suitable purchase order no", vbInformation
   Exit Sub
   End If
   If Trim(txtAmount) = "" Then
   txtAmount.SetFocus
   MsgBox "enter suitable amount", vbInformation
```

```
Exit Sub
End If
If Trim(text1) = "" Then
cmbb_name.SetFocus
MsgBox "enter RECIPIENT NAME", vbInformation
Exit Sub


End If
If Trim(Text2) = "" Then
txtDraft_che.SetFocus
MsgBox "enter number only", vbInformation
Exit Sub
End If
If Trim(cmbf_type.Text) = "" Then
txtQuantity.SetFocus
MsgBox "enter FROM", vbInformation
Exit Sub
End If
If Trim(Combo1.Text) = "" Then
txtQuantity.SetFocus
MsgBox "enter SENd TO", vbInformation
Exit Sub
End If
If Trim(Combo2.Text) = "" Then
txtQuantity.SetFocus
MsgBox "enter WAIT", vbInformation
```

```
    Exit Sub
    End If
    If RS.State = 1 Then RS.Close
    RS.Open "select * from purchase", con, adOpenKeyset,
adLockOptimistic, adCmdText
    RS.AddNew
     RS(0) = DTd_o_p.Value

   RS(1) = Val(Text2.Text)
     RS(2) = txtPur_Or_n.Text
     RS(3) = text1.Text
     RS(4) = cmbf_type.Text
     RS(5) = Val(txtAmount.Text)
     RS(6) = Val(Text4.Text)
     RS(7) = Val(Text3.Text)
    RS.Update
    txtPur_Or_n.Enabled = False
    MsgBox "record ADDED", vbInformation
    Adodc1.RecordSource = "select * from purchase where
Purchase_Order_no=" & Val(Text2.Text)
    Adodc1.Refresh
End Sub



Private Sub Form_Load()
Dim sql As String
```

```
centerform Me
DTd_o_p.Value = Date
sql = "select * from purchase"
If RS.State = 1 Then
RS.Close
Set RS = Nothing
End If
RS.Open sql, con, adOpenKeyset, adLockPessimistic, adCmdText

If RS.RecordCount = 0 Then
   cmdsave.Enabled = False
   cmdupdate.Enabled = False
   cmdfind.Enabled = False
End If
List1.Visible = False
If RS.State = 1 Then RS.Close
RS.Open "select purchase_order_no from  purchase GROUP BY
PURCHASE_ORDER_NO", con, adOpenKeyset, adLockOptimistic
List1.Clear
While Not RS.EOF = True
List1.AddItem RS(0)
RS.MoveNext
Wend

   If RS.State = 1 Then RS.Close
   RS.Open "SELECT * FROM MAIN_SERVICE order by
SERVICE_NAME", con, adOpenKeyset, adLockOptimistic, adCmdText
```

```
    text1.Clear
    If RS.RecordCount >= 1 Then
    While RS.EOF = False
    text1.AddItem RS(0)
    RS.MoveNext
    Wend
    End If
    RS.Close


cmbf_type.Clear
DTd_o_p = Date
txtAmount = ""
cmbb_name = ""
txtDraft_che = ""
txtQuantity = ""
cmdfind.Enabled = True
Text2.Text = ""
  text1.Text = ""
  txtPur_Or_n.Text = ""
  cmbf_type.Text = ""
  txtAmount.Text = ""
  Text3.Text = ""
   Text4.Text = ""
  txtPur_Or_n.Enabled = True
  text1.Enabled = True
  Command2.Enabled = False
```

```
  Adodc1.RecordSource = "select * from purchase where
Purchase_Order_no=0"
  Adodc1.Refresh
End Sub

Private Sub List1_Click()
Dim sql As String
txtPur_Or_n = List1.Text
If RS.State = 1 Then RS.Close

sql = "select * from purchase where purchase_order_no =" &
Val(List1.Text)
RS.Open sql, con, adOpenKeyset, adLockOptimistic
If RS.RecordCount > 0 Then
   DTd_o_p.Value = RS(0)
   Text2.Text = RS(1)
   txtPur_Or_n.Text = RS(2)
   text1.Text = RS(3)
   cmbf_type.Text = RS(4)
   Combo1.Text = RS(7)
   Combo2.Text = RS(6)
   txtAmount.Text = RS(5)
End If

cmdupdate.Enabled = True
Adodc1.RecordSource = "select * from purchase where
purchase_order_no =" & Val(List1.Text)
```

```
Adodc1.Refresh
End Sub



Private Sub text1_Click()
   If RS.State = 1 Then RS.Close
   RS.Open "SELECT * FROM SUB_SERVICE where service_NAME ='" &
text1.Text & "'  order by SUB_SERVICE_NAME", con, adOpenKeyset,
adLockOptimistic, adCmdText

   If RS.RecordCount >= 1 Then
   cmbf_type.Clear
   While RS.EOF = False
     cmbf_type.AddItem RS(1)
   RS.MoveNext
   Wend
   End If
End Sub



Private Sub Text4_KeyPress(KeyAscii As Integer)
validno KeyAscii
End Sub
```

```
Private Sub Text4_KeyUp(KeyCode As Integer, Shift As Integer)
Text3.Text = Val(txtAmount.Text) * Val(Text4.Text)
End Sub


Private Sub txtAmount_KeyPress(KeyAscii As Integer)
Call validno(KeyAscii)
End Sub


Private Sub txtDraft_che_KeyPress(KeyAscii As Integer)
Call validno(KeyAscii)
End Sub


Private Sub txtQuantity_KeyPress(KeyAscii As Integer)
Call validno(KeyAscii)
End Sub
```

RATE REPORT FORM

```
Private Sub Command1_Click()

DataReport4.Show

End Sub
```

MODULE FORM

```
Public con As New ADODB.Connection
Public RS As New Recordset
Public Administrator_Password As Boolean



Private Sub main()

Administrator_Password = True
    If App.PrevInstance = True Then
        MsgBox "project allready opened", vbInformation
        End
    End If
        Call setconnection
        user_login.Show
End Sub



Private Sub setconnection()
con.CursorLocation = adUseClient
```

```
con.Open "Provider=MSDAORA.1;Password=SYSTEM;User
ID=system;Persist Security Info=True"
If DataEnvironment1.Connection1.State = 1 Then
DataEnvironment1.Connection1.Close

Exit Sub
errhand:
MsgBox "connection error", vbExclamation + vbOKOnly, "error"
con.Open "provider=microsoft.jet.oledb.4.0;data source= Data.mdb"
End
End Sub



Public Function centerfrm(frm As Frame)
frm.Left = (Screen.Width - frm.Width) / 2

frm.Top = (Screen.Height - frm.Height) / 2

End Function
```

```
Public Function centerform(frm As Form)
frm.Left = (Screen.Width - frm.Width) / 2

frm.Top = (Screen.Height - frm.Height) / 2

End Function



Public Function validno(KeyAscii As Integer)
Dim str As String
str = "0123456789."
If KeyAscii > 25 Then
  If InStr(str, Chr(KeyAscii)) = 0 Then
    MsgBox "enter number only", vbInformation
    KeyAscii = 0
  End If
End If

End Function
```

```
Public Function validchar(KeyAscii As Integer)
Dim str As String
str = "0123456789.!@#$%^&*()"
If KeyAscii > 25 Then
  If InStr(str, Chr(KeyAscii)) <> 0 Then
    MsgBox "enter character only", vbInformation
    KeyAscii = 0
  End If
End If

End Function
```

SUPPLIER SELL FROM

```
Dim RS As New ADODB.Recordset




Private Sub cmbf_type_click()
   If RS.State = 1 Then RS.Close
   RS.Open "SELECT * FROM SUB_SERVICE where service_NAME ='" &
text1.Text & "' and sub_service_name='" & cmbf_type.Text & "'", con,
adOpenKeyset, adLockOptimistic, adCmdText

   If RS.RecordCount >= 1 Then
   txtAmount.Text = RS(2)
   End If
Text3.Text = ""
Text4.Text = ""
End Sub




Private Sub cmdClose_Click()
Unload Me
End Sub
```

```
Private Sub cmddel_Click()
 If MsgBox("IT WILL ALSO DELETE ALL METIRIAL OF THIS SERVICE" &
vbCrLf & "                    ARE YOU WANT TO DELETE.", vbYesNo) =
vbYes Then
    If RS.State = 1 Then RS.Close
       RS.Open "DELETE FROM PURCHASE where Purchase_Order_no =
" & Val(Text2.Text), con, adOpenKeyset, adLockOptimistic, adCmdText
     MsgBox "Recored Deleted", vbInformation
   End If
End Sub



Private Sub cmdfind_Click()
List1.Visible = True
End Sub



Private Sub cmdnew_Click()
Dim sql As String
    cmdsave.Enabled = True
    cmdupdate.Enabled = False
    cmdfind.Enabled = False
```

```
    cmdnew.Enabled = False

sql = "select max(purchase_order_no)from purchase"
If RS.State = 1 Then RS.Close
RS.Open sql, con, adOpenDynamic, adLockOptimistic

If IsNull(RS(0)) Then
    Text2.Text = "001"
Else
    Text2.Text = "00" & (RS(0) + 1)
End If
   txtPur_Or_n.SetFocus
   txtPur_Or_n.Text = ""
   txtAmount.Text = ""
   text1.Text = ""
   cmbf_type.Text = ""
   Combo1.Text = "nil"
   Combo2.Text = "nil"
   Command2.Enabled = True

End Sub

Private Sub cmdSave_Click()
Command1_Click
  Form_Load
   cmdnew.Enabled = True
   cmdsave.Enabled = False
```

```
    cmdfind.Enabled = True
End Sub



Private Sub cmdupdate_Click()
Dim sql As String
If RS.State = 1 Then RS.Close
RS.Open "select * from purchase where Purchase_Order_no=" &
Val(Text2.Text), con, adOpenKeyset, adLockOptimistic, adCmdText

    RS(0) = DTd_o_p.Value
    RS(2) = txtPur_Or_n.Text
    RS(3) = text1.Text
    RS(4) = cmbf_type.Text
    RS(5) = Combo1.Text
    RS(6) = Combo2.Text
    RS(7) = Val(txtAmount.Text)
   RS.Update
MsgBox ("record update"), vbInformation
End Sub



Private Sub Combo2_Click()
Select Case Combo2.Text
    Case "10 gm":
```

```
        txtAmount.Text = "40.00"
    Case "20 gm":
        txtAmount.Text = "90.00"


 Case "50 gm":
        txtAmount.Text = "150.00"
    Case "100 gm":
        txtAmount.Text = "320.00"
    Case "250 gm":
        txtAmount.Text = "500.00"
    Case "500 gm":
        txtAmount.Text = "700.00"
    Case "1000 gm":
        txtAmount.Text = "850.00"
    Case "MORE":
        txtAmount.Text = "950.00"

End Select
End Sub



Private Sub Command1_Click()
If DataEnvironment1.rsCommand5_Grouping.State = 1 Then
DataEnvironment1.rsCommand5_Grouping.Close
```

```
DataEnvironment1.rsCommand5_Grouping.Filter =
"PURCHASE_ORDER_NO = " & Val(Text2.Text)
DataReport5.Show
End Sub



Private Sub Command2_Click()
If cmbf_type.Text = "" Then
        MsgBox "Please enter Fuel Type", vbInformation
        cmbf_type.SetFocus
    Exit Sub
End If
If Trim(DTd_o_p) = "" Then
   DTd_o_p.SetFocus
   MsgBox "enter suitable date", vbInformation
   Exit Sub
   End If
   If Trim(txtPur_Or_n) = "" Then
   txtPur_Or_n.SetFocus
   MsgBox "enter suitable purchase order no", vbInformation
   Exit Sub
   End If
   If Trim(txtAmount) = "" Then
   txtAmount.SetFocus
   MsgBox "enter suitable amount", vbInformation
```

```
Exit Sub
End If
If Trim(text1) = "" Then
cmbb_name.SetFocus
MsgBox "enter RECIPIENT NAME", vbInformation
Exit Sub


End If
If Trim(Text2) = "" Then
txtDraft_che.SetFocus
MsgBox "enter number only", vbInformation
Exit Sub
End If
If Trim(cmbf_type.Text) = "" Then
txtQuantity.SetFocus
MsgBox "enter FROM", vbInformation
Exit Sub
End If
If Trim(Combo1.Text) = "" Then
txtQuantity.SetFocus
MsgBox "enter SENd TO", vbInformation
Exit Sub
End If
If Trim(Combo2.Text) = "" Then
txtQuantity.SetFocus
MsgBox "enter WAIT", vbInformation
```

```
    Exit Sub
    End If
    If RS.State = 1 Then RS.Close
    RS.Open "select * from purchase", con, adOpenKeyset,
adLockOptimistic, adCmdText
    RS.AddNew
     RS(0) = DTd_o_p.Value

   RS(1) = Val(Text2.Text)
     RS(2) = txtPur_Or_n.Text
     RS(3) = text1.Text
     RS(4) = cmbf_type.Text
     RS(5) = Val(txtAmount.Text)
     RS(6) = Val(Text4.Text)
     RS(7) = Val(Text3.Text)
    RS.Update
    txtPur_Or_n.Enabled = False
    MsgBox "record ADDED", vbInformation
    Adodc1.RecordSource = "select * from purchase where
Purchase_Order_no=" & Val(Text2.Text)
    Adodc1.Refresh
End Sub



Private Sub Form_Load()
Dim sql As String
```

```
centerform Me
DTd_o_p.Value = Date
sql = "select * from purchase"
If RS.State = 1 Then
RS.Close
Set RS = Nothing
End If
RS.Open sql, con, adOpenKeyset, adLockPessimistic, adCmdText

If RS.RecordCount = 0 Then
   cmdsave.Enabled = False
   cmdupdate.Enabled = False
   cmdfind.Enabled = False
End If
List1.Visible = False
If RS.State = 1 Then RS.Close
RS.Open "select purchase_order_no from  purchase GROUP BY
PURCHASE_ORDER_NO", con, adOpenKeyset, adLockOptimistic
List1.Clear
While Not RS.EOF = True
List1.AddItem RS(0)
RS.MoveNext
Wend

   If RS.State = 1 Then RS.Close
   RS.Open "SELECT * FROM MAIN_SERVICE order by
SERVICE_NAME", con, adOpenKeyset, adLockOptimistic, adCmdText
```

```
    text1.Clear
    If RS.RecordCount >= 1 Then
    While RS.EOF = False
    text1.AddItem RS(0)
    RS.MoveNext
    Wend
    End If
    RS.Close


cmbf_type.Clear
DTd_o_p = Date
txtAmount = ""
cmbb_name = ""
txtDraft_che = ""
txtQuantity = ""
cmdfind.Enabled = True
Text2.Text = ""
  text1.Text = ""
  txtPur_Or_n.Text = ""
  cmbf_type.Text = ""
  txtAmount.Text = ""
  Text3.Text = ""
   Text4.Text = ""
  txtPur_Or_n.Enabled = True
  text1.Enabled = True
  Command2.Enabled = False
```

```
  Adodc1.RecordSource = "select * from purchase where
Purchase_Order_no=0"
  Adodc1.Refresh
End Sub

Private Sub List1_Click()
Dim sql As String
txtPur_Or_n = List1.Text
If RS.State = 1 Then RS.Close

sql = "select * from purchase where purchase_order_no =" &
Val(List1.Text)
RS.Open sql, con, adOpenKeyset, adLockOptimistic
If RS.RecordCount > 0 Then
   DTd_o_p.Value = RS(0)
   Text2.Text = RS(1)
   txtPur_Or_n.Text = RS(2)
   text1.Text = RS(3)
   cmbf_type.Text = RS(4)
   Combo1.Text = RS(7)
   Combo2.Text = RS(6)
   txtAmount.Text = RS(5)
End If

cmdupdate.Enabled = True
Adodc1.RecordSource = "select * from purchase where
purchase_order_no =" & Val(List1.Text)
```

```
Adodc1.Refresh
End Sub



Private Sub text1_Click()
   If RS.State = 1 Then RS.Close
   RS.Open "SELECT * FROM SUB_SERVICE where service_NAME ='" &
text1.Text & "'  order by SUB_SERVICE_NAME", con, adOpenKeyset,
adLockOptimistic, adCmdText

   If RS.RecordCount >= 1 Then
   cmbf_type.Clear
   While RS.EOF = False
     cmbf_type.AddItem RS(1)
   RS.MoveNext
   Wend
   End If
End Sub



Private Sub Text4_KeyPress(KeyAscii As Integer)
validno KeyAscii
End Sub
```

```
Private Sub Text4_KeyUp(KeyCode As Integer, Shift As Integer)
Text3.Text = Val(txtAmount.Text) * Val(Text4.Text)
End Sub


Private Sub txtAmount_KeyPress(KeyAscii As Integer)
Call validno(KeyAscii)
End Sub


Private Sub txtDraft_che_KeyPress(KeyAscii As Integer)
Call validno(KeyAscii)
End Sub


Private Sub txtQuantity_KeyPress(KeyAscii As Integer)
Call validno(KeyAscii)
End Sub
```

STOCK DETAILSH FORM

```
Dim RS As New ADODB.Recordset




Private Sub cmbf_type_click()
   If RS.State = 1 Then RS.Close
   RS.Open "SELECT * FROM SUB_SERVICE where service_NAME ='" &
text1.Text & "' and sub_service_name='" & cmbf_type.Text & "'", con,
adOpenKeyset, adLockOptimistic, adCmdText

   If RS.RecordCount >= 1 Then
   txtAmount.Text = RS(2)
   End If
Text3.Text = ""
Text4.Text = ""
End Sub




Private Sub cmdClose_Click()
Unload Me
End Sub
```

```
Private Sub cmddel_Click()
 If MsgBox("IT WILL ALSO DELETE ALL METIRIAL OF THIS SERVICE" &
vbCrLf & "                    ARE YOU WANT TO DELETE.", vbYesNo) =
vbYes Then
    If RS.State = 1 Then RS.Close
        RS.Open "DELETE FROM PURCHASE where Purchase_Order_no =
" & Val(Text2.Text), con, adOpenKeyset, adLockOptimistic, adCmdText
      MsgBox "Recored Deleted", vbInformation
    End If
End Sub



Private Sub cmdfind_Click()
List1.Visible = True
End Sub



Private Sub cmdnew_Click()
Dim sql As String
     cmdsave.Enabled = True
     cmdupdate.Enabled = False
     cmdfind.Enabled = False
```

```
    cmdnew.Enabled = False

sql = "select max(purchase_order_no)from purchase"
If RS.State = 1 Then RS.Close
RS.Open sql, con, adOpenDynamic, adLockOptimistic

If IsNull(RS(0)) Then
    Text2.Text = "001"
Else
    Text2.Text = "00" & (RS(0) + 1)
End If
   txtPur_Or_n.SetFocus
   txtPur_Or_n.Text = ""
   txtAmount.Text = ""
   text1.Text = ""
   cmbf_type.Text = ""
   Combo1.Text = "nil"
   Combo2.Text = "nil"
   Command2.Enabled = True

End Sub

Private Sub cmdSave_Click()
Command1_Click
  Form_Load
   cmdnew.Enabled = True
   cmdsave.Enabled = False
```

```
    cmdfind.Enabled = True
End Sub



Private Sub cmdupdate_Click()
Dim sql As String
If RS.State = 1 Then RS.Close
RS.Open "select * from purchase where Purchase_Order_no=" &
Val(Text2.Text), con, adOpenKeyset, adLockOptimistic, adCmdText

    RS(0) = DTd_o_p.Value
    RS(2) = txtPur_Or_n.Text
    RS(3) = text1.Text
    RS(4) = cmbf_type.Text
    RS(5) = Combo1.Text
    RS(6) = Combo2.Text
    RS(7) = Val(txtAmount.Text)
   RS.Update
MsgBox ("record update"), vbInformation
End Sub



Private Sub Combo2_Click()
Select Case Combo2.Text
    Case "10 gm":
```

```
        txtAmount.Text = "40.00"
    Case "20 gm":
        txtAmount.Text = "90.00"


 Case "50 gm":
        txtAmount.Text = "150.00"
    Case "100 gm":
        txtAmount.Text = "320.00"
    Case "250 gm":
        txtAmount.Text = "500.00"
    Case "500 gm":
        txtAmount.Text = "700.00"
    Case "1000 gm":
        txtAmount.Text = "850.00"
    Case "MORE":
        txtAmount.Text = "950.00"

End Select
End Sub



Private Sub Command1_Click()
If DataEnvironment1.rsCommand5_Grouping.State = 1 Then
DataEnvironment1.rsCommand5_Grouping.Close
```

```
DataEnvironment1.rsCommand5_Grouping.Filter =
"PURCHASE_ORDER_NO = " & Val(Text2.Text)
DataReport5.Show
End Sub



Private Sub Command2_Click()
If cmbf_type.Text = "" Then
        MsgBox "Please enter Fuel Type", vbInformation
        cmbf_type.SetFocus
    Exit Sub
End If
If Trim(DTd_o_p) = "" Then
   DTd_o_p.SetFocus
   MsgBox "enter suitable date", vbInformation
   Exit Sub
   End If
   If Trim(txtPur_Or_n) = "" Then
   txtPur_Or_n.SetFocus
   MsgBox "enter suitable purchase order no", vbInformation
   Exit Sub
   End If
   If Trim(txtAmount) = "" Then
   txtAmount.SetFocus
   MsgBox "enter suitable amount", vbInformation
```

```
Exit Sub
End If
If Trim(text1) = "" Then
cmbb_name.SetFocus
MsgBox "enter RECIPIENT NAME", vbInformation
Exit Sub


End If
If Trim(Text2) = "" Then
txtDraft_che.SetFocus
MsgBox "enter number only", vbInformation
Exit Sub
End If
If Trim(cmbf_type.Text) = "" Then
txtQuantity.SetFocus
MsgBox "enter FROM", vbInformation
Exit Sub
End If
If Trim(Combo1.Text) = "" Then
txtQuantity.SetFocus
MsgBox "enter SENd TO", vbInformation
Exit Sub
End If
If Trim(Combo2.Text) = "" Then
txtQuantity.SetFocus
MsgBox "enter WAIT", vbInformation
```

```
    Exit Sub
    End If
    If RS.State = 1 Then RS.Close
    RS.Open "select * from purchase", con, adOpenKeyset,
adLockOptimistic, adCmdText
    RS.AddNew
     RS(0) = DTd_o_p.Value

   RS(1) = Val(Text2.Text)
     RS(2) = txtPur_Or_n.Text
     RS(3) = text1.Text
     RS(4) = cmbf_type.Text
     RS(5) = Val(txtAmount.Text)
     RS(6) = Val(Text4.Text)
     RS(7) = Val(Text3.Text)
    RS.Update
    txtPur_Or_n.Enabled = False
    MsgBox "record ADDED", vbInformation
    Adodc1.RecordSource = "select * from purchase where
Purchase_Order_no=" & Val(Text2.Text)
    Adodc1.Refresh
End Sub



Private Sub Form_Load()
Dim sql As String
```

```
centerform Me
DTd_o_p.Value = Date
sql = "select * from purchase"
If RS.State = 1 Then
RS.Close
Set RS = Nothing
End If
RS.Open sql, con, adOpenKeyset, adLockPessimistic, adCmdText

If RS.RecordCount = 0 Then
   cmdsave.Enabled = False
   cmdupdate.Enabled = False
   cmdfind.Enabled = False
End If
List1.Visible = False
If RS.State = 1 Then RS.Close
RS.Open "select purchase_order_no from  purchase GROUP BY
PURCHASE_ORDER_NO", con, adOpenKeyset, adLockOptimistic
List1.Clear
While Not RS.EOF = True
List1.AddItem RS(0)
RS.MoveNext
Wend

   If RS.State = 1 Then RS.Close
   RS.Open "SELECT * FROM MAIN_SERVICE order by
SERVICE_NAME", con, adOpenKeyset, adLockOptimistic, adCmdText
```

```
    text1.Clear
    If RS.RecordCount >= 1 Then
    While RS.EOF = False
    text1.AddItem RS(0)
    RS.MoveNext
    Wend
    End If
    RS.Close


cmbf_type.Clear
DTd_o_p = Date
txtAmount = ""
cmbb_name = ""
txtDraft_che = ""
txtQuantity = ""
cmdfind.Enabled = True
Text2.Text = ""
  text1.Text = ""
  txtPur_Or_n.Text = ""
  cmbf_type.Text = ""
  txtAmount.Text = ""
  Text3.Text = ""
   Text4.Text = ""
  txtPur_Or_n.Enabled = True
  text1.Enabled = True
  Command2.Enabled = False
```

```
  Adodc1.RecordSource = "select * from purchase where
Purchase_Order_no=0"
  Adodc1.Refresh
End Sub

Private Sub List1_Click()
Dim sql As String
txtPur_Or_n = List1.Text
If RS.State = 1 Then RS.Close

sql = "select * from purchase where purchase_order_no =" &
Val(List1.Text)
RS.Open sql, con, adOpenKeyset, adLockOptimistic
If RS.RecordCount > 0 Then
   DTd_o_p.Value = RS(0)
   Text2.Text = RS(1)
   txtPur_Or_n.Text = RS(2)
   text1.Text = RS(3)
   cmbf_type.Text = RS(4)
   Combo1.Text = RS(7)
   Combo2.Text = RS(6)
   txtAmount.Text = RS(5)
End If

cmdupdate.Enabled = True
Adodc1.RecordSource = "select * from purchase where
purchase_order_no =" & Val(List1.Text)
```

```
Adodc1.Refresh
End Sub



Private Sub text1_Click()
   If RS.State = 1 Then RS.Close
   RS.Open "SELECT * FROM SUB_SERVICE where service_NAME ='" &
text1.Text & "'  order by SUB_SERVICE_NAME", con, adOpenKeyset,
adLockOptimistic, adCmdText

   If RS.RecordCount >= 1 Then
   cmbf_type.Clear
   While RS.EOF = False
     cmbf_type.AddItem RS(1)
   RS.MoveNext
   Wend
   End If
End Sub



Private Sub Text4_KeyPress(KeyAscii As Integer)
validno KeyAscii
End Sub
```

```
Private Sub Text4_KeyUp(KeyCode As Integer, Shift As Integer)
Text3.Text = Val(txtAmount.Text) * Val(Text4.Text)
End Sub


Private Sub txtAmount_KeyPress(KeyAscii As Integer)
Call validno(KeyAscii)
End Sub


Private Sub txtDraft_che_KeyPress(KeyAscii As Integer)
Call validno(KeyAscii)
End Sub


Private Sub txtQuantity_KeyPress(KeyAscii As Integer)
Call validno(KeyAscii)
End Sub
```

CONCLUSION

Now days computerizations of existing manual system is going on a large scale because of the versatility, speed, accuracy and diligence it offers to its users. Computers provide practical means to organize things systematically and economically in the organization the use of computers for managing transactions; information processing and preparation of reports can prove to be a blessing.

The project is discussion is an attempt to attain all the above said objectives. Its development was mean to replace the manual system and to achieve the goal to maximum accuracy and most efficiently. But like every other system might process faults to its credits and has its own limitation. Neglecting these few negations, the project can be called a stepping stone to automate processes in organizations.

REFRENCES

References are always needed for the development of any System / Software development. Through the system development life cycle, I refer the following books and manuals for related sources.

Book Name	Author Name
The Complete Reference Visual Basic 6 McGraw Hill	**Tata**
Informatics Practices	**Sumita Arora**
Visual Programming Black Book	**Steven Holzner**