**INTRODUCTION**

**1.INTRODUCTION**

**1.1 PROJECT OVERVIEW**

In earlier days our company used traditional book keeping system for keeping all the details regarding to that company. It has many demerits .It is more time consuming, need more space and there is a chance to damage also. It is very hard to find a record about a particular transaction. And more tedious manual processing is required to get summary. The hardest work is payments made step by step. It requires extra records to maintain these transaction details.

The new system should be fast and overcome the limitations of the existing system. The system should secure, faster, error free and interactive than the existing system and provide better option for data reporting and data comparisons.

Our software is a desktop based system and run on windows platform. This software provides easy and efficient user interface. So that it is easy to understand and easy to use. It reduces manage mental difficulties of the company. This project manages the purchase, order, sales and payment details of the company in an efficient manner. Here we want to communicate with number of suppliers, who distributes the raw materials for the production and customers, who buy our product. The raw materials are three types. They are LD, HM, and LLD. The system stores the raw material details to manipulate the report about current stock. We take order from the customers. While taking orders we have to store the quality, height, width and color etc about the polythene bag. After the sale we collect money from the customers by cheque or by cash. It provides reports about purchase, order and sales.

The software system has administrator. The person who has the central control of the software. He can purchase material from suppliers, take orders from customers and sale products.

**MODULES**

1. Administration

2. Purchase Material

3. Order Polythene Bags

4. Sales and Billing

5. Payments

**MODULE DISCRIPTION**

**1. Administration**

The software system used by the company owner himself who has the central control over the software system. He is the administrator of the software system. He can login to the software using secret username and password. This makes the software system more secure. Only the administrator can make changes into transaction details. He can purchase raw materials from suppliers; take orders from customers and sale products. All these tasks are recorded only by the administrator.

**2**. **Purchase Material**

The company purchase materials from suppliers. So that system stores the supplier details. We buy raw materials from these suppliers. The raw materials are mainly three types. They are HM, LD, and LLD. The system stores purchase details, items to be purchased with company names. It also maintains records about the cash transactions. It generates the purchase and cash reports based on date criteria.

**3. Order Polythene Bags**

The system stores details about the customers with tin number. The company takes orders from customers. The software system stores the details of the order and items. The items have set of properties. They are height, width, gage and color. It provides order report based on date criteria. It mainly used to get information about the orders from the customers.

**4. Sales and Billing**

The company makes products based on the customer’s order. This module keeps sales details along with company name. It also keeps records about sales cash details. Company sales products to the registered customers and also to the non-registered users (Users without TIN Number).It also store the items to be sold. It generates reports based on date criteria.

**5. Payments**

This module mainly deals with purchase and sales payment and settlement details. It stores the purchase and sales cash details. The main aim is to get information about unpaid bills and generate reports of cash and settlement details.

**1.2 ORGANIZATION PROFILE**

The name of the firm is “Cherukayil Polymers”. It established in 1995. It is situated in Arimpur. The main materials used for the production are High-density polyethylene (HDPE),

Medium-density polyethylene (MDPE), linear low-density polyethylene (LLDPE),

Low-density polyethylene (LDPE).

They can provide a huge selection of bespoke and stock products including opaque mailers, cash bags, metallic envelopes and many environmentally friendly options. The main products are given below

*Stock Items*: A wide range of self-sealing envelopes from C6 to C3 size in either opaque or clear polythene. Suitable for mailing brochures, magazines, clothes and other mail order items.

*Sequential Numbering/Cash Bags*: Polythene bags with sequential numbering for security -ideal for cash, vouchers, and coinage. Bags can be overprinted and manufactured in clear or opaque polythene.

*Grip-Seal Bags*: Manufactured in 50 or 75 micron polythene these bags are ideal for craft materials, components, jewellery.

*100% Recycled Grey Mailing Envelopes*: Made from 100% recycled 60 micron grey polythene. The envelopes are waterproof, strong and tough, suitable for most mail order items.

*Carrier bag*: Bespoke Carrier bags available in clear or colored film and can be printed in up to 8 colors. Please call our team for a quotation.

They take environmental responsibilities very seriously. They offer an extensive range of products that are kinder to our environment including bio-degradable, compostable and recycled materials.

Polythene and polypropylene are fully recyclable whether they have been printed on or not. The printed or colored films are dyed in the recycling process and are used for products like black bin liners. The clear waste is made back into clear film (although the end product is slightly cloudier than virgin film) – They buy this film back to make stock and bespoke products.

In addition to the products they offer, the company also does everything it can to limit environmental impacts. This includes recycling all process waste and all possible office waste including print cartridges and paper. They have also had an energy reduction programe to reduce our carbon footprint and have succeeded in reducing our power usage by 15%.

**SYSTEM ANALYSIS**

**2.SYSTEM ANALYSIS**

**2.1 EXISTING SYSTEM**

Existing system is a traditional book keeping system. It is very difficult to process and maintain. Searching for a particular purchase detail, sales detail and order detail are tedious process. In earlier days our company used traditional book keeping system for keeping all the details regarding to that company. It has many demerits .It is more time consuming, need more space and there is a chance to damage also. It is very hard to find a record about a particular transaction. And more tedious manual processing is required to get summary. The hardest work is payments made step by step. It requires extra records to maintain these transaction details.

**DRAWBACKS OF EXISTING SYSTEM:**

* Most of the processes are manually operated; hence there is a chance of erroneous data processing.
* Poor reliability of finally submitted data.
* Large time consumption
* Manual process is very tedious and highly complex.
* Not flexible & not user friendly.
* Requires lots of human efforts and is less efficient.
* Record keeping is very difficult.
* Manual records are very difficult to be maintained safe
* Business can see itself in fines and penalties if records are lost
* Manual records are easier to be falsified, modified, altered or vanished, as compared   
  to computerized records which become very safe when using passwords, firewalls,   
  and back-ups.

**2.2 PROPOSED SYSTEM**

The Proposed system is a computerized system which is maintained at Centralized databases. And maintaining all the records in database which makes it very easy to access and retrieve data from the database. If they want any record they can easily search all the records. The reporting facility makes system more useful.

**MERITS OF PROPOSED SYSTEM:**

* The managing tasks of administrator can be made easy.
* Less expensive and without any complications.
* This is a less time consuming system with high efficiency and great reliability.
* It can manage large amount of data, keep their information accurately and update the performance up-to-date.
* There is no chance of losing the data and security features are provided (By username and password).
* Provide good interaction with the user.
* Reduces paper works and brain works.
* The system should be faster.
* Provide efficient performance and high security.
* Ease of use and less time consuming.
* Reduces money and effort.

**2.3 FEASIBILITY STUDY**

A **feasibility study** is an evaluation of a proposal designed to determine the difficulty in carrying out a designated task. Generally, a feasibility study precedes technical development and project implementation. In other words, a feasibility study is an evaluation or analysis of the potential impact of a proposed project.

**FEASIBILITY ANALYSIS**

Feasibility is a practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software or not. Such information as resource availability, cost estimate for software development, benefits of the software to organization, and cost to be incurred on its maintenance are considered. The objective of the feasibility study is to establish the reasons for developing software that is acceptable to users, adaptable to change, and comfortable to accomplished standards.

A feasibility analysis is performed to choose the system that meets the performance requirement at the least cost .For this software performance evaluation the requirements must be achieved within low cost. So the detailed study is necessary. The most essential tasks performed by this feasibility analyses are:

**Technology and system feasibility**

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not. Technological feasibility is carried out to determine whether the company has the capability, in terms of software, hardware, personnel and expertise, to handle the completion of the project.

**Economic feasibility**

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

Cost-based study: It is important to identify cost and benefit factors, which can be categorized as follows: 1. Development costs; and 2. Operating costs. This is an analysis of the costs to be incurred in the system and the benefits derivable out of the system. Time-based study: This is an analysis of the time required to achieve a return on investments. The future value of a project is also a factor.

**Legal feasibility**

Determines whether the proposed system conflicts with legal requirements, e.g. a data processing system must comply with the local Data Protection Acts.

**Operational feasibility**

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Proposed projects are beneficial only if they can be turned into information system that will meet the organization’s operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to implementation?

The purpose of the operational feasibility study is to determine whether the new system will be used if it is developed and implemented. And whether there will be resistance from users that will undermine the possible application benefits.

**SYSTEM REQUIREMENTS**

**3.SYSTEM REQUIREMENTS**

The system environments with regard to an application project are the environments which shaped the flesh and bones of the system from the concepts and ideas to a real working model. It is so crucial in selecting the various environments for the system development since they plays a vital role in determining the properties and behavior of the system. The overall project development is associated with the system environment selected. So the system environment should be selected such that the project development becomes smooth and easy. So a detailed study should be conducted and proper system environment should be selected. Basically the system environment is divided into Software Configuration and Hardware Configuration

**3.1 .Software Requirements**

Platform: Windows® 2000, Windows XP, Windows Server 2003, or Windows NT 4.0x, Windows Vista, Windows 7, Windows 8.

Front End: C#

Backend: SqlServer 2005

**3.2. Hardware Requirements**

System: IBM Compatible PC

Processor: Intel Pentium 4

Speed: 2.4GHz

Memory: 512 Mb RAM

Hard Disc Drive: 80GB HDD

Floppy Disk Drive: 1.44MB

Keyboard: 104Keys

Monitor: 15inch, SVGA Digital color monitor

CD-ROM: 52X

Mouse: OPTICAL

**SYSTEM DESIGN**

**4.SYSTEM DESIGN**

The design phase focuses on the detailed implementation of the system. Emphasis is on translating performance specification into design specifications. The design phase is a transition from a user-oriented document (system proposal) into a document oriented programmers’ or data base personnel. System design goes through two phase of development: logical and physical design.

**4.1 DATA FLOW DIAGRAM**

Data flow diagram is a structured analysis design tool that can be used for flow charting in place of or in association with information oriented and process oriented system flow charts. A DFD is a network that describes the flow of data and the process that change, or transformed data throughout a system. This network is constructed by using a set of symbols that do not imply a physical implementation.

The four basic symbols used to construct DFD’s are shown below:

ENTTY.jpg

A double square represents a data source or destination.

ARRW.jpg

A directed line represents the flow of data that is a data stream.

PRCS.jpg

An enclosed figure represents a process that transforms data stream.

DS.jpg

An open ended rectangle represents data storage.

**Steps to Construct Data Flow Diagrams**

Four steps are commonly used to construct a DFD

Process should be named and numbered for easy reference. Each name should be representative of the process.

The destination of flow is from top to bottom and from left to right.

When a process is exploded in to lower level details they are numbered.

The names of data stores, sources and destinations are written in capital letters.

**Rules for constructing a Data Flow Diagram**

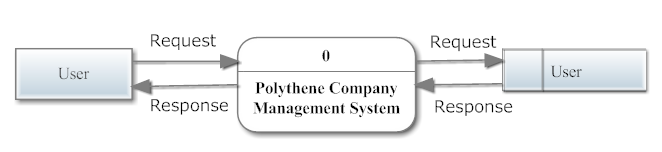
Arrows should not cross each other.

Squares, circles and files must bear names.

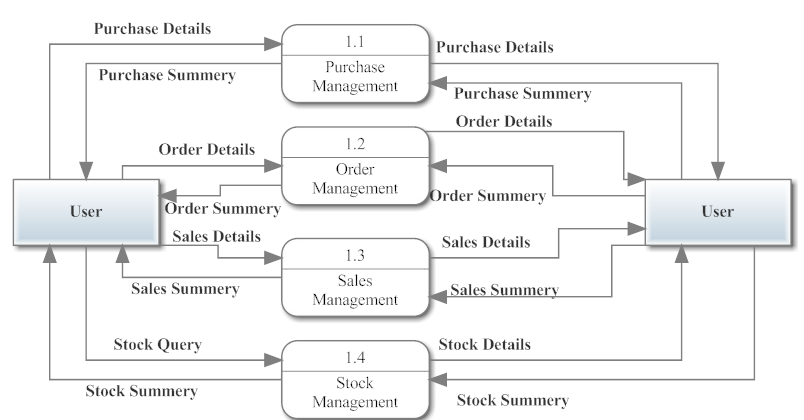
Decomposed data flow squares and circles can have same names.

Draw all data flow around the outside of the diagram.

**Context Level (Level 0)**

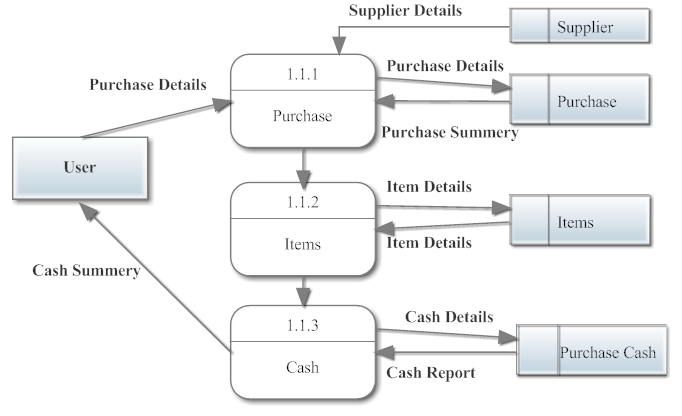


**Level 1**

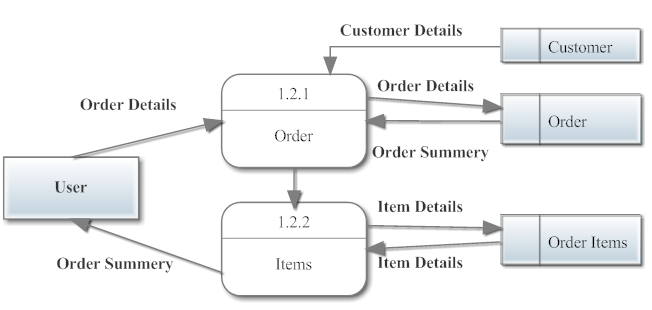


**Level 2**

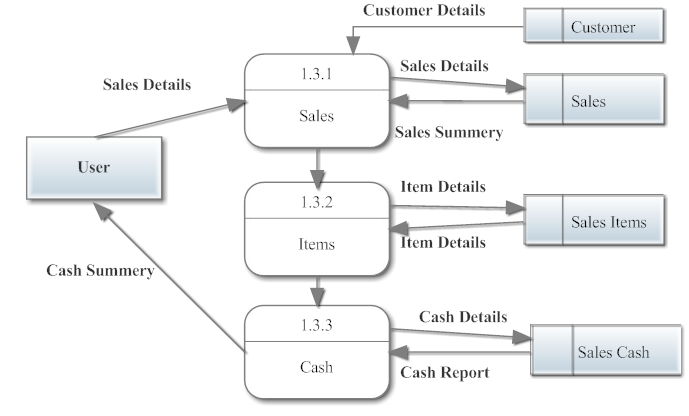
*Purchase Material*



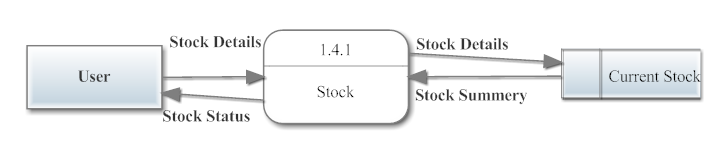
*Take Orders from Customers*



*Sale Products*

**

*Stock*



**4.2 ER DIAGRAM**

Data modeling is a part of the development process. In the linear development cycle, it is used during the system requirements phase to construct the data components of the analysis model. This model represents the major data objects and the relation between them. As in a DFD, a model of data consists of a number of symbols joined up according to certain conventions. System designers describe this conceptual modeling method known as entity relationship analysis.

Entity relationship analysis uses three major abstractions to describe data.

1. Entities, which are distinct things in the enterprise.

2. Relationships, which are meaningful interaction between objects.

3. Attributes, which are the properties of the entities and relationships.

An entity-relationship(ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information.

Major ER Diagram Components:

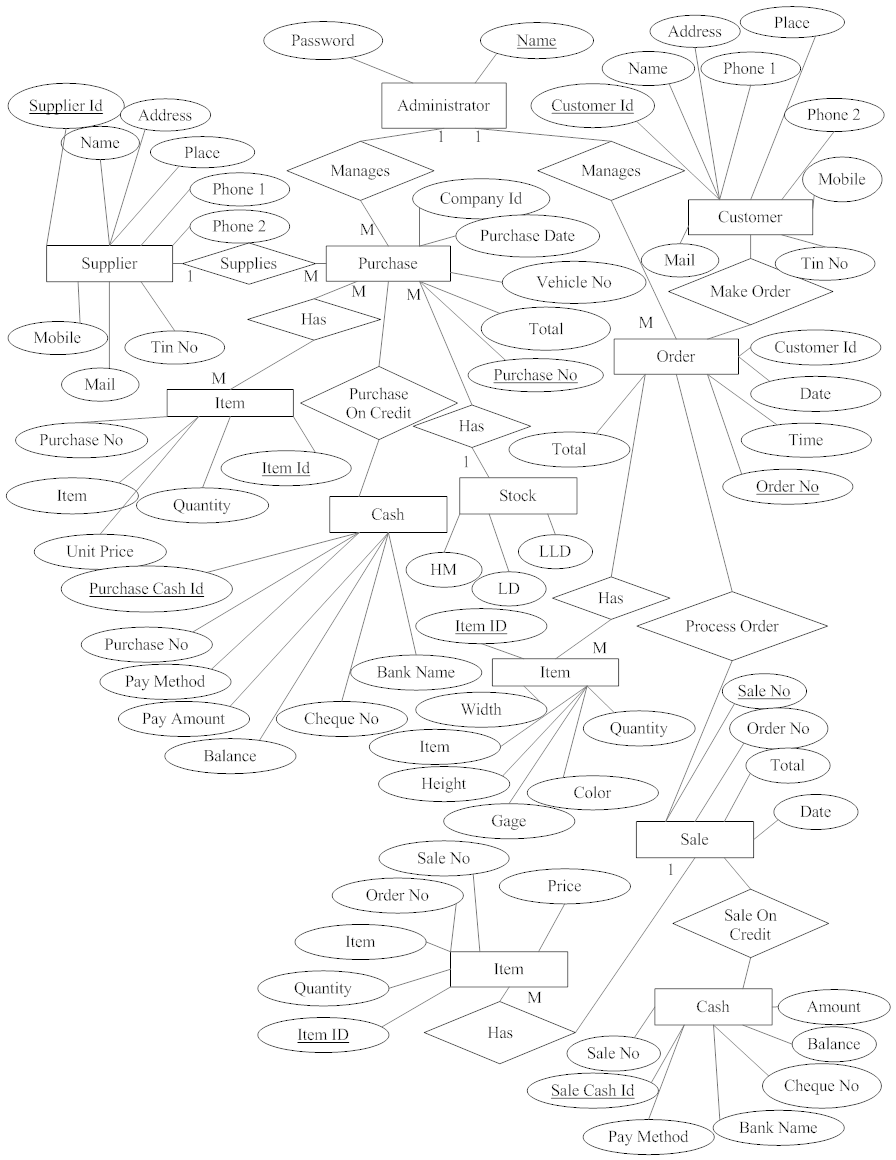
Rectangles, which represent the entity set

Ellipses, which represent attributes

Diamonds, which represent relationship sets

Lines, which link attributes to entity sets and entity sets to relationships

Entity Relationship Diagram Of Polymer Management System



**4.3 INPUT DESIGN**

In the input design, user-oriented inputs are converted into a computer based system format. It also includes determining the record media, method of input, speed of capture and entry on to the screen. Online data entry accepts commands and data through a keyboard. The major approach to input design is the menu and the prompt design. In each alternative, the user’s options are predefined. The data flow diagram indicates logical data flow, data stores, source and destination. Input data are collected and organized into a group of similar data. Once identified input media are selected for processing.

In this software, importance is given to develop Graphical User Interface (GUI), which is an important factor in developing efficient and user-friendly software. For inputting user data, attractive forms are designed. User can also select desired options from the menu, which provides all possible facilities.

Also the important input format is designed in such a way that accidental errors are avoided. The user has to input only just the minimum data required, which also helps in avoiding the errors that the users may make. Accurate designing of the input format is very important in developing efficient software. The goal or input design is to make entry as easy, logical and free from errors.

**4.4 OUTPUT DESIGN**

In the output design, the emphasis is on producing a hard copy of the information requested or displaying the output on the CRT screen in a predetermined format. Two of the most output media today are printers and the screen. Most users now access their reports from a hard copy or screen display. Computer’s output is the most important and direct source of information to the user, efficient, logical, output design should improve the systems relations with the user and help in decision-making.

As the outputs are the most important source of information to the user, better design should improve the system’s relation and also should help in decision-making. The output device’s capability, print capability, print capability, response time requirements etc should also be considered form design elaborates the way output is presented and layout available for capturing information. It’s very helpful to produce the clear, accurate and speedy information for end users.

**4.5 DATABASE DESIGN**

In designing a database application you must set up not only the program’s routines for maximum performance, but you must pay attention also to the physical layout of the data storage.

A good data base design does the following:

Provides minimum search times when locating specific records.

Stores the data in the most efficient manner possible to keep the database from growing large.

Make data updates as easy as possible.

Is flexible enough to allow inclusion of new functions required of program.

Data normalization is the process of eliminating redundant data within a database. Taking data normalization fully results in each piece of information in a database appearing only once, although that’s always not practical.

Data normalization can handle by creating child tables. A child table is a table in which all the entries share common information that is stored in another tablet table containing the common information is called a parent table, and table containing member’s first name is child table. A look up table is another way to store information to prevent data redundancy and to increase the accuracy of data entry functions.

The objectives of database are accuracy and integrity, successful recovery from failure, privacy and security of data and good overall performance database is a collection of interrelated data stored with minimum redundancy to serve many users quick inexpensive and flexible for users.

**TABLE DESIGN**

1. Table Name: Supplier

Primary Key: SupplierId

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| SupplierId | Int(20) | Primary Key | Supplier Identifier |
| Name | varchar(50) | Notnull | Supplier Name |
| Address | varchar(100) | Notnull | Suplier Address |
| Place | varchar(50) | Notnull | Supplier Place |
| Phone1 | varchar(50) | Notnull | Supplier Phone number |
| Phone2 | varchar(50) | Notnull | Supplier Phone number |
| Mob | varchar(50) | Notnull | Supplier Mobile Number |
| Mail | varchar(50) | Notnull | Supplier Mail address |
| TinNo | int(20) | Null | Supplier Tin Number |

1. Table Name :Purchase

Primary Key: PurchaseNo

Foreign Key: Supplierid

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| PurchaseNo | int(20) | Primary Key | Purchase Number |
| SupplierId | int(20) | Foreign Key | Supplier Identifier |
| Date | date | Notnull | Purchase Date |
| VehicleNo | varchar(50) | Notnull | Vehicle Number |
| Total | float | Notnull | Total Amount |

3.Table Name: Purchase Item

Primary Key:ItemId

Foreign Key :PurchaseNo

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| ItemId | int(20) | Primary Key | Item Identifier |
| PurchaseNo | int(20) | Foreign Key | Purchase Number |
| Item | varchar(50) | Notnull | Item Name |
| Qty | varchar(50) | Notnull | Item Quantity |

4.Table Name: Purchase Cash

Primary Key:PurchaseCashId

Foreign Key :PurchaseNo

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| PurchaseCashId | int(20) | Primary Key | Purchase Cash Identifier |
| PurchaseNo | int(20) | Foreign Key | Purchase Number |
| PayMethod | varchar(50) | Notnull | Paying Method |
| PayDate | date | Notnull | Paying Date |
| Amount | float | Notnull | Amount Paying |
| Balance | float | Notnull | Balance |
| ChequeNo | varchar(50) | Notnull | Cheque Number |
| BankName | varchar(50) | Notnull | Bank Name |

5.Table Name: Customer

Primary Key:CustomerId

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| CustomerId | Int(20) | Primary Key | Customer Identifier |
| Name | varchar(50) | Notnull | Customer Name |
| Address | varchar(100) | Notnull | Customer Address |
| Place | varchar(50) | Notnull | Customer Place |
| PhNo1 | varchar(50) | Notnull | Customer Phone number |
| PhNo2 | varchar(50) | Notnull | Customer Phone number |
| Mob | varchar(50) | Notnull | Customer Mobile Number |
| Mail | varchar(50) | Notnull | Customer Mail address |
| Tin | int(20) | Null | Customer Tin Number |

6.Table Name: Order

Primary Key:PurchaseCashId

Foreign Key :PurchaseNo

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| OrderId | int(20) | Primary Key | Order Identifier |
| CustomerId | int(20) | Foreign Key | Customer Identifier |
| Date | date | Notnull | Order Date |
| Time | date | Notnull | Order Time |
| Total | float | Notnull | Total Amount |

7.Table Name : Order Item

Primary Key: Item\_Id

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| Item\_Id | int(20) | Primary Key | Item Identifier |
| Item | varchar(20) | Notnull | Item Name |
| Width | int(20) | Notnull | Item Width |
| Height | int(20) | Notnull | Item Height |
| Gage | int(20) | Notnull | Item Gage |
| Color | varchar(20) | Notnull | Item Color |
| Qty | float | Notnull | Item Quantity |
| ItemType | varchar(20) | Notnull | Item Type |

8.Table Name: Current Stock

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| LD | float | Notnull | Material LD |
| LLD | float | Notnull | Material LLD |
| HM | float | Notnull | Material HM |

9.Table Name: Sale Item

Primary Key:Sale\_No

Foreign Key :OrderNo

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| Sale\_No | int(20) | Primary Key | Sale Number |
| OrderNo | int(20) | Foreign Key | Order Number |
| Item | varchar(50) | Notnull | Item Name |
| Qty | varchar(50) | Notnull | Item Quantity |
| Price | float | Notnull | Item Price |

10.Table Name: Sale

Primary Key:Sale\_No

Foreign Key :OrderNo

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| Sale\_No | int(20) | Primary Key | Sale Number |
| OrderNo | int(20) | Foreign Key | Order Number |
| Total | float | Notnull | Total Amount |
| Date | date | Notnull | Sale Date |

11.Table Name: Sale Cash

Foreign Key :Sale\_No

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| Sale\_No | int(20) | Foreign Key | Sale Number |
| PayMethod | varchar(50) | Notnull | Paying Method |
| Amount | float | Notnull | Amount Paying |
| Balance | float | Notnull | Balance |
| Cheque\_No | varchar(50) | Notnull | Cheque Number |
| Bank\_Name | varchar(50) | Notnull | Bank Name |

12.Table Name :Admin

Primary Key :Name

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| Name | varchar(50) | Primary Key | Admin Name |
| Password | varchar(50) | Notnull | Admin Password |

**SYSTEM DEVELOPEMENT**

**5.SYSTEM DEVELOPMENT**

**5.1 CODING STRUCTURE**

**.Net Platform**

The .NET Framework (pronounced dot net) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large library and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for the .NET Framework execute in a software environment (as contrasted to hardware environment), known as the Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework.

The .NET Framework's Base Class Library provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their own source code with the .NET Framework and other libraries. The .NET Framework is intended to be used by most new applications created for the Windows platform. Microsoft also produces an integrated development environment largely for .NET software called Visual Studio.

Design features

***Interoperability***

Because computer systems commonly require interaction between newer and older applications, the .NET Framework provides means to access functionality implemented in newer and older programs that execute outside the .NET environment. Access to COM components is provided in the System.Runtime. InteropServices and System.EnterpriseServices namespaces of the framework; access to other functionality is achieved using the P/Invoke feature.

***Common Language Runtime engine***

The Common Language Runtime (CLR) serves as the execution engine of the .NET Framework. All .NET programs execute under the supervision of the CLR, guaranteeing certain properties and behaviors in the areas of memory management, security, and exception handling.

***Language independence***

The .NET Framework introduces a Common Type System, or CTS. The CTS specification defines all possible datatypes and programming constructs supported by the CLR and how they may or may not interact with each other conforming to the Common Language Infrastructure (CLI) specification. Because of this feature, the .NET Framework supports the exchange of types and object instances between libraries and applications written using any conforming .NET language.

***Base Class Library***

The Base Class Library (BCL), part of the Framework Class Library (FCL), is a library of functionality available to all languages using the .NET Framework. The BCL provides classes that encapsulate a number of common functions, including file reading and writing, graphic rendering, database interaction, XML document manipulation, and so on. It consists of classes, interfaces of reusable types that integrates with CLR(Common Language Runtime).

***Simplified deployment***

The .NET Framework includes design features and tools which help manage the installation of computer software to ensure it does not interfere with previously installed software, and it conforms to security requirements.

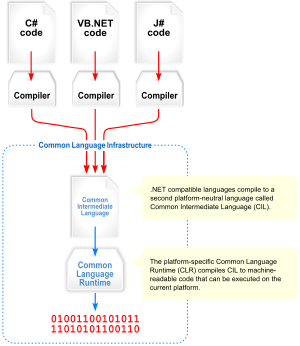
***Security***

The design addresses some of the vulnerabilities, such as buffer overflows, which have been exploited by malicious software. Additionally, .NET provides a common security model for all applications.

***Portability***

While Microsoft has never implemented the full framework on any system except Microsoft Windows, it has engineered the framework to be platform-agnostic, and cross-platform implementations are available for other operating systems (see Silverlight and the Alternative implementations section below). Microsoft submitted the specifications for the Common Language Infrastructure (which includes the core class libraries, Common Type System, and the Common Intermediate Language), the C# language, and the C++/CLI language to both ECMA and the ISO, making them available as official standards. This makes it possible for third parties to create compatible implementations of the framework and its languages on other platforms.

***Architecture***



***Visual overview of the Common Language Infrastructure (CLI)***

Common Language Infrastructure (CLI)

The purpose of the Common Language Infrastructure (CLI) is to provide a language-neutral platform for application development and execution, including functions for Exception handling, Garbage Collection, security, and interoperability. By implementing the core aspects of the .NET Framework within the scope of the CL, this functionality will not be tied to a single language but will be available across the many languages supported by the framework. Microsoft's implementation of the CLI is called the Common Language Runtime, or CLR.

***Security***

.NET has its own security mechanism with 2 general features: Code Access Security (CAS), and validation and verification. Code Access Security is based on evidence that is associated with a specific assembly. Typically the evidence is the source of the assembly (whether it is installed on the local machine or has been downloaded from the intranet or Internet). Code Access Security uses evidence to determine the permissions granted to the code. Other code can demand that calling code is granted a specified permission. The demand causes the CLR to perform a call stack walk: every assembly of each method in the call stack is checked for the required permission; if any assembly is not granted the permission a security exception is thrown.

***Class library***

The .NET Framework includes a set of standard class libraries. The class library is organized in a hierarchy of namespaces. Most of the built-in APIs are part of either System.\* or Microsoft.\* namespaces. These class libraries implement a large number of common functions, such as file reading and writing, graphic rendering, database interaction, and XML document manipulation, among others. The .NET class libraries are available to all CLI compliant languages. The .NET Framework class library is divided into two parts: the Base Class Library and the Framework Class Library

***Memory management***

The .NET Framework CLR frees the developer from the burden of managing memory (allocating and freeing up when done); it handles memory management itself by detecting when memory can be safely freed. Memory is allocated to instantiations of .NET types (objects) from the managed heap, a pool of memory managed by the CLR. As long as there exists a reference to an object, which might be either a direct reference to an object or via a graph of objects, the object is considered to be in use. When there is no reference to an object, and it cannot be reached or used, it becomes garbage, eligible for collection. NET Framework includes a garbage collector which runs periodically, on a separate thread from the application's thread, that enumerates all the unusable objects and reclaims the memory allocated to them.

**C Sharp (programming language)**

C# (pronounced see sharp) is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure.C# is intended to be a simple, modern, general-purpose, object-oriented programming language.

***Design goals***

The ECMA standard lists these design goals for C#

* The C# language is intended to be a simple, modern, general-purpose, object-oriented programming language.
* The language, and implementations thereof, should provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. Software robustness, durability, and programmer productivity are important.
* The language is intended for use in developing software components suitable for deployment in distributed environments.
* Source code portability is very important, as is programmer portability, especially for those programmers already familiar with C and C++.
* Support for internationalization is very important.
* C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions.
* Although C# applications are intended to be economical with regard to memory and processing power requirements, the language was not intended to compete directly on performance and size with C or assembly language.

SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database, it is just a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for different workloads (ranging from small applications that store and retrieve data on the same computer, to millions of users and computers that access huge amounts of data from the Internet at the same time). Its primary query languages are T-SQL and ANSI SQL.

SQL Server Advantages

* It does support Multiple active result sets (MARS)
* It has been enhanced in such a way that developers program more powerful and error resistant SQL code with structured exception handling
* Performance of query execution is much faster as compared to sqlserver 2000
* It supports Integration of the .NET common language runtime (CLR).By default it is off. we explicitly need to start configuration from setting
* It has a new XML data type as well as a new varbinary(max) data type

Features in SQL Server 2005

1. T-SQL (Transaction SQL) enhancements

T-SQL is the native set-based RDBMS programming language offering high-performance data access. It now incorporates many new features including error handling via the TRY and CATCH paradigm, Common Table Expressions (CTEs), which return a record set in a statement, and the ability to shift columns to rows and vice versa with the PIVOT and UNPIVOT commands.

2. CLR (Common Language Runtime)

The next major enhancement in SQL Server 2005 is the integration of a .NET compliant language such as C#, ASP.NET or VB.NET to build objects (stored procedures, triggers, functions, etc.). This enables you to execute .NET code in the DBMS to take advantage of the .NET functionality. It is expected to replace extended stored procedures in the SQL Server 2000 environment as well as expand the traditional relational engine capabilities.

3. Service Broker

The Service Broker handles messaging between a sender and receiver in a loosely coupled manner. A message is sent, processed and responded to, completing the transaction. This greatly expands the capabilities of data-driven applications to meet workflow or custom business needs.

4. Data encryption

SQL Server 2000 had no documented or publicly supported functions to encrypt data in a table natively. Organizations had to rely on third-party products to address this need. SQL Server 2005 has native cap

**TESTING & MPLEMENTATION**

**6.TESTING AND IMPLEMENTATION**

**6.1 TESTING**

System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error. A good test is one that has a probability of finding an as yet undiscovered error. The purpose of testing is to identify and correct bugs in the developed system. Nothing is complete without testing. Testing is the vital to the success of the system.

In the code testing the logic of the developed system is tested. For this every module of the program is executed to find an error. To perform specification test, the examination of the specifications stating what the program should do and how it should perform under various conditions.

Unit testing focuses first on the modules in the proposed system to locate errors. This enables to detect errors in the coding and logic that are contained within that module alone. Those resulting from the integration between modules are initially avoided. In unit testing step each module has to be checked separately.

System testing does not test the software as a whole, but rather than integration of each module in the system. The primary concern is the compatibility of individual modules. One has to find areas where modules have been designed with different specifications of data lengths, type and data element name.

Testing and validation are the most important steps after the implementation of the developed system. The system testing is performed to ensure that there are no errors in the implemented system. The software must be executed several times in order to find out the errors in the different modules of the system.

Validation refers to the process of using the new software for the developed system in a live environment i.e., new software inside the organization, in order to find out the errors. The validation phase reveals the failure and the bugs in the developed system. It will be come to know about the practical difficulties the system faces when operated in the true environment. By testing the code of the implemented software, the logic of the program can be examined. A specification test is conducted to check whether the specifications starting the program are performing under various conditions.

The entire testing process is different types.

* Unit testing
* Integration testing
* Validation testing
* Output testing
* User acceptance testing

**Unit Testing:-**

It focuses efforts on the smallest unit of software design. This is known as module testing. Each unit is tested separately for errors and defects found are corrected. The modules are tested separately. In database management, the data and design of tables are tested separately for errors. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output for the module. There are some validation checks for the fields.

**Integration Testing:-**

It is a systematic approach for constructing the program structure, while at the same time constructing test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

**Validation testing:-**

Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that can reasonable expected by the client. That means the input we are entering to the system “Project Scheduling” will be checked whether the entered input is valid or not.

**Output Testing:-**

After performing validation testing, the next step is output testing of the proposed system could be useful if it does not produce the required output in the specific format. Here the output format is considered in two ways. One is on screen and another one is in printed format. All the reports which are generating will be tested.

**User acceptance Testing:-**

User acceptance testing of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the system uses at the time of developing and making changes whenever required. This is done with regard to the following points.

* Input screen design
* Output screen design
* Menu driven system
* Format reports

# 6.2 IMPLEMENTATION

Implementation includes all those activities that take place to convert from the old system to the new. The old system consists of manual operations, which is operated in a very different manner from the proposed new system. A proper implementation is essential to provide a reliable system to meet the requirements of the organizations. An improper installation may affect the success of the computerized system.

Implementation Methods:

There are several methods for handling the implementation and the consequent conversion from the old to the new computerized system.

The most secure method for conversion from the old system to the new system is to run the old and new system in parallel. In this approach, a person may operate in the manual older processing system as well as start operating the new computerized system. This method offers high security, because even if there is a flow in the computerized system, we can depend upon the manual system. However, the cost for maintaining two systems in parallel is very high. This outweighs its benefits.

Another commonly method is a direct cut over from the existing manual system to the computerized system. The change may be within a week or within a day. There are no parallel activities. However, there is no remedy in case of a problem. This strategy requires careful planning.

Implementation Plan:

The implementation plan includes a description of all the activities that must occur to implement the new system and to put into operation. It identifies the personnel responsible for the activities and prepares a time chart for implementing the system. The implementation plan consists of the following steps:

\* List all files required for implementation

\* Identify all data required to build new files during the implementation.

\* List all new documents and procedures that go into the new system.

The implementation plan should anticipate possible problems and must be able to deal with them. The usual problems may be missing documents; mixed data formats between current and files, errors in data translation, missing data etc.

Education and training:

The implementation of the proposed system includes the training of system operators. Training the system operators includes not only instructions in how to use the equipment, but also in how to diagnose malfunctions and in what steps to take when they occur. So proper training should be provided to the system operators. No training is complete without familiarizing users with simple system maintenance activities. Since the proposed system is developed in a GUI, training will be comparatively easy than systems is developed in a non GUI. There are different types of training. We can select off-site to give depth knowledge to the system operators.

Success of the system depends on the way in which it is operated and used. Therefore the quality of training given to the operating person affects the successful implementation of the system. The training must ensure that the person can handle all the possible operations.

Training must also include data entry personnel. They must also be given training for the installation of new hardware, terminals, how to power the system, how to power it down, how to detect the malfunctions, how to solve the problems etc. The operators must also be provided with the knowledge of trouble shooting which involves the determination of the problem.

The proposed system requires trained personnel for operating the system. Data entry jobs must be done utmost carefully to avoid errors. This will reduce the data entry errors considerably. It is preferable to provide the person with some kind of operating manuals that will explain all the details of the system.

**SYSTEM MAINTENANCE**

**7. SYSTEM MAINTENANCE**

Authentication and Authorization

Only the administrator can login to the system. The Administrator can do the changes like insertion, deletion and updating of data. Therefore anonymous deletion of data protected.

Record Verification

At every stage of data entry the values entered are checked and necessary comments and validity massages are provided to the user. Every record is checked for completeness and accuracy and then it is allowed to enter into the database.

**FUTURE ENHANCEMENT**

**8.SCOPE FOR FUTURE ENHANCEMENT**

High speed, accuracy and non-redundant data are the main advantage of the proposed system. In proposed system the user is provided with a choice of data screen, which are similar in formats to the source documents. Data entry errors can be minimized through validity checks. After the verification only the data are placed the permanent database. The software can be developed further to include a lot of modules because the proposed system is developed on the view of future. Now the proposed system is based on pc and intranet but in the future .if we need to convert it into internet then we need to change the front end only because we are developing this on the basis of OOP technology and most of the business logic’s are bounded in the class files and modules like reusable components.

**CONCLUSION**

**9.CONCLUSION**

The “Polythene Company Management System” has been developed to overcome the problem faced in the present system. The developed software was fond to work out the operation effectively. The objectives of the system have been achieved.

The goals that have been achieved:

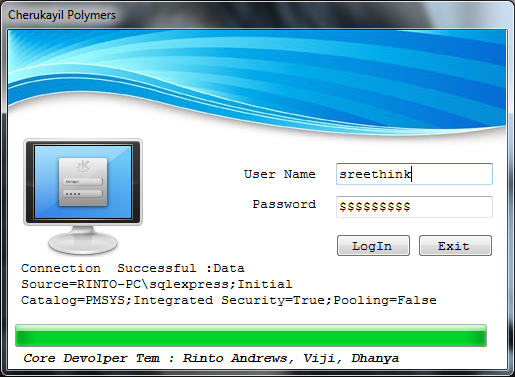
* Reduce data redundancy and inconsistency.
* Ensure process time and increase throughput.
* Simplifies the operation.
* User friendly input screens to enter data.
* Avoid some manual work in the existing system.

A consistent system and efficient system has been successfully developed, implemented and tested. The system has been developed using C# and SqlServer .The system is very flexible and user friendly. So further changes can be incorporated into the system easily. Adequate documentation provides for maintenance and further enhancement.

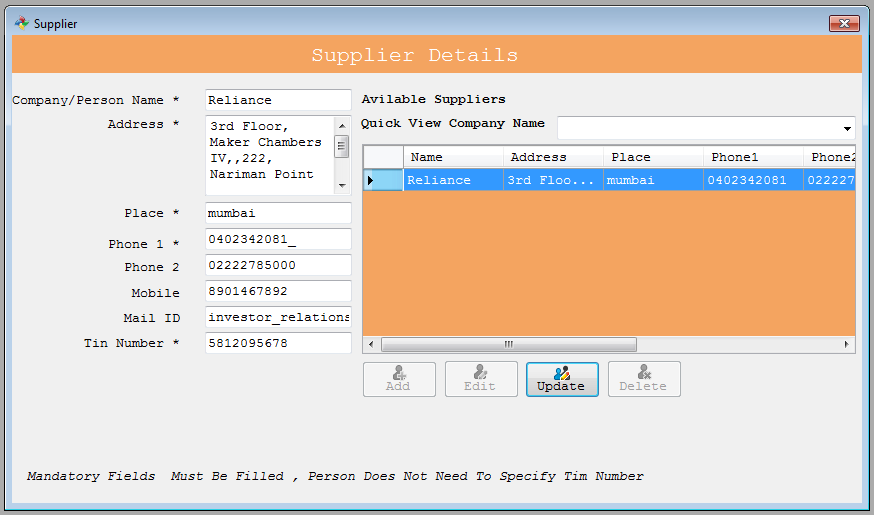
**SCREEN LAYOUTS**

**10.SCREEN LAYOUTS**

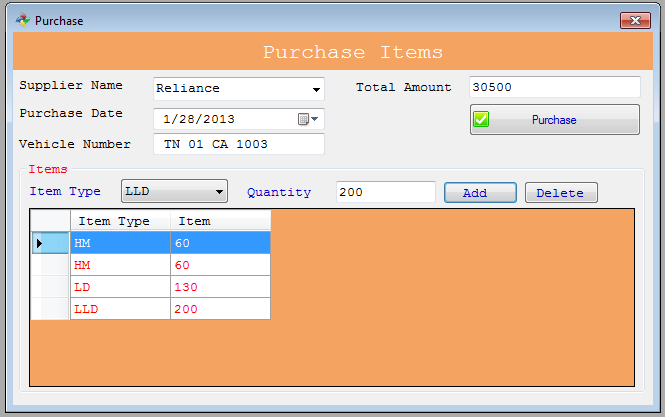
**Login**

****

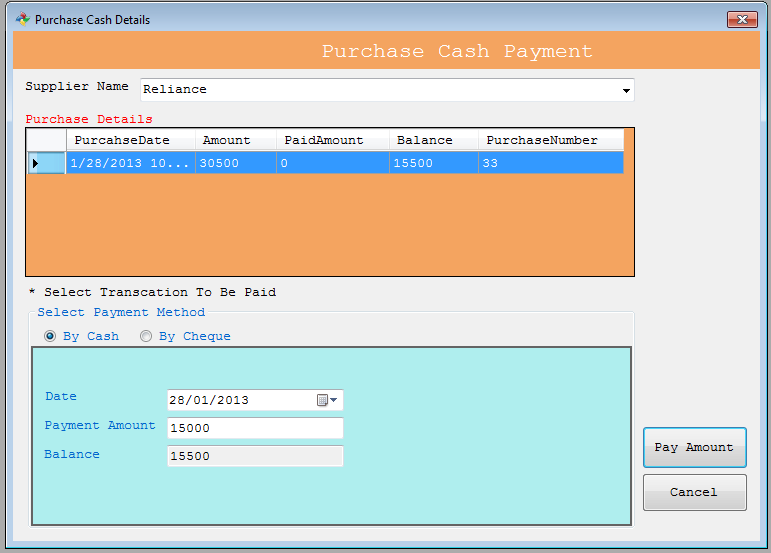
**Supplier Details**

****

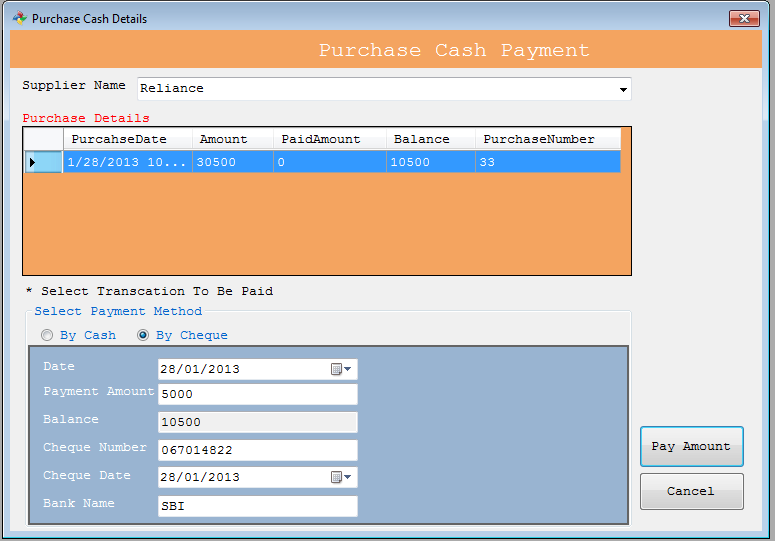
**Purchase**

****

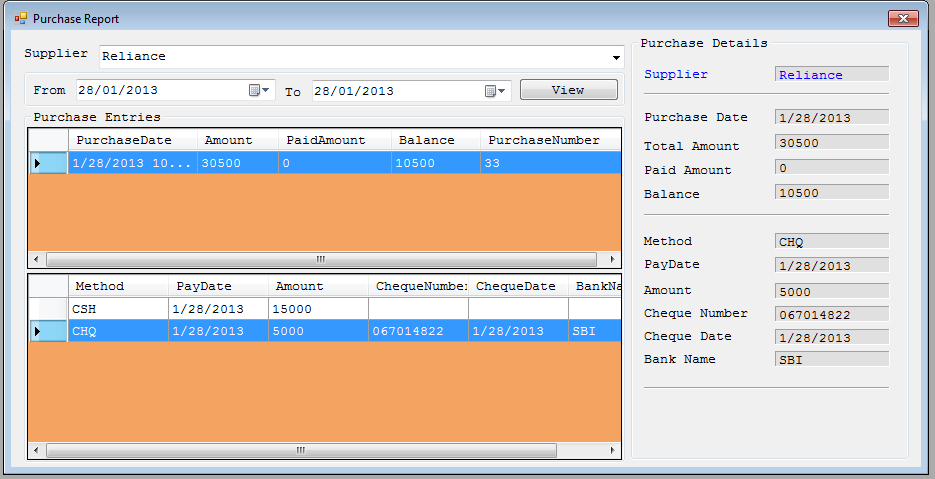
**Payment by Cash**



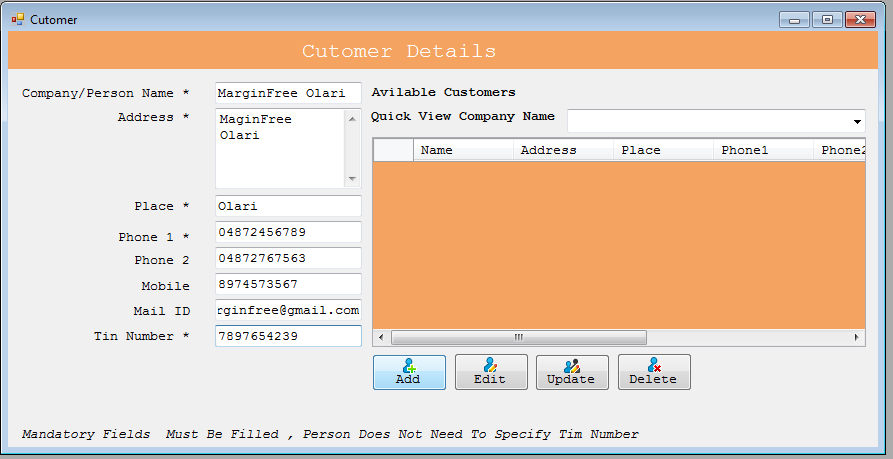
**Payment by Cheque**

****

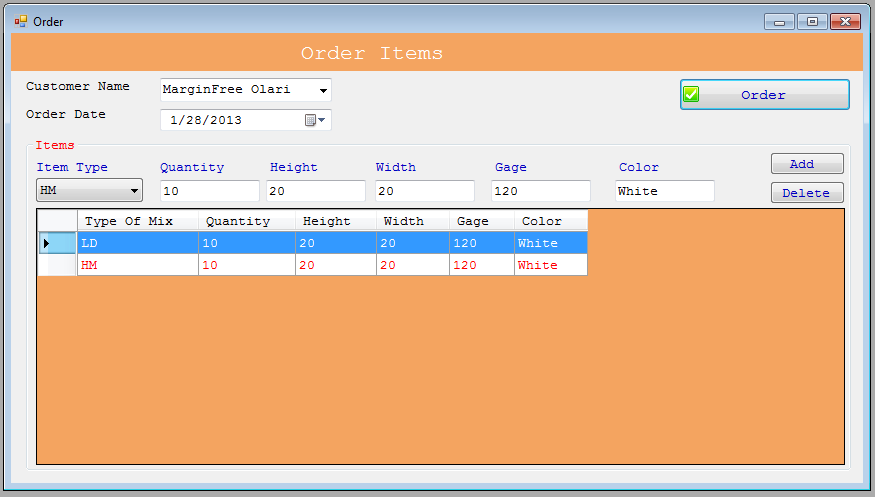
**Purchase Report**

****

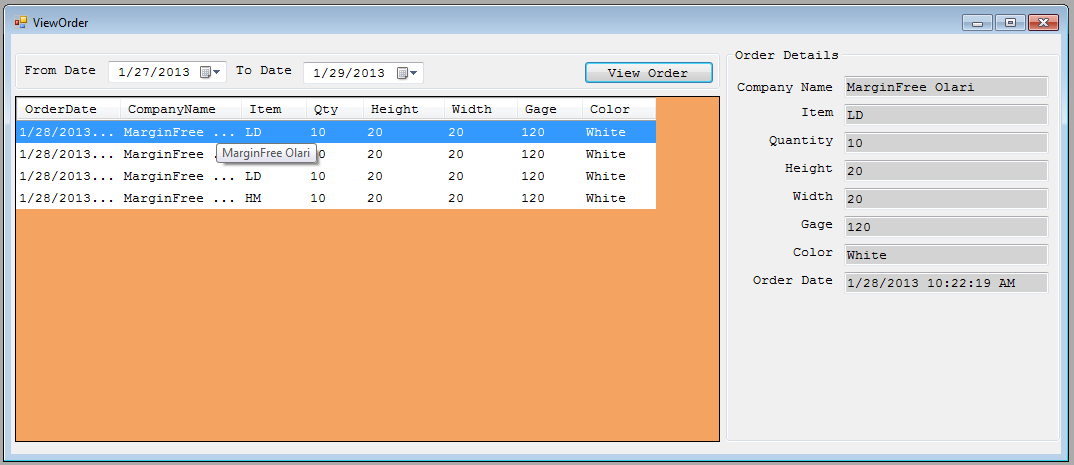
**Customer Details**

****

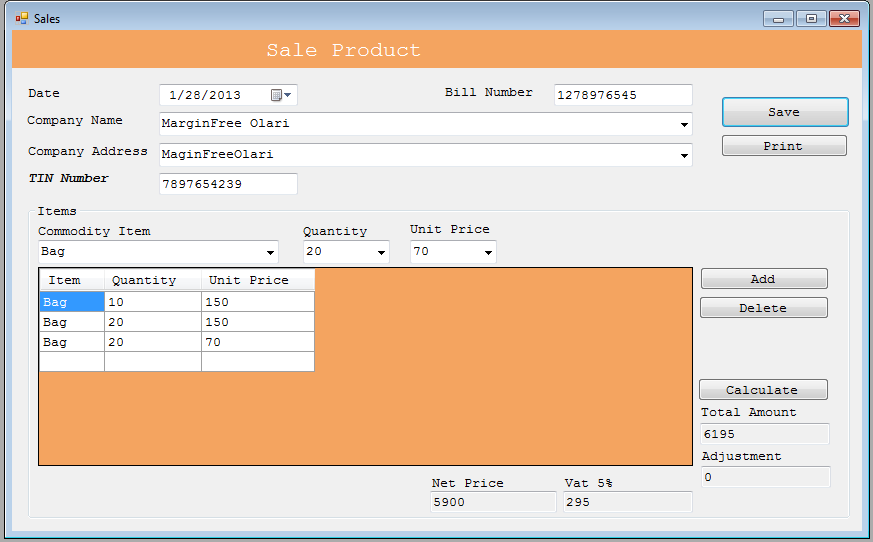
**Order**

****

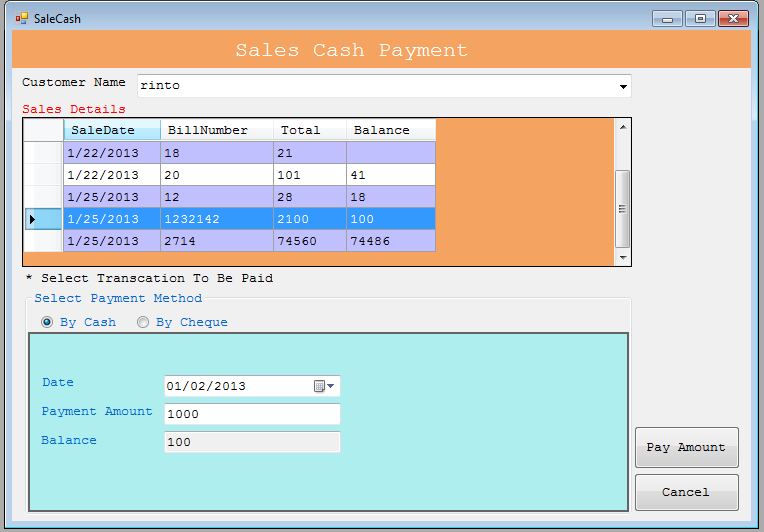
**Order Report**



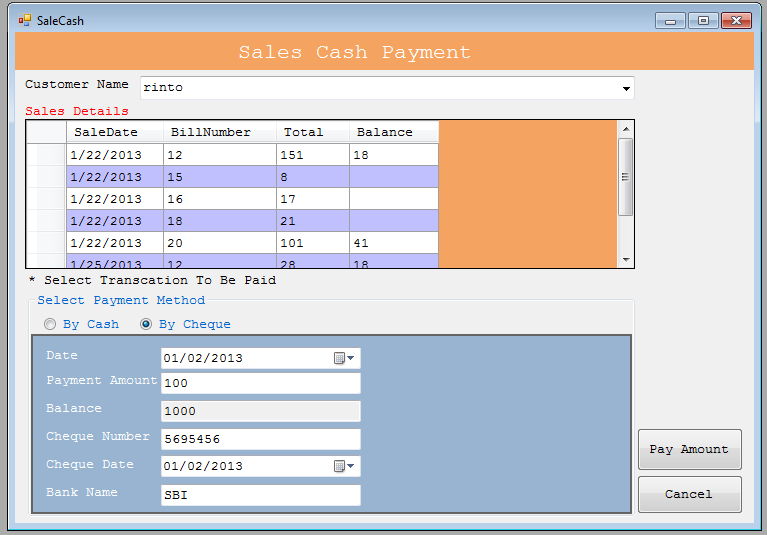
**Sales**



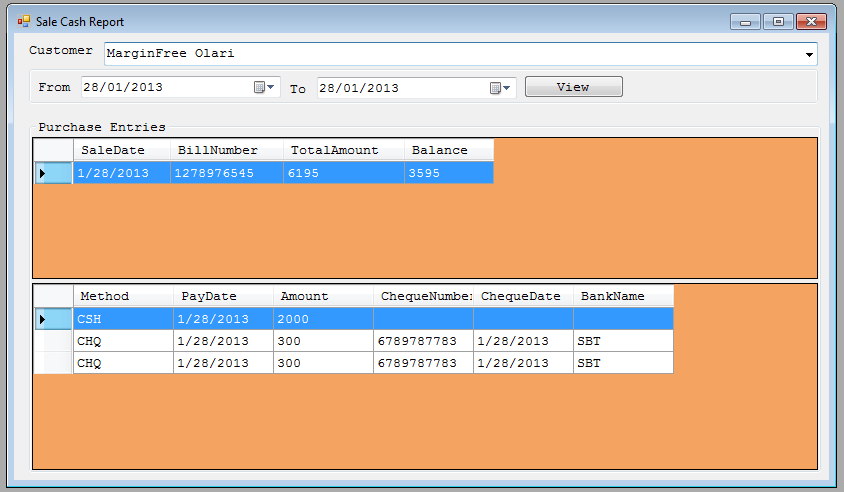
**Payment by Cash**

****

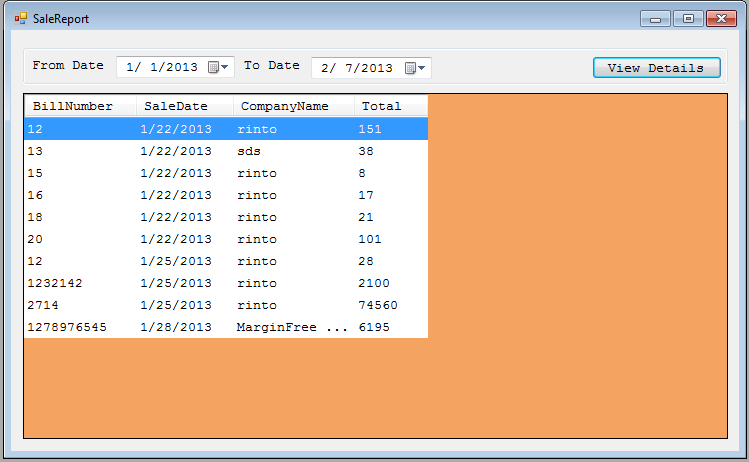
**Payment by Cheque**

****

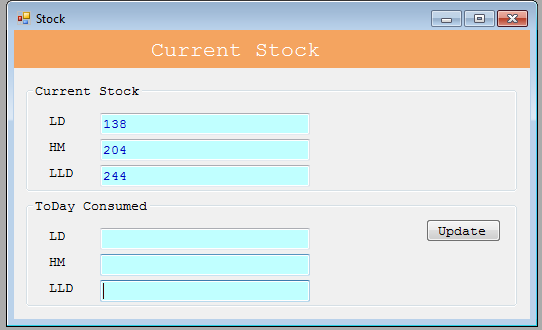
Sales Cash Report



**Sales Report**

****

Stock Details



**BIBLIOGRAPHY**

**11. BIBLIOGRAPHY**

* Rob Miles - C# Yellow Book
* Programming In C# - Balagurusamy
* Programming C#, 4th Edition - O'Reilly Media
* Books - C# Corner
* Structured Query Language (SQL): a Practical Introduction

**WEBLIOGRAPHY**

http://msdn.microsoft.com

http://[www.w3schools.com](http://www.w3schools.com)