**ABSTRACT**

This project **“Stock Brokering System”** is developed to maintain the information about stock availability, purchase of different items from different producers and after packing, delivering the items to customers for sales in retail market.

The **Stock Brokering System** will keep track of details of its buyers, suppliers and the stock. This software can be used to compare the rate among the products and buy cheaper product at better quality and sell the same for better profit. This software will provide the exact amount of stock available for sales, stock-in- progress and stock in warehouse.

This software contains various modules to maintain information about stock, production, purchase, sales, accounts, etc. The software will reduce the hindrance on the stock management. This software will provide proper updated information on time.

This software is developed in Visual Basic6.0 as front end tool and MS-Access as back end tool.

**CHAPTER-I**

**INTRODUCTION**

This project, **Stock Brokering System** is developed to facilitate the process of buying items from producers and selling the packed items in retail market. This software is developed to decrease the work that is done manually.

**Stock Brokering System** maintains information about stock, production, purchase, sales, suppliers and customers.

Various items are purchased from various producers, and then sent for packing in different quantities. The packed items are then sent to customers for retail sale. The stock availability should be properly monitored so that production should not be affected because of the unavailability of the stock.

This system maintains the record of all sales and purchases, finally reports are generated for the interested group of people to check the operation of the company.

**CHAPTER - II**

**PROBLEM DEFINITION AND FEASIBILITY ANALYSIS**

**2.1 PROBLEM DEFINITION**

The **Stock Brokering System** is project to manage the stock of a company, which purchases items in bulk from producers and pack it into different sized to sell in retail market.

The company needs to maintain information about producers or suppliers for purchase, items details for production, customer details for sales and accounting information.

So it is proposed to develop a software for the company to manage the information and produce various reports on time for the smooth run of the company.

**2.2 EXISTING SYSTEM**

In the existing system, all entry are done manually (i.e.) maintain the Customer purchase Details, Stock purchase Details and Plan Details. The newly computerized system has more efficiency, accuracy and speed. Quick to access and timely services are also provided

**2.3 DRAWBACKS OF EXISTING SYSTEM**

* Time consuming
* Requires more manual work
* Requires manual calculation

**2.4 PROPOSED SYSTEM**

The proposed software is developed in VB 6.0 as the front end tool and MS-Access as back end tool. This system contains details of the customer, stock suppliers, production, etc. As the proposed software is developed in VB, it manages all information and produces various reports for the smooth run of the company.

**2.5 FEATURES OF PROPOSED SYSTEM**

The features in computerizing over the existing system

* Reduced chance of miscalculation.
* User friendly and easy to handle.
* Reduce manual work.

**2.6 FEASIBILITY STUDY**

The feasibility study is made to see if the project on complication will serve the purpose of the organization of the amount of the work effort and the time that spent on it. The main of feasibility study is to weight up three types of feasibility

* Technical Feasibility
* Economical Feasibility
* Operational Feasibility

**2.6.1 TECHNICAL FEASIBILITY**

Technical Feasibility analysis makes a comparison between the levels of technology available and the technology that is needed for the development of the project. The levels of technology are determined by the factors such as the software tools, machine environment, platform, etc. Since the resources required for the development and operation of the project is available, this process is technically feasible.

**2.6.2 ECONOMICAL FEASIBLITY**

Economical feasibility is the most important characteristics that have to be evaluated. This is necessary for the project. It is always observed that the benefit overrides the cost. Hence the project is economically feasible.

**2.6.3 OPERATIONAL FEASIBILITY**

Operational Feasibility study is necessary as it ensures that the project developed would be used by the user for routine work. The operational feasibility of this project is high, since it is User-friendly and easy to operate. Hence, the entire system is feasible.

**CHAPTER-III**

**SOFTWARE REQUIREMENTS SPECIFICATION**

**3.1 PURPOSE**

The purpose of the software requirements specification is to produce the specification of the system analysis task. Also it establishes complete information about the behavior, constrains, performance, functional and non-functional requirements of the system. This document stands as a contract between the customer and the developers/designers pertaining to functional and non-functional requirements that system must exhibit.

**3.2 SCOPE**

This document will be used throughout the subsequent development phases to ensure the successful development of the product and it act as the basis for validating the final delivered system. This document characterizes the software requirements of the system agreed upon by the customer and developers/designers.

It may be referred in case of confusion or inconsistence, the scope extends to customer and developers/designers. The document should be made available for reference on demand. In future, if any changes are to be made in the requirements specification, it must undergo a change approval process.

**3.3 HARDWARE REQUIREMENTS**

Central processing unit : Intel ® Core TM 2 Quad

Main memory : 512 MB

Hard Disk Capacity : 40 GB

Monitor : 14”SVGA color

CD Drive : 16 x RW

Mouse : Scroll Mouse

Keyboard : Multimedia

**3.4 SOFTWARE REQUIREMENTS**

Operating System : Window XP/7

Front End Tool : VB6.0

Back End Tool : MS-Access

**CHAPTER-IV**

**SYSTEM DESIGN**

**4.1 ARCHITECTURAL DESIGN**

The primary objective of architectural design is to develop a modular program structure and represent the control relationship modules. In addition, architectural design module program structure and data structure, defining interface that enables data to flow throughout the program. Architectural design has its origins in earlier design concepts that stressed modularity, top-down design and structured programming.

The goal of architectural design is to specify internal structure and processing details to record design decision and indicate why certain alternative tradeoffs where chosen and to provide blue print for implementation, testing and maintenance activities. This program consists of the following modules

* Supplier module
* Purchase stock module
* Purchase module
* Production plan module
* Customer module
* Sales stock module
* Sales module

**4.1.1 SUPPLIER MODULE**

* The Supplier module contains basic information like name, address, contact number and email address.
* Contains option to register new Supplier to supply the stock.
* Contains option to edit the information of Supplier.

**4.1.2 PURCHASE STOCK MODULE**

Contain details of Purchase stock such as the quantity in warehouse, unit of measurement and minimum level.

**4.1.3 PURCHASE MODULE**

The Purchase module is used to purchase stock from Supplier and to record the quantity of stock arrived and the quantity yet to arrive.

**4.1.4 PRODUCTION PLAN MODULE**

The Production plan module is used to frame plan for daily production of stock, it contains quantity to be produced on daily basis.

**4.1.5 CUSTOMER MODULE**

* The Customer module contains basic information like name, address, contact number and email address.
* Option to register new Customer for selling the stock.
* Option to edit the information of Customer.

**4.1.6 SALES STOCK MODULE**

This module is used to determine how the Purchase stocks are packed into smaller items. This sale stock is used in Production plan.

**4.1.7 SALES MODULE**

The Sales module is used to sell stock to Customer and to record each selling process.

**4.2** **DATA DICTIONARY :**

In database management system, a file that defines organization of a database. A data dictionary contains a list of all files in the database, the no of records in each file, and the names and the types of each field .Most database management system keeps the data dictionary hidden from user to prevent them from accidently destroying its contents.

A well-developed data dictionary should be able to provide following information:

* How many character are there in a data item
* Where it is used in the system

Data dictionary is an organized listing of all data elements that are pertinent to the system, with precise, rigorous definition so that both user and system analyst will have a common understanding of inputs, outputs, component of stores and even intermediate calculations.

**4.2.1 TABLE DESIGN**

**TABLE NAME:** Customer

**PRIMARY KEY:** Cust\_id

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **TYPE** | **DESCRIPTION** |
| Cust\_id | Text | Specifies the Customer ID |
| Cust\_name | Text | Specifies Customer name |
| House\_no | Integer | Specifies house number |
| Street | Text | Specifies street |
| Area | Number | Specifies area |
| City | Number | Specifies city |
| State | Text | Specifies state |
| Country | Text | Specifies country |
| Pincode | Long | Specifies pin code |
| Zipcode | Long | Specifies zip code |
| Phone\_no1 | Text | Specifies phone number |
| Phone\_no2 | Text | Specifies alternate phone number |
| Lanline | Text | Specifies landline number |
| Email | Text | Specifies email address |

**TABLE NAME:** Prodction\_complete

**PRIMARY KEY:** Pcomplete\_no

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **TYPE** | **DESCRIPTION** |
| Pcomplete\_no | Text | Specifies production complete number |
| Pplan\_no | Text | Specifies production plan number |
| Date | Text | Specifies date |
| Sitem\_no | Text | Specifies sale item number |
| Sitem\_name | Text | Specifies sale item name |
| Pitem\_no | Text | Specifies purchase item number |
| Quantity\_produced | Integer | Specifies quantity produced |

**TABLE NAME:** Production\_plan

**PRIMARY KEY:** Pplan\_no

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **TYPE** | **DESCRIPTION** |
| Pplan\_no | Text | Specifies the supplier name |
| Date | Text | Specifies date |
| Sitem\_no | Text | Specifies sale item number |
| Sitem\_name | Text | Specifies sale item name |
| Pitem\_no | Text | Specifies purchase item number |
| Quantity | Integer | Specifies quantity |
| Status | Text | Specifies status |
| Stock\_in\_progress | Integer | Specifies stock in progress |

**TABLE NAME:** Purchase

**PRIMARY KEY:** Porder\_no

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **TYPE** | **DESCRIPTION** |
| Porder\_no | Text | Specifies purchase order number |
| Date | Text | Specifies date |
| Supp\_id | Text | Specifies supplier id |
| Supp\_name | Text | Specifies supplier name |
| Pitem\_no | Text | Specifies purchase item number |
| Pitem\_name | Text | Specifies purchase item name |
| Quantity | Integer | Specifies quantity |
| Amount | Double | Specifies total amount |
| Mode\_of\_payment | Text | Specifies mode of payment |
| Cheque\_no | Text | Specifies cheque number |
| Account\_no | Text | Specifies account number |
| Account\_holder\_name | Text | Specifies account holder name |
| Amount\_paid | Double | Specifies amount paid |
| Balance\_amount | Double | Specifies balance amount |
| Status | Text | Specifies status |
| Quantity\_ordered | Integer | Specifies quantity ordered |

**TABLE NAME:** Purchase\_stock

**PRIMARY KEY:** Pitem\_no

|  |  |  |
| --- | --- | --- |
| **FILED NAME** | **TYPE** | **DESCRIPTION** |
| Pitem\_no | Text | Specifies purchase item number |
| Pitem\_name | Text | Specifies purchase item name |
| Quantity | Double | Specifies quantity |
| Unit\_of\_measurement | Text | Specifies unit of measurement |
| Min\_level | Double | Specifies minimum level |
| Stock\_in\_progress | Double | Specifies stock in progress |

**TABLE NAME:** Sale\_stock

**PRIMARY KEY:** Sitem\_no

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **TYPE** | **DESCRIPTION** |
| Sitem\_no | Text | Specifies sale item number |
| Pitem\_no | Text | Specifies purchase item number |
| Pitem\_name | Text | Specifies purchase item name |
| Sur\_name1 | Integer | Specifies 2nd last name of sale stock |
| Sur\_name2 | Text | Specifies last name of sale stock |
| Sitem\_name | Text | Specifies sale item name |
| Min\_level | Double | Specifies minimum level |
| Quantity | Double | Specifies quantity |
| Rate | Double | Specifies the rate |
| Pack\_charge | Double | Specifies packing charge |
| Transp\_charge | Double | Specifies transportation charge |
| Cost\_price | Double | Specifies cost price of the item |
| Margin | Double | Specifies margin |

**TABLE NAME:** Stock\_arrival

**PRIMARY KEY:** Stock\_arrival\_no

|  |  |  |
| --- | --- | --- |
| **FILED NAME** | **TYPE** | **DESCRIPTION** |
| Stock\_arrival\_no | Text | Specifies stock arrival number |
| Date | Text | Specifies date of arrival |
| Porder\_no | Text | Specifies purchase order number |
| Quantity\_arrived | Integer | Specifies quantity arrived |

**TABLE NAME:** Sales

**PRIMARY KEY:** Sale-no

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **TYPE** | **DESCRIPTION** |
| Sale\_no | Text | Specifies sale number |
| Date | Text | Specifies date |
| Sitem\_no | Text | Specifies sale item number |
| Sitem\_name | Text | Specifies sale item name |
| Cust\_id | Text | Specifies customer number |
| Cust\_name | Text | Specifies customer name |
| Quantity | Integer | Specifies quantity |
| Amount | Double | Specifies total amount |
| Mode\_of\_payment | Text | Specifies mode of payment |
| Cheque\_no | Text | Specifies cheque number |
| Account\_no | Text | Specifies account number |
| Account\_holder\_name | Text | Specifies account holder name |
| Amount\_paid | Double | Specifies amount paid |
| Balance\_amount | Double | Specifies balance amount |

**TABLE NAME:** Supplier

**PRIMARY KEY:** Supp\_id

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **TYPE** | **DESCRIPTION** |
| Supp\_id | Text | Specifies the Supplier ID |
| Supp\_name | Text | Specifies the Supplier name |
| House\_no | Integer | Specifies house number |
| Street | Text | Specifies street |
| Area | Number | Specifies area |
| City | Number | Specifies city |
| State | Text | Specifies state |
| Country | Text | Specifies country |
| Pincode | Long | Specifies pincode |
| Zipcode | Long | Specifies zipcode |
| Phone\_no1 | Text | Specifies phone number |
| Phone\_no2 | Text | Specifies alternate phone number |
| Lanline | Text | Specifies lanline number |
| Email | Text | Specifies email address |

**TABLE NAME:** Supplier stock

**PRIMARY KEY:** Supp\_id, Pitem\_no

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **TYPE** | **DESCRIPTION** |
| Supp\_id | Text | Specifies supplier id |
| Supp\_name | Text | Specifies supplier name |
| Pitem\_no | Text | Specifies purchase item number |
| Pitem\_name | Text | Specifies purchase item name |
| Rate | Double | Specifies rate |

**4.3 DATA FLOW DIAGRAM**

**CONTENT DIAGRAM**

****

**SALES**

**STOCK**

**SUPPLIER**

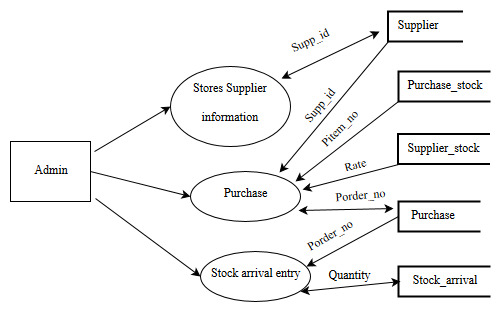
**PURCHASE STOCK**

**SALES**

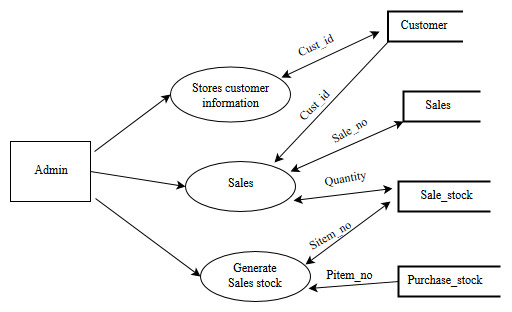
**PURCHASE**

**PRODUCTION PLAN**

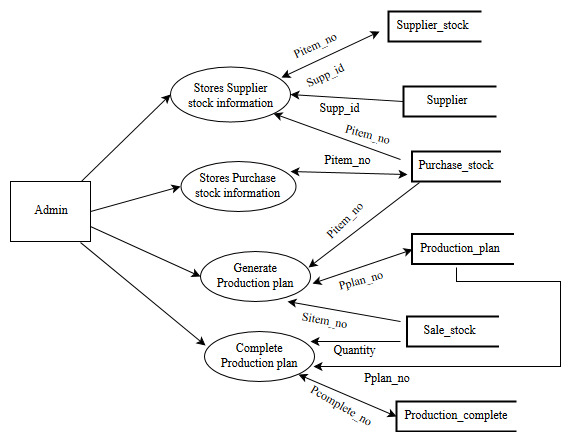
**PURCHASE PROCESS**

****

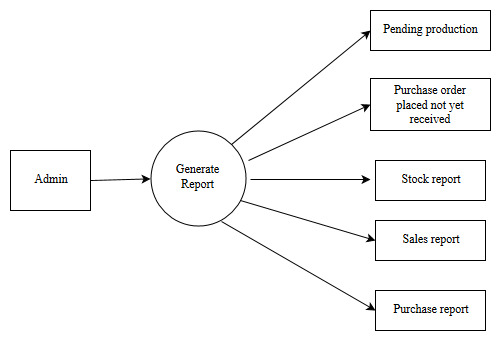
**SALES PROCESS**



**STOCK MAINTENANCE**

****

**REPORT**



**CHAPTER-V**

**STSTEM IMPLEMENTATION**

**5.1 CODING STANDARD**

Coding is the process of converting the design specification into source code. VB6.0 MS-Access has been used to implement this project. Although the time spent in coding will be comparatively less when the total system development time is considered this phase cannot be underestimated. Insufficient effort in the coding phase will affect both the testing and maintenance phase heavily.

Program should be written in easily readable and understandable way. Program’s readability is very important in bug fixing and also maintenance. Hence the developer should adopt a standard to ensure readability and easy maintenance of one’s programs. Throughout the coding phase these standards were strictly followed.

**5.2 NAMING CONVENTIONS**

**5.2.1 GENERAL CONVENTIONS**

* Hungarian notation is use in naming the controls.
* In – line comments are used to explain complicated code.
* Descriptive names are used all variables, procedure and constants.
* Multiword identifiers are internally distinguished by capital letters.
* Enough spacing is used to separate the statements have been followed to increase readability.
* The name of variable, function or procedures has been abbreviated if they are too long.

**5.2.2 VARIABLES**

For readability, the variables names area composed of number of words.

**5.2.3 METHODS**

The same conventions like that of variable have been followed for methods also.

**5.2.4 USER – DEFINED DATATYPES**

User – defined data types such as classes are composed of number of words. They start with a lower case letter. Each following word in the name begins with an upper case.

**5.2.5 COMMENTS**

Documentation of program using comments is an important tool that helps to make the code more readable. God readability ensures maintainability of the code and hence reduces the total development cost. Comments should hence be used throughout the code. They should be meaningful and redundant.

**5.3 IMPLEMENTATION**

The project is implemented in a user friendly manner to prove easy interface between the organization and subscriber. The database and the various table in it are implemented in Microsoft Access. The field properties are set to maintain the validity of the data. The required forms are designed in Microsoft Visual Basics 6.0 in elegant manner.

**5.3.1 METHOD USED**

The method used in this application is

**Validation:**

This method is used to check the user input and its data types. A control that should accept a numerical value should not take character values and vice versa. A control that accepts float values should not have more than one decimal point. Thus this module informs the user if any mistake is committed by them there by enabling them to avoid mistakes to a greater extent. This also helps to increase the efficiency of the application.

**CHAPTER – VI**

**SYSTEM TESTING**

**6.1 INTRODUCTOIN**

Testing plays a very critical role in quality assurance and helps in ensuring the reliability of the software. The purpose of testing a system is to identify various bugs and fix them. The software must be tested with respect to stated functional requirements and also with respect to interaction among the various modules. For discovering maximum errors, the developer must generate all test cases. The developed system is subjected to various strategies before it is brought into operation.

**6.2 UNIT TESTING**

The candidate system was subjected to this test and the result was verified. Unit testing focuses on testing the individual modules developed. Each module was checked for its consistency. The modules were also checked by giving some unexpected values for which the appropriate error messages were displays and entries were not accepted by the application.

**6.3 INTEGRATION TESTING**

The testing operation conducted after combining all the subsystem modules to check for the correctness of the output is integration testing. Even though a successful unit testing is performed, it is necessary to check the integration, since the integration links lead to erroneous results. This test was performed and the result proved to be consistent.

**6.4 ACCEPTANCE TESTING**

The acceptance testing involves planning and execution of functional tests and performance tests to demonstrate that the implement system satisfies its requirements. Acceptance tests are typically performed by the customer organization. This test was performed and the results were found satisfactory.

**CHAPTER-VII**

**CONCLUSION**

This system is flexible and any amendments can be made easily. This software package is designed for the specific need of the organization and is found to work effectively and efficiently. The major advantage of the system is faster, accurate information and reducing man power and time involved.

This tool is being implemented and is found to replace the existing manual system effectively. The tool is implemented in various systems and tested for inter-operability and scalability. The possibility of human error is fully minimized and bulk quantity of data can be maintained and processed with great ease.

Need for improvement has been realized and this new application has been developed in the best manner possible. The developed application has been tested with simple data and found to be consistent.

**CHAPTER – VIII**

**FUTURE ENHANCEMENTS**

There is always a room for improvements in any software package, however good and efficient it may be. But the improvement requires that the system should be flexible enough for further modification. Considering this important factor, the system is designed in such a way that further enhancements can be done without affecting the system presently developed.

Some of the possible future enhancements that can be done are

* This project can be enhanced to include the feature of contacting the organization and placing order through E – mail.
* Placing order through online and tracking its status.
* Online payment for purchase and sales.
* Reports can be generated in various formats to allow easy customization by the users as per their needs.

**APPENDIX- A**

**BIBLIOGRAPHY**

**REFERENCE BOOKS**

1. VB UNLEASHED

**-**Stephen Walther, First Indian Edition 2002

1. The Complete Reference VB

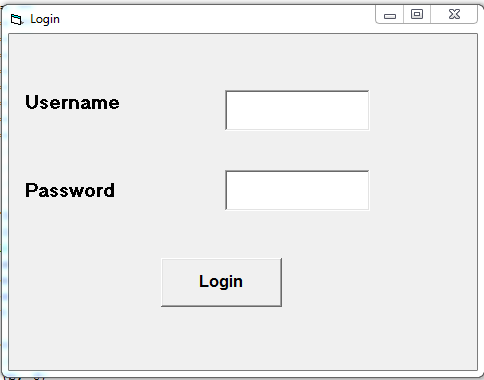
-Johan, Third Edition

**WEBSITES**

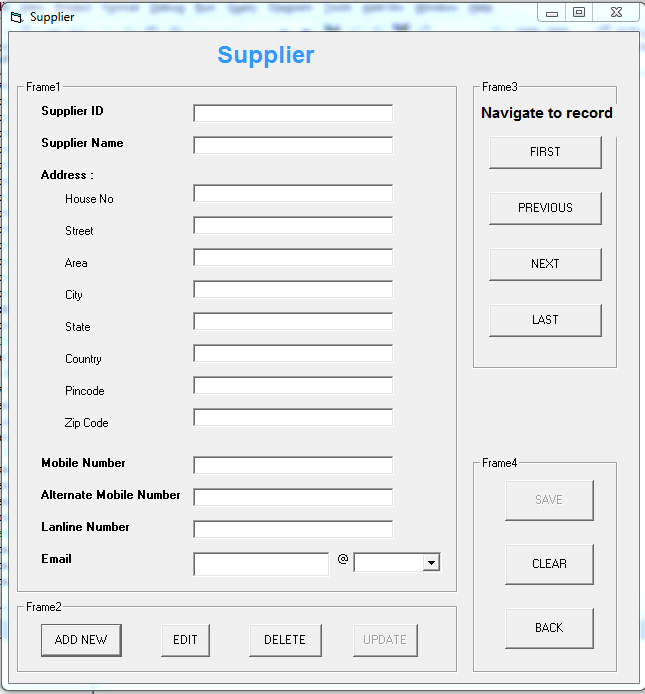
* <http://www.microsoft.com>
* <https://www.stackoverflow.com>
* <https://www.quora.com>

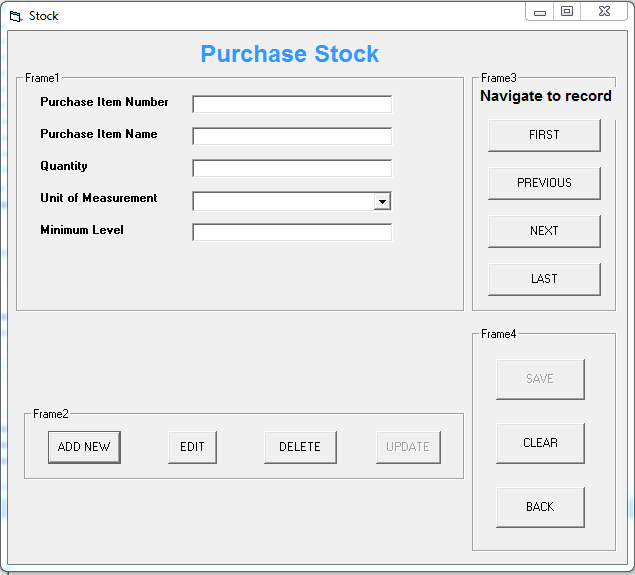
**APPENDIX- B**

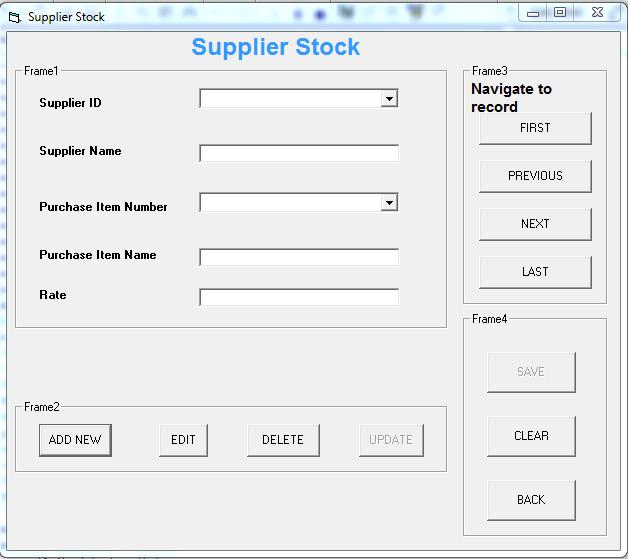
**INPUT DESIGN**

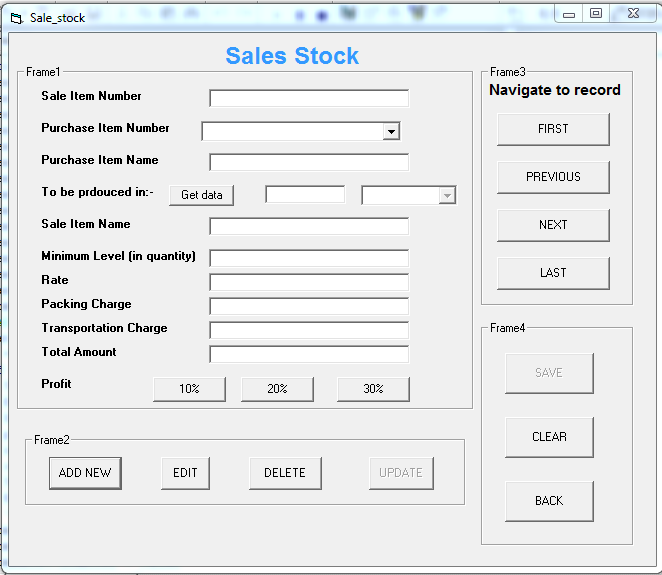




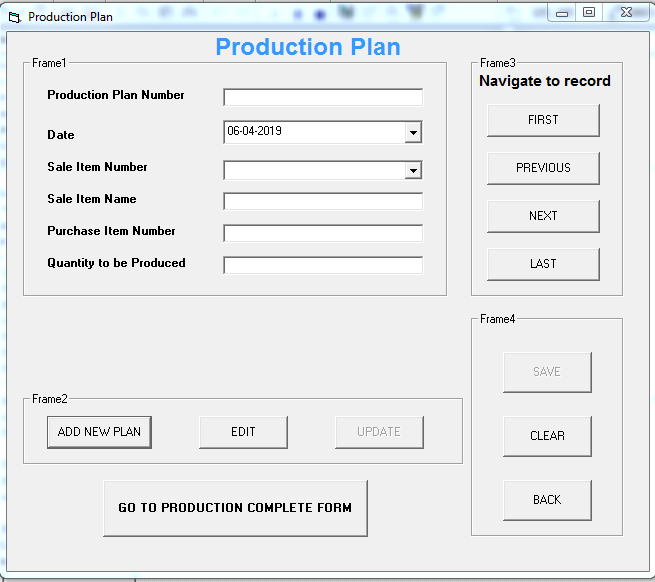


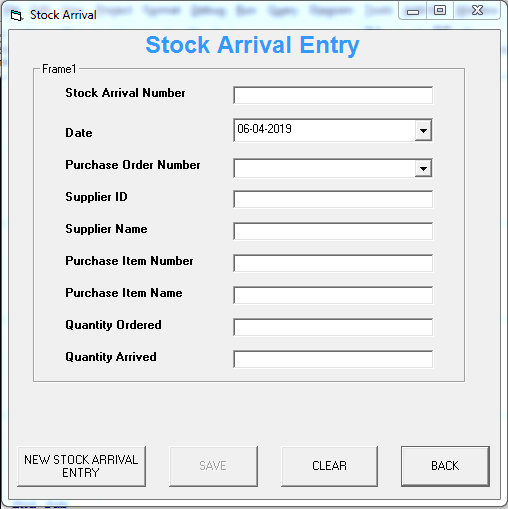
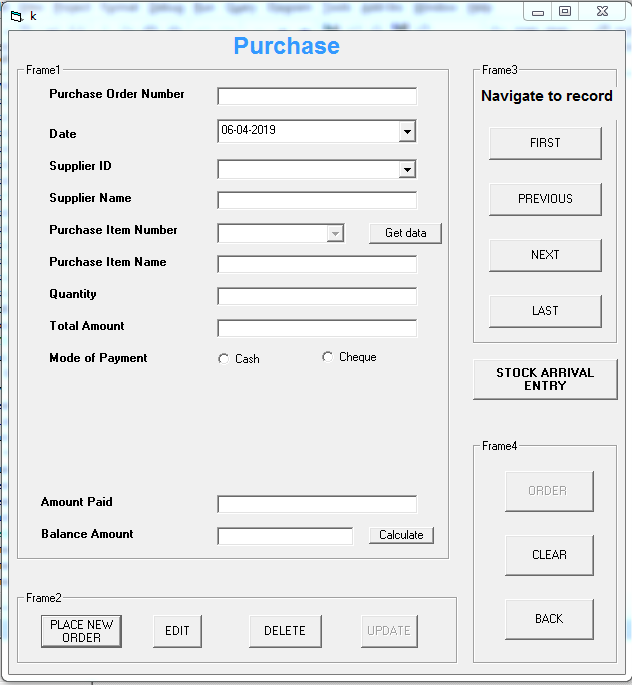


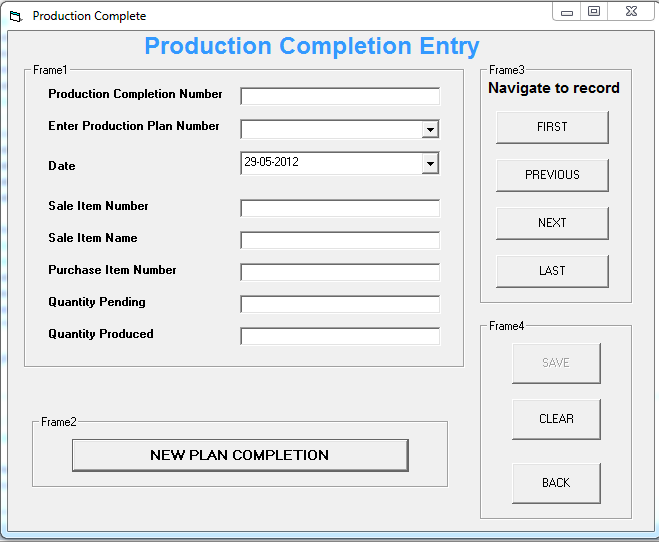


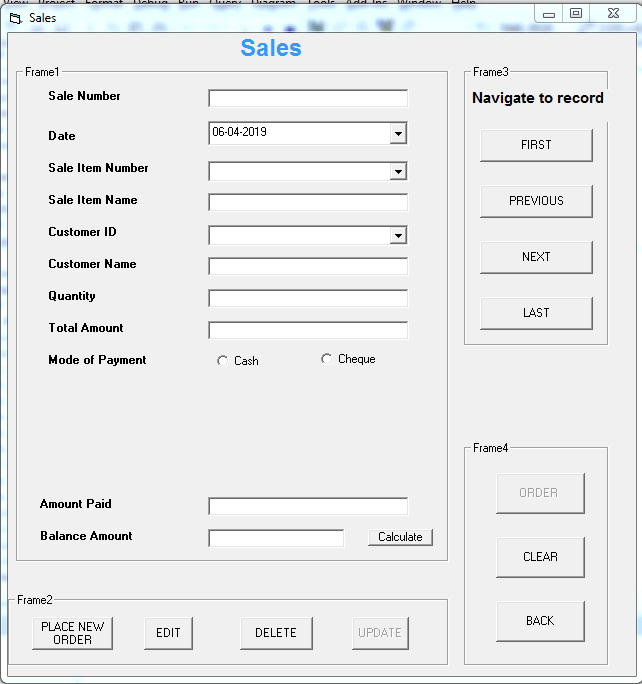










**APPENDIX- C**

**SAMPLE CODING**

**Private Sub Command1\_Click()**

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

DTPicker1.Value = Date

Combo1.ListIndex = -1

'AUTO GENERATE

Data1.Refresh

v = Data1.Recordset.RecordCount

If v = 0 Then

Text1.Text = "Pplan\_1"

Else

Data1.Recordset.MoveLast

p = Data1.Recordset.Fields(0)

num = Mid(p, 7)

res = num + 1

Text1.Text = "Pplan\_" & res

End If

Command12.Enabled = False

'edit,delete and update

Command2.Enabled = False

'Command3.Enabled = False

Command4.Enabled = False

'navigation frame

Frame3.Enabled = False

Command5.Enabled = False

Command6.Enabled = False

Command7.Enabled = False

Command8.Enabled = False

'SAVE

Command10.Enabled = True

**End Sub**

**Private Sub Command10\_Click()**

f = 0

z = 0

If Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Or Combo1.Text = "" Then

a = MsgBox("Please fill all the contents!", vbExclamation)

f = 1

ElseIf Not (DTPicker1.Value = Date) Then

MsgBox ("Enter today's date")

f = 1

ElseIf Val(Text4.Text) <= 0 Then

MsgBox ("Quantity cannot be less than 0")

Text4.Text = ""

f = 1

'to cal quantity to be deducted from purchase\_stock

ElseIf z = 0 Then

Set db1 = OpenDatabase("e:\\III year Project\PROJECT.MDB")

Set rs1 = db1.OpenRecordset("select sur\_name1, sur\_name2 from sale\_stock where sitem\_no = '" & Combo1.Text & "' ")

rs1.MoveFirst

'100

n = rs1.Fields(0).Value

'ml

uom = rs1.Fields(1).Value

'quantity to be produced

q = Val(Text4.Text)

Set rs1 = db1.OpenRecordset("select quantity,unit\_of\_measurement,stock\_in\_progress from purchase\_stock where pitem\_no = '" & Text3.Text & "' ")

rs1.MoveFirst

base\_quantity = rs1.Fields(0).Value

base\_uom = rs1.Fields(1).Value

sip = rs1.Fields(2).Value

If uom = "ml" Then

totalq = (n \* q) / 1000

ElseIf uom = "mg" Then

totalq = (n \* q) / 1000

ElseIf uom = "litres" Then

totalq = n \* q

ElseIf uom = "kilogram" Then

totalq = n \* q

End If

'if unit of measurement is ton in purchase stock

If base\_uom = "ton" Then

totalq = totalq / 1000

End If

'to check if stock is available

If (base\_quantity - totalq) > 0 Then

base\_quantity = base\_quantity - totalq

sip = sip + totalq

Else

MsgBox ("Raw material is not available for given quantity")

Text4.Text = ""

f = 1

End If

End If

If f = 0 Then

Data1.Refresh

Data1.Recordset.AddNew

Data1.Recordset.Fields(0) = Text1.Text

Data1.Recordset.Fields(1) = DTPicker1.Value

Data1.Recordset.Fields(2) = Combo1.Text

Data1.Recordset.Fields(3) = Text2.Text

Data1.Recordset.Fields(4) = Text3.Text

Data1.Recordset.Fields(5) = Val(Text4.Text)

Data1.Recordset.Fields(6) = "Pending"

Data1.Recordset.Fields(7) = Val(Text4.Text)

'to reduce quantity in purchase stock

Data2.Refresh

Data2.Recordset.MoveFirst

Do While Not Data2.Recordset.EOF

If (Text3.Text = Data2.Recordset.Fields(0)) Then

Data2.Recordset.Edit

Data2.Recordset.Fields(2) = base\_quantity

Data2.Recordset.Fields(5) = sip

Data2.Recordset.Update

Exit Do

Else

Data2.Recordset.MoveNext

End If

Loop

Data1.Recordset.Update

MsgBox ("Plan added successfully")

'Navigation frame

Frame3.Enabled = True

Command5.Enabled = True

Command6.Enabled = True

Command7.Enabled = True

Command8.Enabled = True

'edit,delete

Command2.Enabled = True

'Command3.Enabled = True

Command12.Enabled = True

Command10.Enabled = False

End If

**End Sub**

**Private Sub Command2\_Click()**

a = InputBox("Enter Production Plan Number")

flag = 0

Data1.Recordset.MoveFirst

Do While Not Data1.Recordset.EOF

If (LCase(a) = LCase(Data1.Recordset.Fields(0))) Then

If Data1.Recordset.Fields(6) = "Completed" Then

flag = 3

Exit Do

ElseIf Not Data1.Recordset.Fields(5) = Data1.Recordset.Fields(7) Then

flag = 4

Exit Do

Else

d = Data1.Recordset.Fields(1)

da = Date

If StrComp(d, da) = 0 Then

Text1.Text = Data1.Recordset.Fields(0)

DTPicker1.Value = Data1.Recordset.Fields(1)

Combo1.Text = Data1.Recordset.Fields(2)

Text2.Text = Data1.Recordset.Fields(3)

Text3.Text = Data1.Recordset.Fields(4)

DTPicker1.Enabled = False

Combo1.Enabled = False

Text2.Enabled = False

Text3.Enabled = False

Text4.Text = Data1.Recordset.Fields(5)

num = Data1.Recordset.Fields(5)

flag = 1

Exit Do

Else

flag = 2

Exit Do

End If

End If

Else

Data1.Recordset.MoveNext

End If

Loop

If flag = 1 Then

Data1.Recordset.Edit

Command4.Enabled = True

Frame3.Enabled = False

Frame4.Enabled = False

Command1.Enabled = False

'Command3.Enabled = False

Command12.Enabled = False

Command5.Enabled = False

Command6.Enabled = False

Command7.Enabled = False

Command8.Enabled = False

Command9.Enabled = False

Command10.Enabled = False

Command11.Enabled = False

ElseIf flag = 2 Then

MsgBox ("Enter today's Production plan number")

ElseIf flag = 3 Then

MsgBox ("This Production plan is completed")

ElseIf flag = 4 Then

MsgBox ("Production has started cannot edit")

Else

MsgBox ("Record not found")

End If

**End Sub**

**Private Sub Command10\_Click()**

z = 0

If Combo1.Text = "" Or Combo2.Text = "" Or Text4.Text = "" Or mop = 0 Or Text9.Text = "" Or Text10.Text = "" Then

a = MsgBox("Please fill all the contents!", vbExclamation)

z = 1

ElseIf Not (DTPicker1.Value = Date) Then

MsgBox ("Enter today's date")

z = 1

ElseIf Val(Text4.Text) <= 0 Then

MsgBox ("Quantity cannot be less than 0")

Text4.Text = ""

z = 1

ElseIf Val(Text9.Text) <= 0 Then

MsgBox ("Amount paid cannot be less than 0")

Text9.Text = ""

z = 1

ElseIf Option2.Value = True Then

If Text6.Text = "" Or Text7.Text = "" Or Text8.Text = "" Then

b = MsgBox("Please fill all the contents!", vbExclamation)

z = 1

End If

End If

If z = 0 Then

Data1.Refresh

Data1.Recordset.AddNew

Data1.Recordset.Fields(0) = Text1.Text

Data1.Recordset.Fields(1) = DTPicker1.Value

Data1.Recordset.Fields(2) = Combo1.Text

Data1.Recordset.Fields(3) = Text2.Text

Data1.Recordset.Fields(4) = Combo2.Text

Data1.Recordset.Fields(5) = Text3.Text

'quantity arrived

Data1.Recordset.Fields(6) = 0

' quantity ordered

Data1.Recordset.Fields(15) = Val(Text4.Text)

Data1.Recordset.Fields(7) = Val(Text5.Text)

If Option1.Value = True Then

Data1.Recordset.Fields(8) = "cash"

Data1.Recordset.Fields(9) = "nil"

Data1.Recordset.Fields(10) = "nil"

Data1.Recordset.Fields(11) = "nil"

End If

If Option2.Value = True Then

Data1.Recordset.Fields(8) = "cheque"

Data1.Recordset.Fields(9) = Text6.Text

Data1.Recordset.Fields(10) = Text7.Text

Data1.Recordset.Fields(11) = Text8.Text

End If

Data1.Recordset.Fields(12) = Val(Text9.Text)

Data1.Recordset.Fields(13) = Val(Text10.Text)

Data1.Recordset.Fields(14) = "Pending"

Data1.Recordset.Update

MsgBox ("Order placed successfully")

'Navigation frame

Frame3.Enabled = True

Command5.Enabled = True

Command6.Enabled = True

Command7.Enabled = True

Command8.Enabled = True

'edit,delete

Command2.Enabled = True

Command3.Enabled = True

'SELF

Command10.Enabled = False

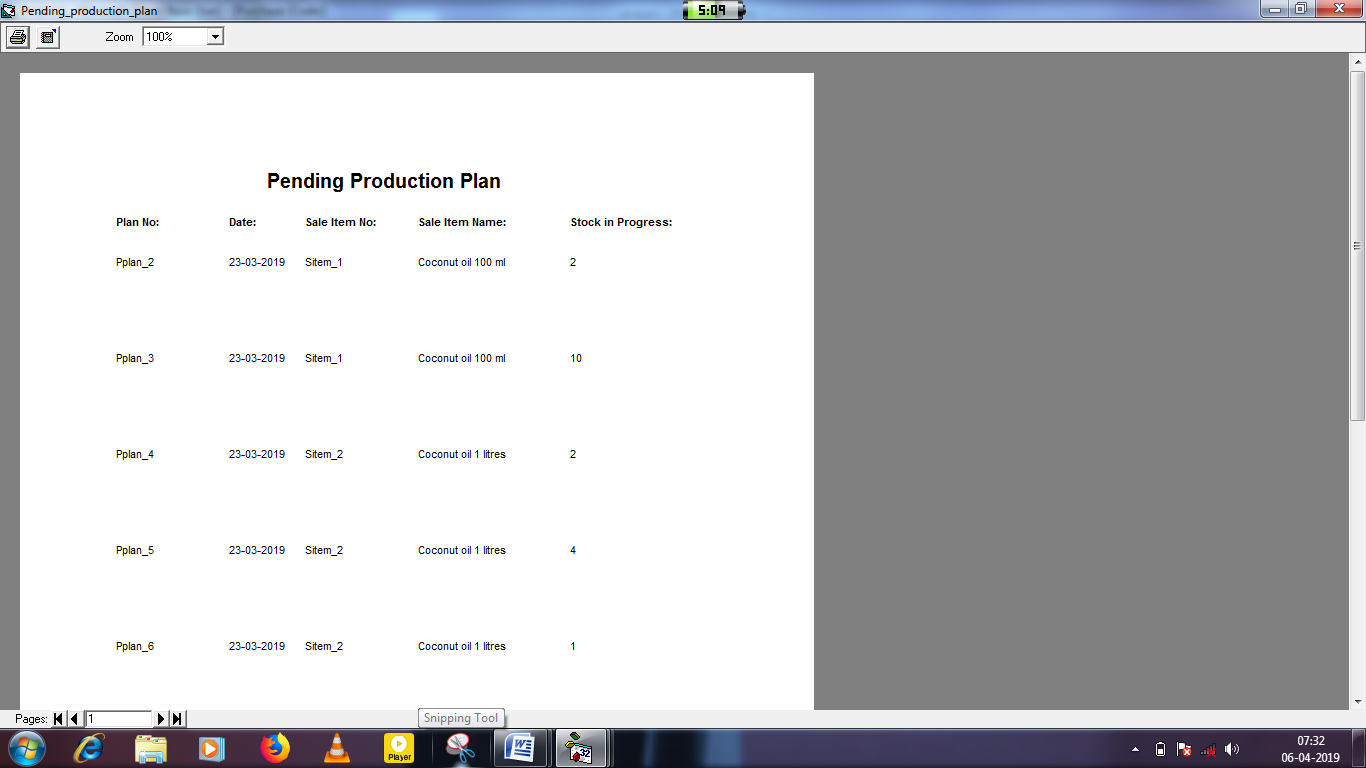
Command14.Enabled = True

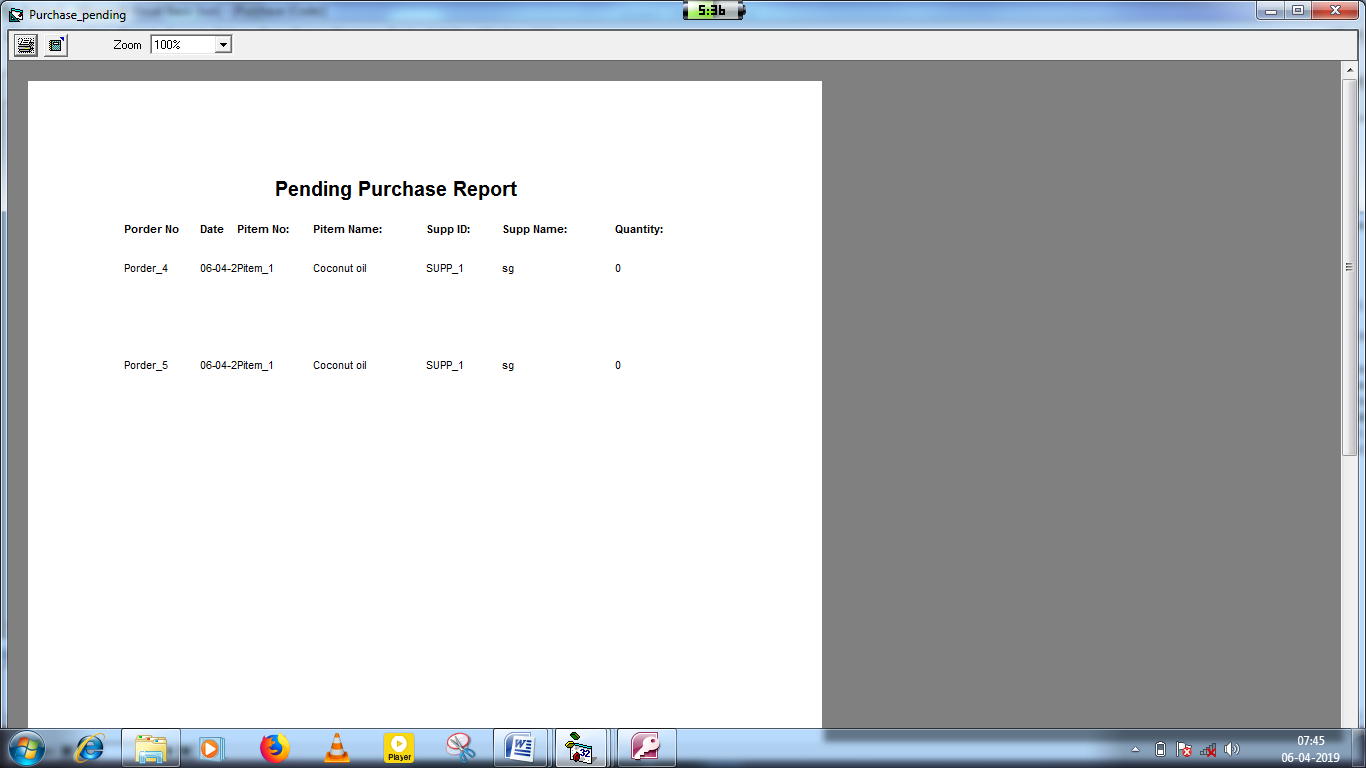
End If

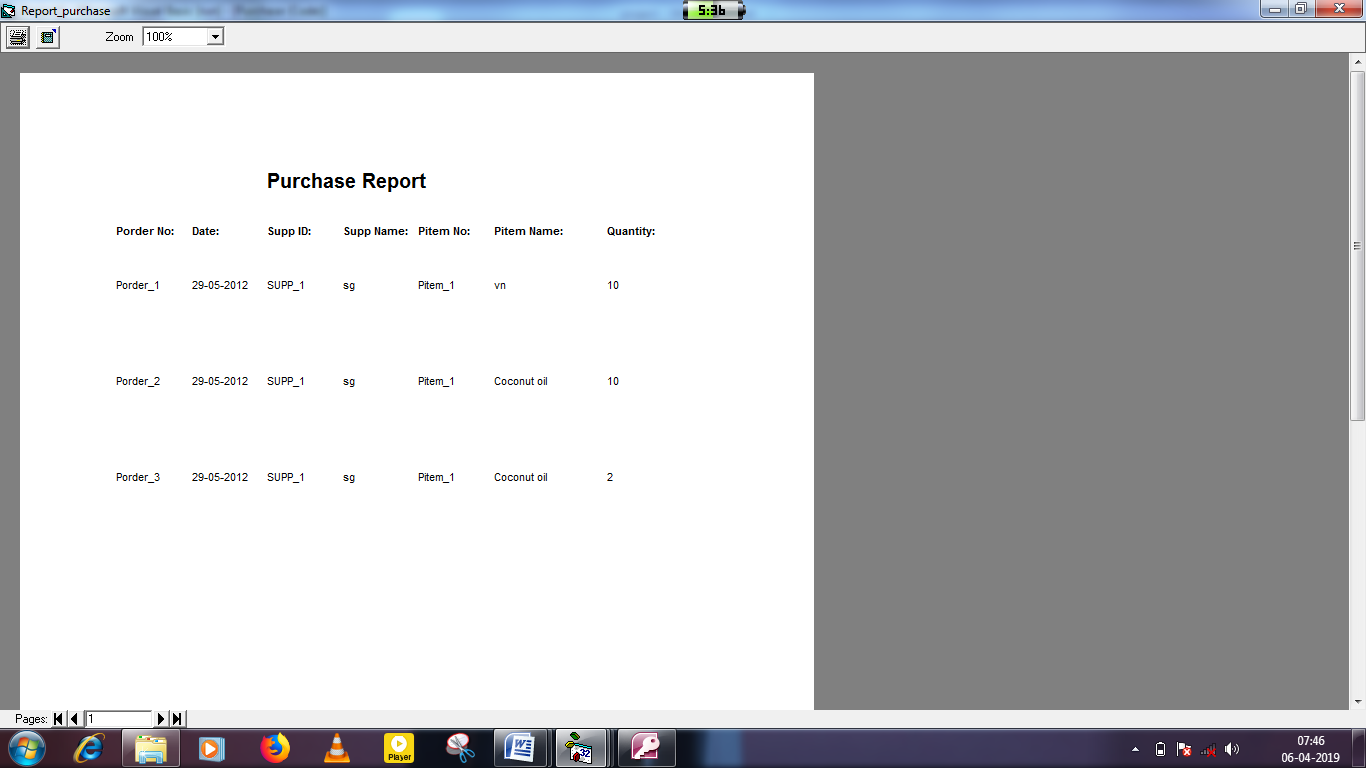
**End Sub**

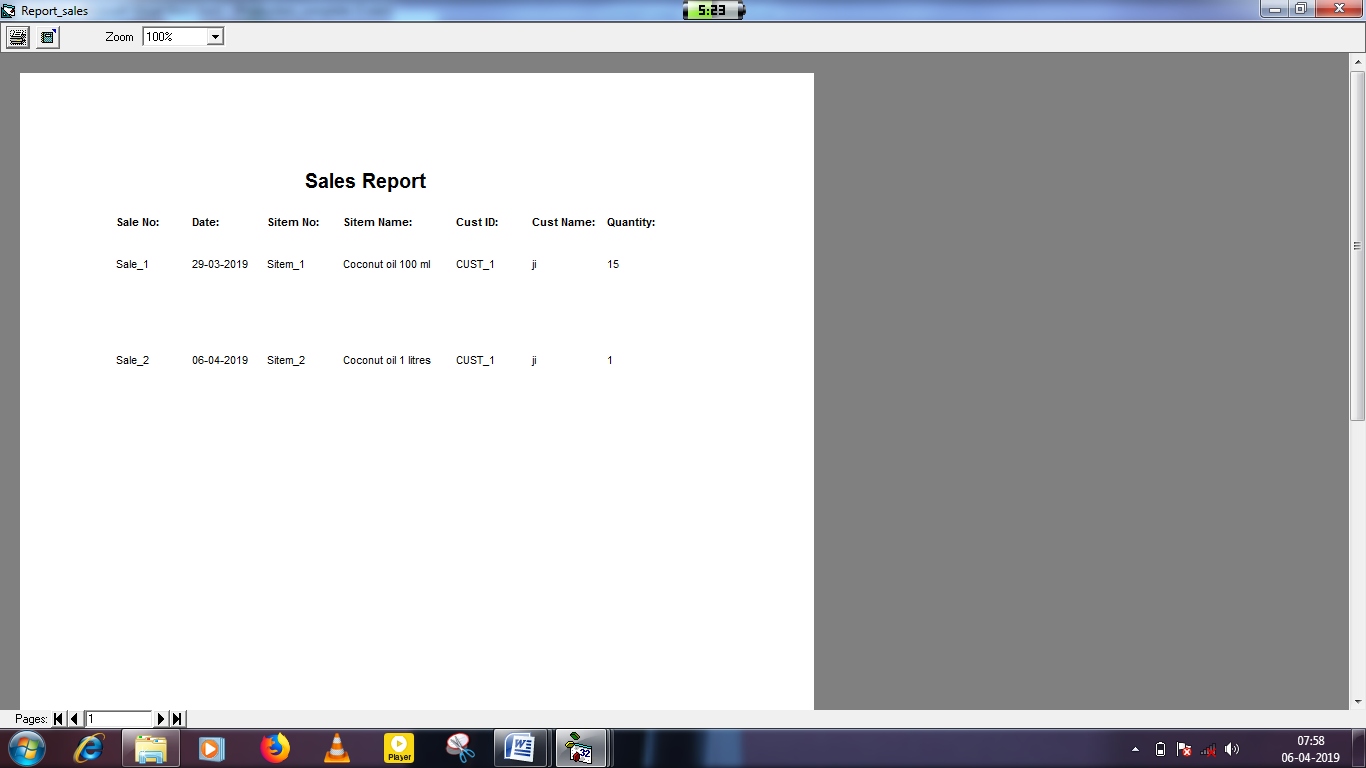
**APPENDIX- D**

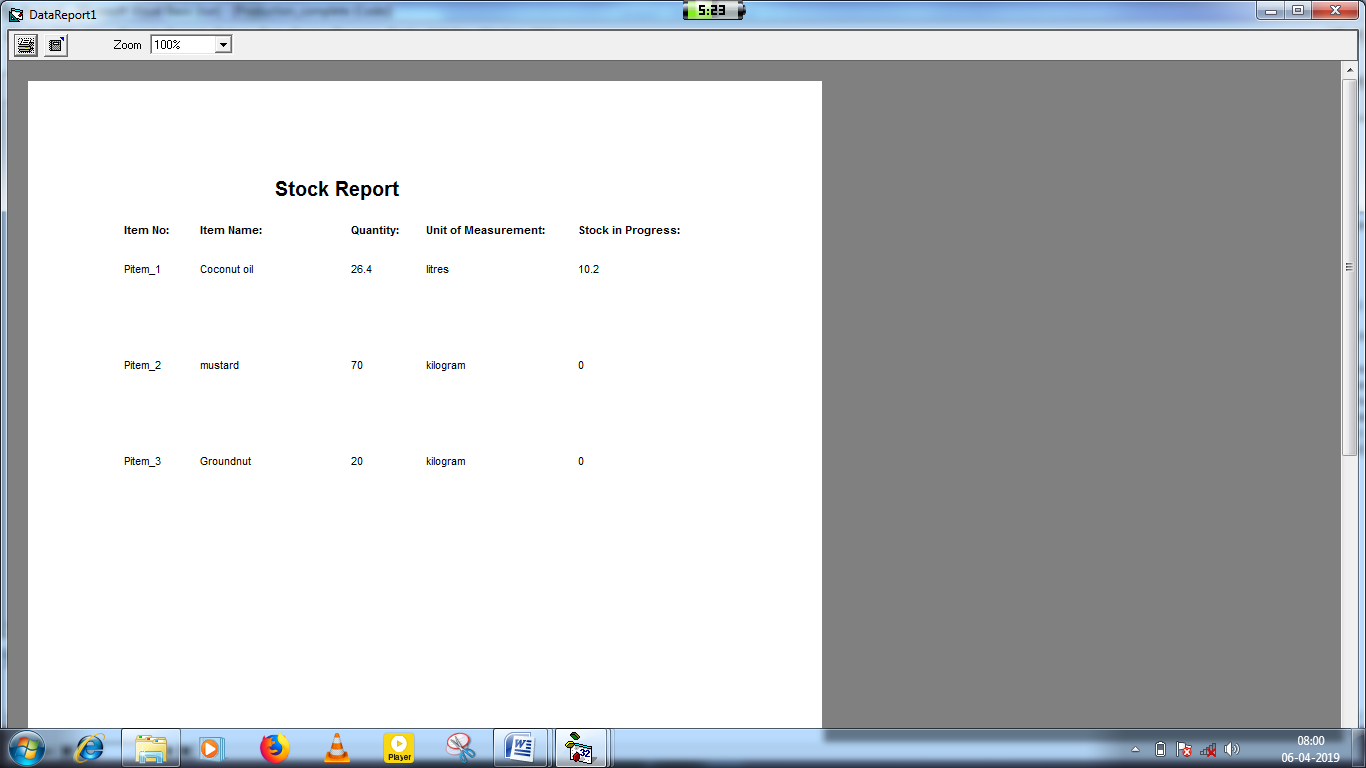
**OUTPUT**

****

****







**APPENDIX- E**

**REVIEW OF LITERATURE**

**The VB Framework**

The VB Framework is multi-language environment for building, deploying, and running web services and application.

Unified programming classes

The framework provides developers with a unified, object-oriented, hierarchical, and extensible set of class libraries (APIs). Currently, C++ developers use the Microsoft Foundation Classes and java developers use the Window Foundation Classes. The framework unifies these disparate models and gives Visual Basic and Jscript programmer’s access to class libraries as well. By creating a common set of APIs across all programming language, the common language runtime enables cross-language, error handling, and debugging. All programming languages, from Jscript to C++, have similar access to the framework and developers are free to choose the language that they want to use.

**MS-ACCESS**

**Client/server for MS-Access**

The MS-Access is N-tier model architecture. In N-tier one server maintains architecture the data storage and the business logic are handled by another server. The client server supports both multi- process database engine and single process multithreaded engines. The system with multi-process database engine supports multiple simultaneously users. Providing for data centralization on a network. It provide for scalability through the addition of more CPUs in physical machine in this system, the CPU runs multiple tasks by allotting time slices for each task. The Operating system assigns another task to CPU finishes one task the MS-Access supports the single process, multithread database engine. Instead of running distinct executable and running to each task, this application relies on multithreading work on single application. The benefit is substantially lower hardware requirements for a performance level.

The MS-Access process the query as explained below

MS-Access has some implicit assumptions here.

* The client is capable of formulating queries a network exists for transporting these query instruction to the server. The server is capable of using the network to the return the result of the query.
* DB library associates that string with a particular user connection opened at server.
* DB library process the String along the connection to the network library.
* MS-Access network library “disassembles: into a data packet for the network we are running. The data of the packet confirms to MS-Access formatting, which is called tabular data stream.
* The network delivers those packets to the MS-Access.
* MS-Access as a network library to which listens for queries to arrive the server side network library reassemble the query processer.
* MS-Access process the query and if it is a select statement generates a result set.
* MS-Access passes the result set to the server network library, which dissembles the result set into tabular data stream packets.
* The client side network library receives the packets and reassembles them into a result set, which resides on the client in a connection specific memory region.
* DB library API calls the memory region. Extracting the data held there and putting into the host language variable for display to the user.

**Features of MS-Access**

The proposed system needed an organized database equipment and history of maintenance events. This can only be fulfilled by a well-defined database system that could handle the size, desired. Complexity of the equipment and maintenance of the database to go for and external vendor supplied DBMS. To have an organized database and effectively controlled one, a RDBMS was suggested.

The company already has MS leading RDBMS MS-Access and proven systems having it as backend were running in the company itself. So the obvious choice was MS-Access. The following well describes the features of the RDBMS, there by justifying its selection.

The MS-Access is a relative DB management system that runs on WinNT OS with version 4.0 or above. In the client side a winNT or Win95 OD is enough to work on. The SQL client utilities or user interfaces through which a user, be an administrator or an end user, interact with the RDBMS.