K.T.S.P. MANDAL'S

K.M.C. COLLEGE, KHOPOLI

DEPARTMENT OF COMPUTER SCIENCE

KHOPOLI – 410203

A

PROJECT REPORT

ON

"PATHOLOGY LAB MANAGEMENT SYSTEM"

UNDER THE GUIDANCE OF

Mrs.Nilam Patil

SUBMITTED TO

UNIVERSITY OF MUMBAI

BY

Mrs. Neha Narendra Ghonge

T.Y.B.Sc (COMPUTER SCIENCE)

2019-2020

• ACKNOWLEDGEMENT

It gives me great pleasure to present my project on, “**PathologyLabManagementSystem”.**

This is my first milestone in B.Sc. Computer Science. I would like to thank our **Prof.Mr.P.P.Wadkar(HOD of ComputerScience**),who helped throughout the project.

I would like to express my sincere gratitude to all the professors who helped me in project. I would also like to acknowledge the help and guidance of **Prof.Miss.NilamPatil** for acknowledging the help & guidance provided by them for project in all the places during the presentation of the project .I would also extend my to our principle Dr.**Mr**.**Khanvilkar Sir** for his support &facilities provided to us for the same.

Onward my project work ,I am also grateful to the staff member of computer department for their moral support & application shown towards my project.

Mrs. Neha Narendra Ghonge

T.Y.BSc(COMPUTER SCIENCE)

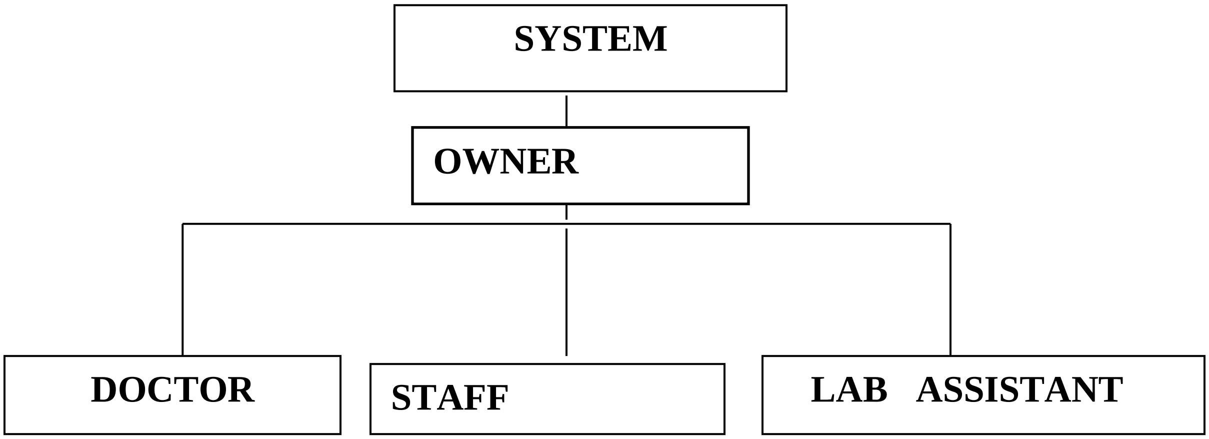
INDEX

• *INDEX*

`

PRELIMINARY INVESTIGATION

Organization Chart:-



❖ History

Pathology lab System:

The Pathology lab was build in the year of 1989 by the Dr.Narayan Swami in Badlapur.

Pathology is the precise study and diagnosis of disease. The word pathology Ancient Greek πάθος, pathos which may be translated into English as either “Experience”or “Suffering”,and λογἰα,logia,”an account of” or “the study of ” Pathologization,to pathologine,e.g.Pathological gambling. Pathology (or pathos) is synonymous with diseases.The suffix “path” is used to indicate a state of diseases(e.g. Cardiomyopathy).

A physician practicing pathology is called a pathologist.

Description Of System

PATHOLOGY LAB MANAGEMENT SYSTEM is the system focuses the data of patient ,Doctor and Supplier.Every supplier&Patient will be given a unique id and particular class will be allocated to them.

Depending on that id and Supplier can confirm their information they require,each supplier attendance will also be saved in the database.

All records are maintained properly in the system,so there is not chance in making a mistake.therefore,if it is easy to search any records about the patient.At the end of every year,the admin can create a report of the patient details or the order of supplier at end of year.so,we save lot of timeand money.this system provides all the work done faster. Due to this,there will be a satisfaction to the Supplier.

This system saves the man power too,because this is very comfortable and makes it very easy. This system gives a lot profit and goodwill to the pathology lab.

❖ Limitation

- All the work done manually was very difficult and takes a lot of time.
- Working of the current system is a lengthy process. lot of redundancy in maintaining supplier records.
- Generations of the correct report are not guaranteed.
- All records are maintained in the registers, so there will be
- Also there is no security because all records are maintained in register.
- Modification of one record to be modified.so work becomes very critical,so some time it may results into data loss.
- Database management is done manually.
- More time is required for searching of records or patient into the registers.

• Proposed System

In Proposed system we can store all details in one machine. All The records can be maintained properly in this system,so there is not chance in making a mistake,therefore it is easy to search any records about the patient.

At the end of every year,the admin can create report of the patient details or the order of supplier at end of year .So,we save the money.

This system provides all the work done faster .Due to this,there will be a satisfaction to Supplier.This system saves the manpower too,because thisIs very comfortable and makes it very easy.

If we want any monthly or weekly reports for calculation, so,we cantake it in very few seconds because it’s very fast process.

At the end of year we can Fields records for showing the results of each supplier&patient.so,we can save a lot of time and money.In this system we can provide all services fast for the supplier.It is very safe way to store the data.

This system saves the man power too because this is a very comfortable and peaceful system.This company and goodwill to the supplier.

♦ Advantages

- It facilitates quick processing of data.
- The best advantage of the system is that it saves the time.
- Data/ records present in the database is highly secure.
- Searching of data is done very easily and more efficiently
- There is no fear of loss of data or data corruption.
- Information of any customer can be retrieved any time.
- This system also provides a facility to modify the records.
- As the information of each Supplier & Patient is maintained in the computers the user will not have to maintain register or do paper work.
- Report is generated automatically.

Feasibility Study

It is an important part of the any system developing life cycle of preliminary investigation because only feasible projects go to development stages.

1)Technical feasibility:

Technical feasibility raises the questions like
a)Is it possible that the work can be done with current equipments, software technology and person?
b)Is new technology required, what is the possibilities that it can be developed?

In case of our project, the software which we have built up fully support current windows OS but it lacks the support of other environment OS. It is not depended on the large number of user. So, it can handle a very large number of user's environment.The support for the hardware:

It has full support for new hardware. So no hardware compatibility issues arise as it requires minimum configuration.

2)Economic feasibility:

It deals with economical impact of the system on the environment it is used i.e. benefit in creating the systems. And the project is economical feasible.

The system development cost will be significant so the proposed system is economically feasible.

3)Operational Feasibility:

As the user is familiar with the window environment and the system is developed in vb.net there is no need of special training for operational system. Hence the system is operationally feasible.

❖ STAKEHOLDERS

- Stakeholders are those who have interest in the Successful implementation of the System.
- Stakeholder can be the administrative people who have all the authority of System.
- Stakeholders can be the management peoples. They have less authority than Administrative person. They maintain all the information about System.

- **Stake Holders**

- Doctor
- Supplier
- Patient

♦ Software Requirements:-

- **Operating System: -** Windows XP/2010.

Front End:-

- VB.NET 2008
- Dot Net FrameWork 3.5

Back End:-

- SQL server 2008
- Visual Basic 2019

Hardware Requirements:-

- **C.P.U:-** Intel Core i5.
- **R.A.M:-** 4 Giga Bytes.
- **Hard Disk:-** 40 Giga Bytes.
- **Type of System:-** Single User

GANTT CHART

Task Name	Date	July					August					September					October					November					Sign	Remark
		w1	w2	w3	w4	w5	w1	w2	w3	w4	w5	w1	w2	w3	w4	w5	w1	w2	w3	w4	w5	w1	w2	w3	w4	w5		
1. Project Searches																												
1.1 Planning the Idea		■	■																									
1.2 Determine Scope			■																									
2. Feasibility Phase																												
2.1Technical Feasibility				■																								
2.2 Economic Feasibility				■																								
2.3 Application Feasibility				■																								
2.4 Operational Feasibility					■																							
3. Integration of system requirements																												
3.1 Write stmt of need					■																							
3.2 Develop Event Table, Use case Diagram						■																						
4. Data and Process Modeling																												
4.1 Define Project Data								■																				
4.2 Identify process work								■																				
4.3 Organize the data									■																			
5. Project Scheduling																												
5.1 Estimate resource Duration procedure										■																		
5.2 Develop Gantt Chart										■																		
6.System Design																												
6.1 Develop Page Layout											■																	
7. Program Coding																												
7.1 Write a code												■																
8. Unit Testing																												
8.1 Check particular function														■														
8.2 Code Modules														■														
9. System Integration															■													
9.1 Integrating Component of design																■												
9.2 Integrating Coding to run																	■											
10. System Implementation																												
10.1 Run the data to give it																			■									
11. Acceptance Testing																												
11.1 Check unit of code																					■							
11.2 Check Well Design Architecture																						■						

K.M.C. College Khopoli , Department of Computer Science Page[ ]

SYSTEM ANALYSIS

♦ Questionnaires

A questionnaire is a research instrument consisting of series of questions and other prompts for the purpose of gathering information from respondents. Although they are often designed for statistical analysis of the responses, this is not always the case. The questionnaire was invented by Sir Francis Galton.

Questionnaires have advantages over some other types of surveys in that they are cheap, do not require as much effort from the questioner as verbal or telephone surveys, and often have standardized answers that it make simple to compile data. However, such standardized answers may frustrate users. Questionnaires are also sharply limited by the fact that respondents must be able to read the questions and respond to them. Thus, for some demographic groups connecting a survey by questionnaire may not be practical.

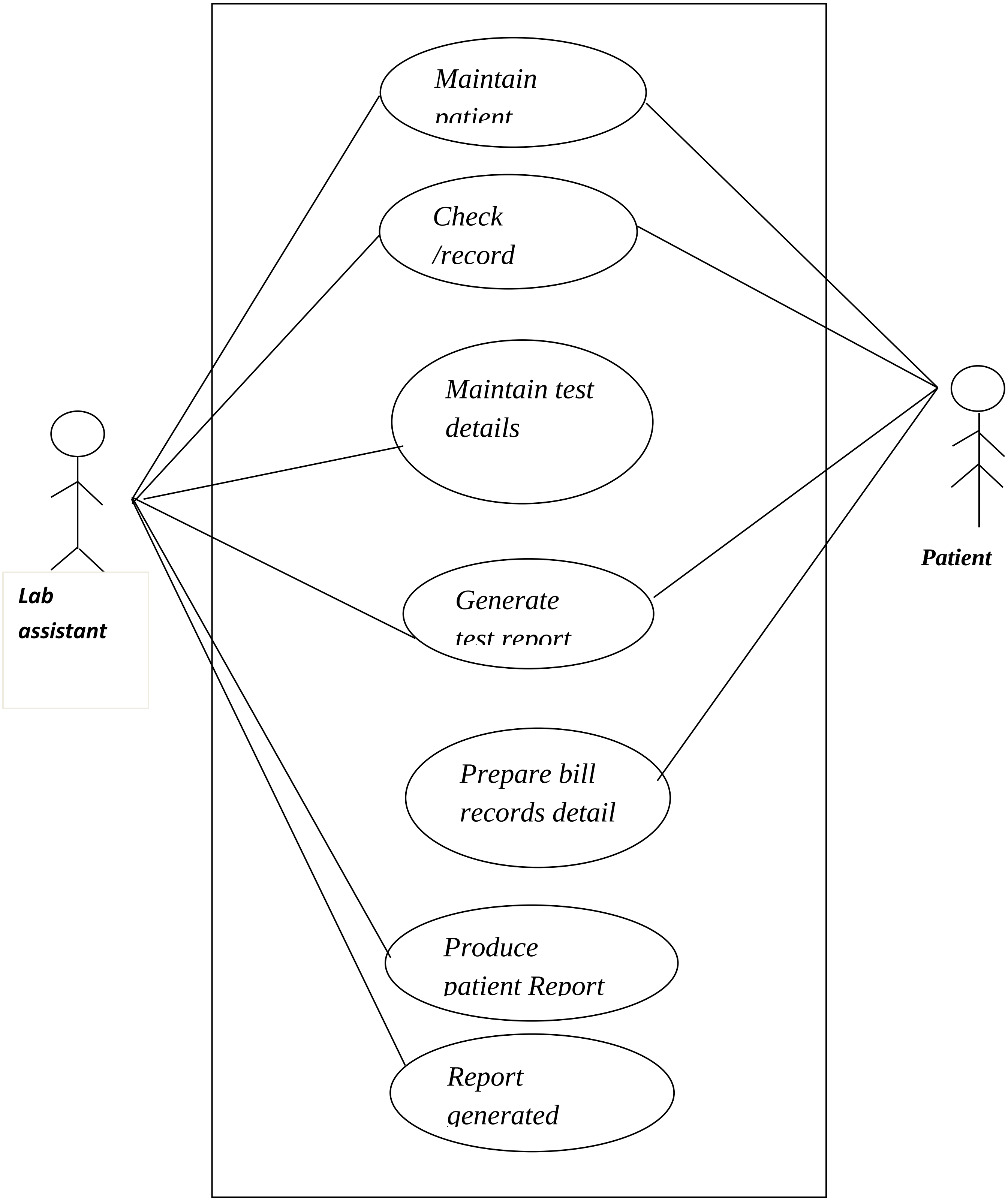
♦ Questions:

- How will the system work?
- What output will we get?
- Who is going to handle system?
- What the system will print?
- Is anyone can use this system?
- Is output from the system is correct or not?
- Is it expensive for to use?
- Is any training taken by the user to use this system?

EventTable

Event	*Triggers*	*Source*	*Activity*	*Response*	*Destination*
Lab assistant maintain patient details.	*Patient visit.*	*Lab assistant.*	*Maintain patient details*	*Patient details*	*Lab assistant.*
Lab assistant check reference of doctor.	*Doctor Reference*	*Doctor*	*Check/ record reference details*	*referencedetails*	*Lab assistant*
Lab assistant maintain test details	*Test details*	*Lab assistant*	*Maintain test details*	*Test details*	*Patient.*
Patient Ask test/report details	*Test report request*	*Patient*	*Generate test report*	*Test report generate*	*Patient.*
Patient pay bill.	*Payment request*	*patient*	*Prepare bill/record details*	*Bill details*	*Patient.*
Time to produce patient summary report.	*End of the week*	*Lab assistant*	*Produce patient summary report*	*Patient summary report produce.*	*Lab assistant*
Test details report	*End of week*	*Lab assistant*	*Produce patient summary report*	*Patient summary report produce.*	*Lab assistant*

UseCaseDiagram



Scenario:

1) **Student subsystem:**

In this subsystem, two actors are Supplier and Admin

Processes includes are:

i. Enter Supplier details
ii. Update Supplier details
iii. Look up for Supplier details
iv. Delete Supplier details
v. Display Supplier details
vi. Pays fees.

2)**Patient subsystem:**

In this subsystem two actor is present patient and Admin
Processes includes are:

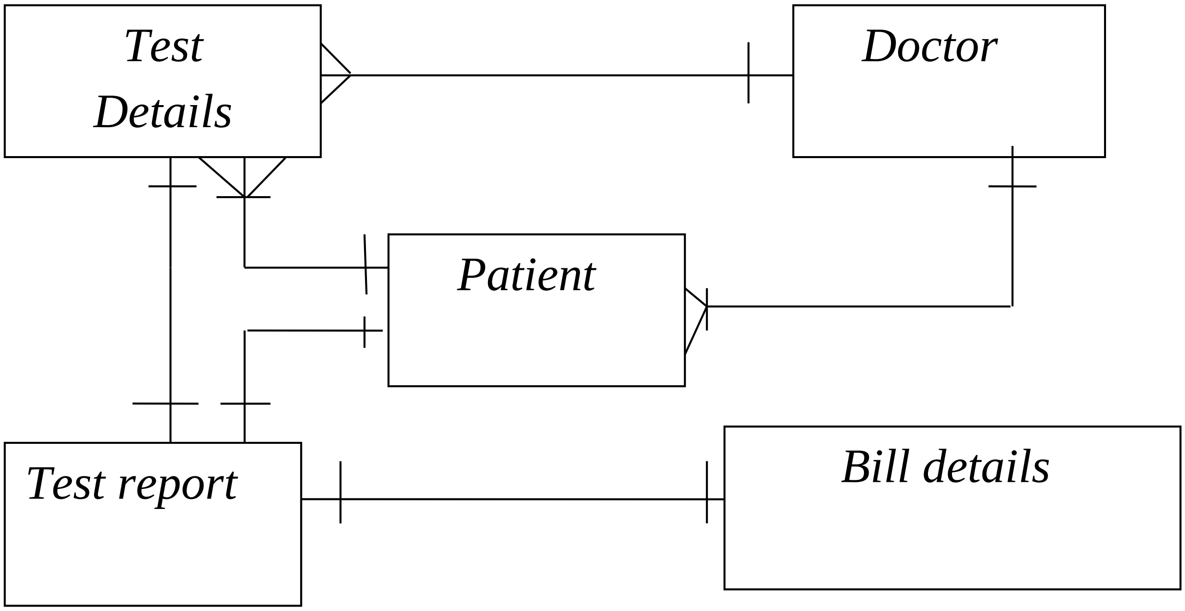
i. Enter patient details.
ii. Update patient details.
iii. Look for patient details
iv. Delete patient details
v. Display patient details
vi. Pay's Fees

3)**Staff subsystem:**

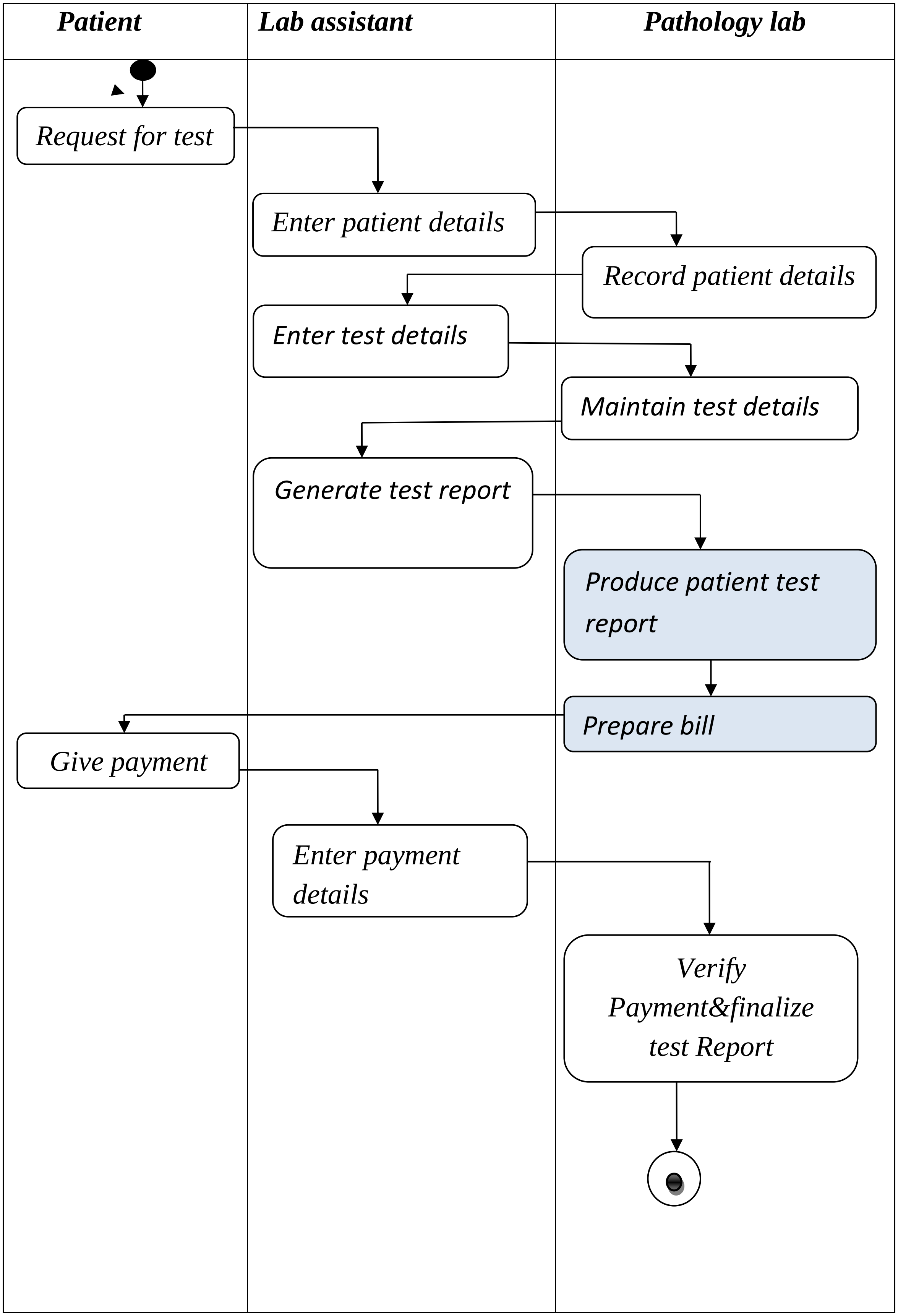
In this subsystem two actors are present i.e. Staff and Admin
Processes includes are:

i. Enter staff details
ii. Update staff details
iii. Look up for staff details
iv. Delete staff details
v. Pay salary to staff.
vi.

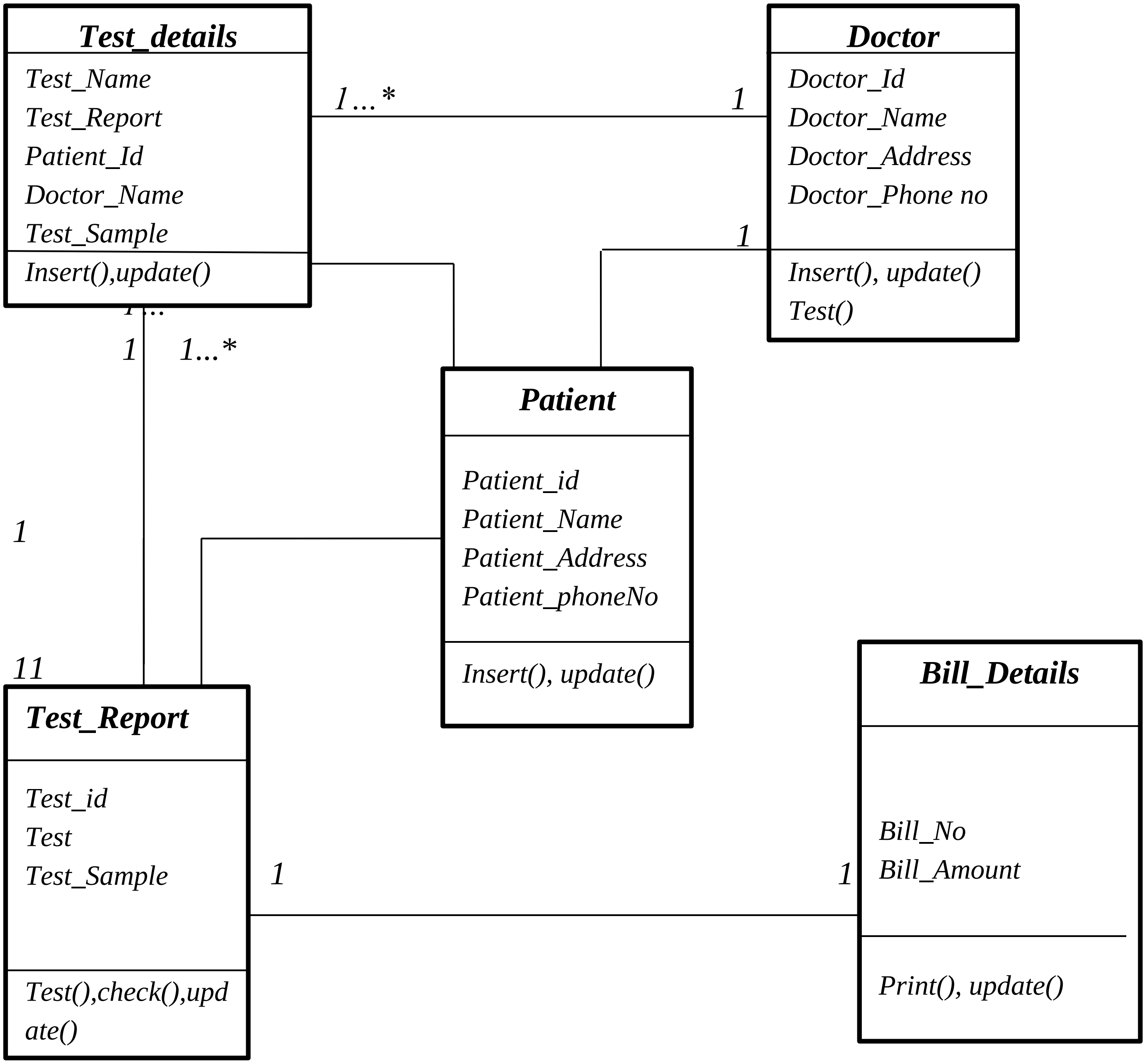
Entity Relationship Diagram(E.R.D)



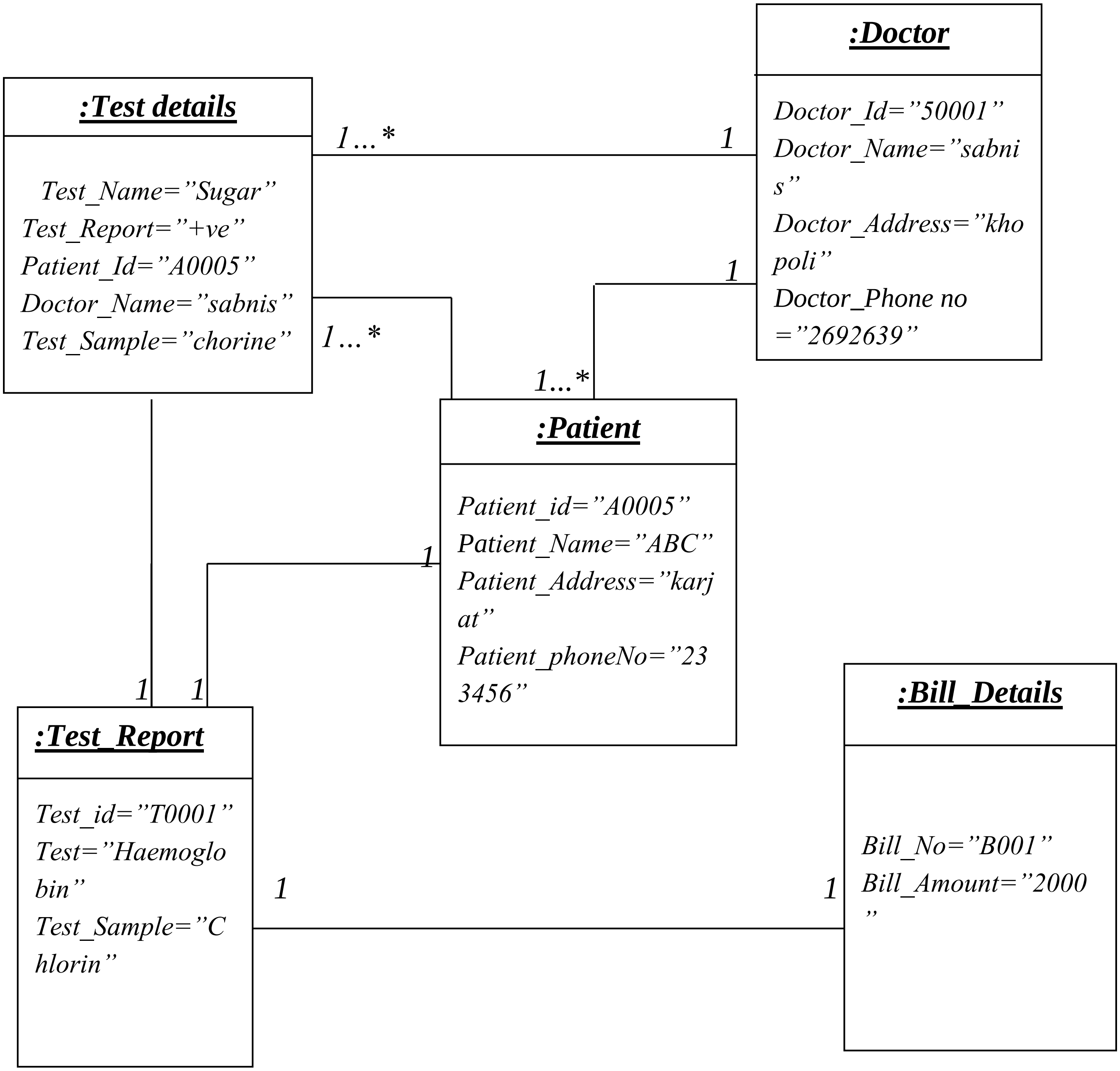
ActivityDiagram



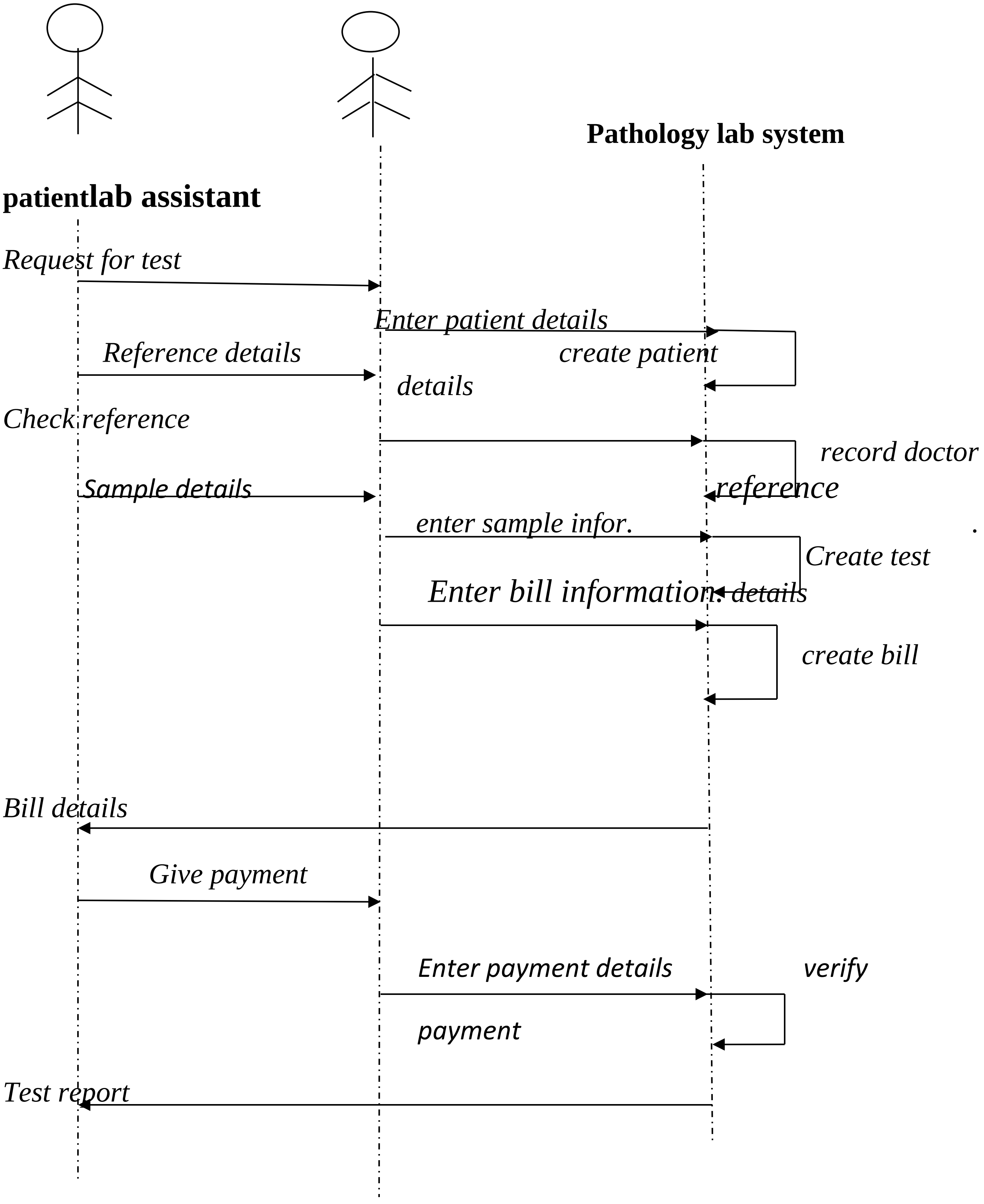
ClassDiagram



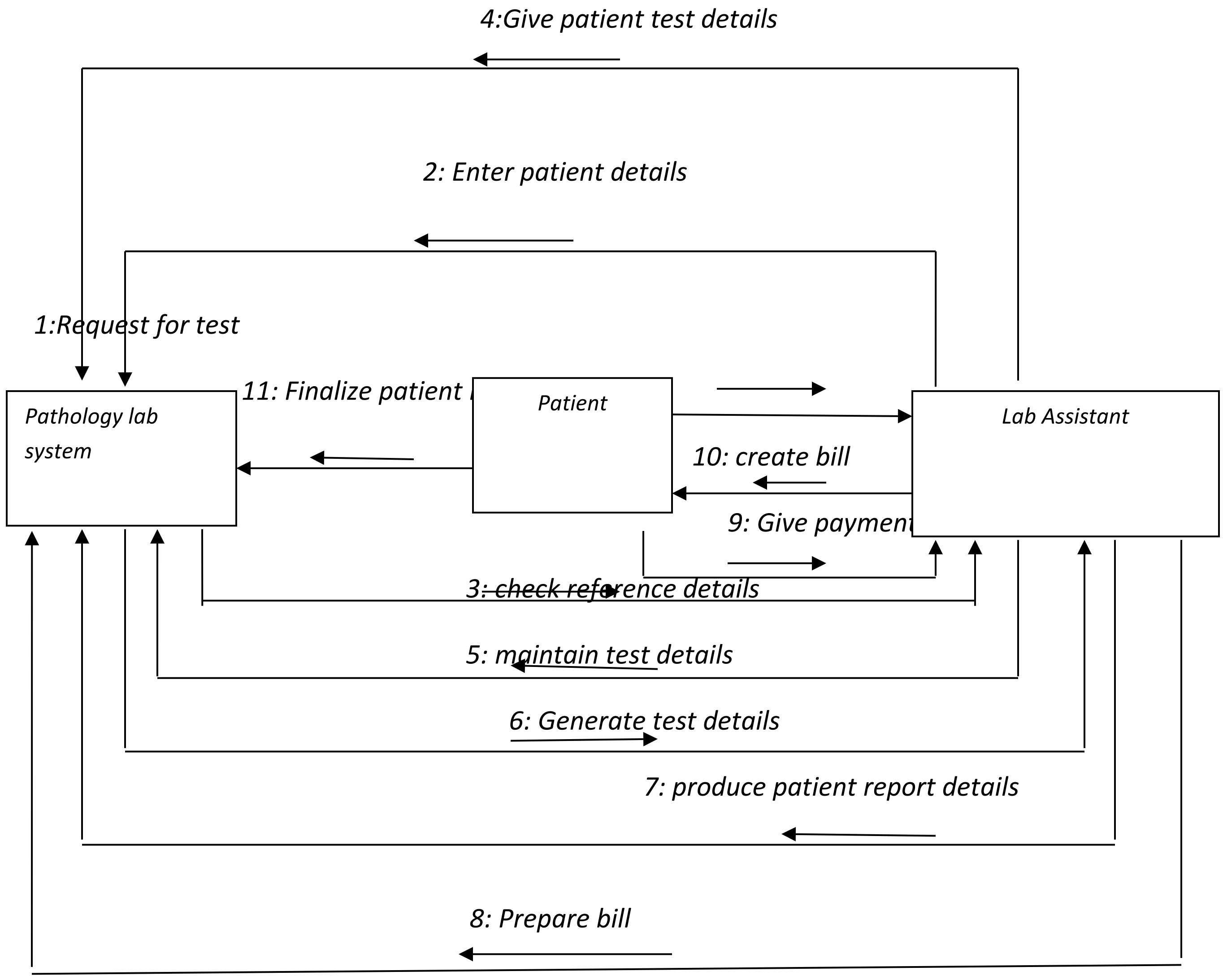
Object Diagram:



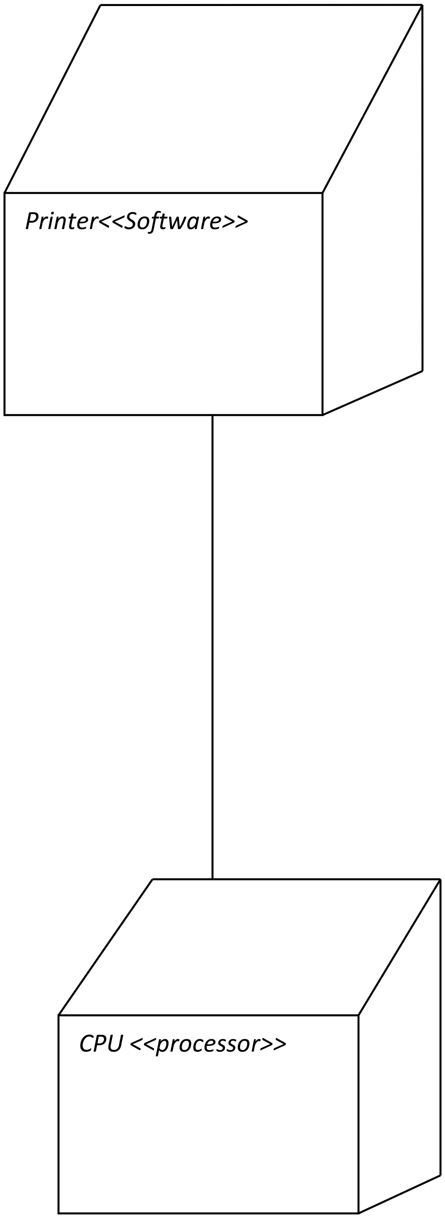
• SequenceDiagram



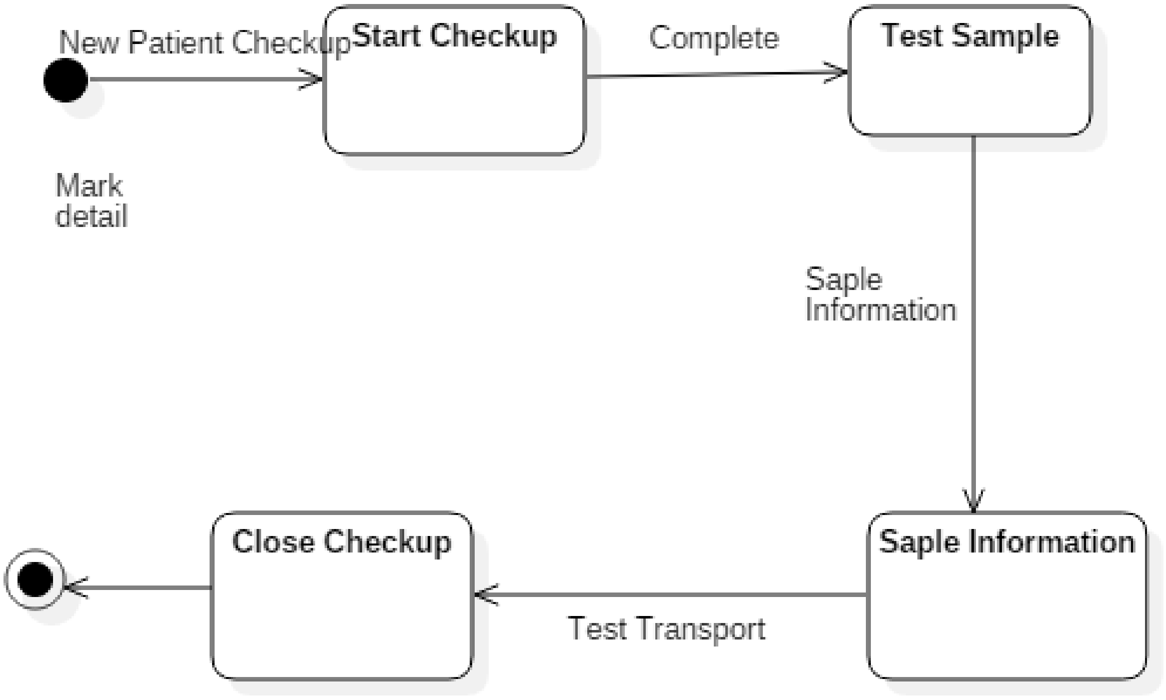
COLLABORATION DIAGRAM



DEPLOYMENT DIAGRAM



STATE DIAGRAM



System Design

ConvertingERDtoTables

1] Doctor details:-

FIELD_NAME	*DATA TYPE*	*FIELD_SIZE*	*DESCRIPTION*
Doctor _ID	*Numeric*	*50*	*It used to store Doctor id*
Doctor_Name	*Text*	*50*	*It used to store Doctor Name.*
Doctor _Address	*Text*	*50*	*It used to store Address.*
Doctor _Phone no	*Numeric*	*50*	*It used to store Doctor phone no.*

2] Patient details:-

FIELD NAME	*DATA TYPE*	*DESCRIPTION*
Patient _ID	*Number*	*It used to store Patient ID.*
Patient _Name	*Text*	*It used to store Patient name.*
Patient _Address	*Text*	*It used to store Patient address.*
Patient _Phone no	*Number*	*It used to store Patient phone no.*

3] Test Details:-

FIELD NAME	*DATA TYPE*	*DESCRIPTION*
Test _Name	*Text*	*It used to store Test name.*
Test _Report	*Text*	*It used to store Test report.*
Patient _Id	*Number*	*It used to store Patient id.*
Test _Sample	*Text*	*It used to store test sample.*

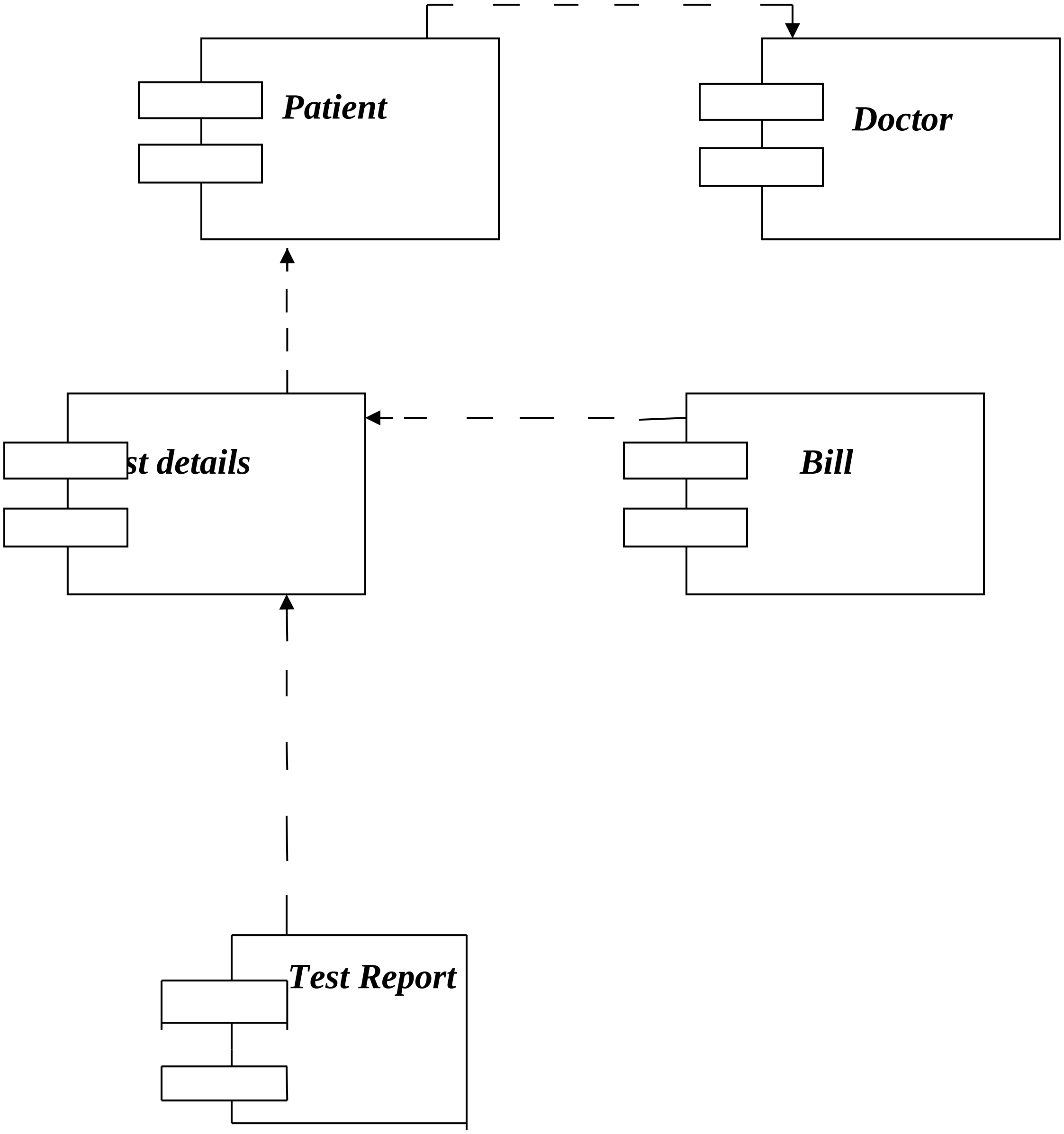
4] Bill Details:-

FIELD NAME	*DATA TYPE*	*DESCRIPTION*
Bill _No	*Number*	*It used to store Bill no.*
Bill _Amount	*Number*	*It used to store Biull amount.*

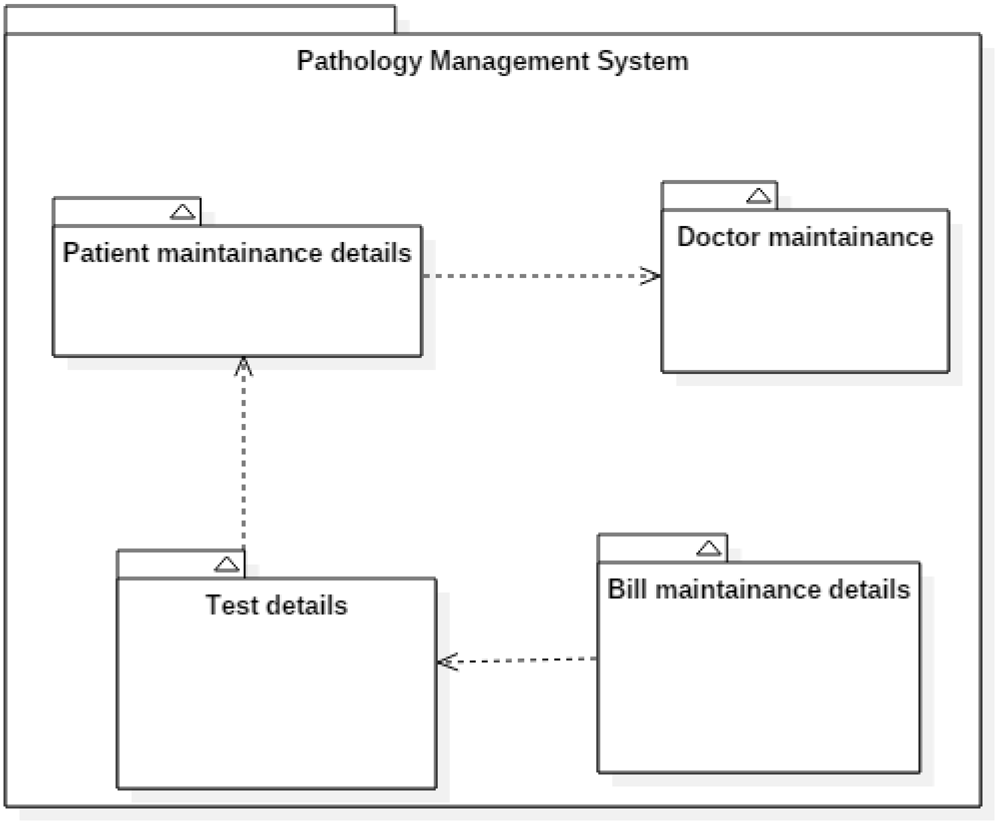
5] Test Report:-

FIELD NAME	*DATA TYPE*	*DESCRIPTION*
Test _Id	*Number*	*It used to store Test id r.*
Test	*Text*	*It used to store Test.*
Test _Sample	*Text*	*It used to store Test sample.*

ComponentDiagram

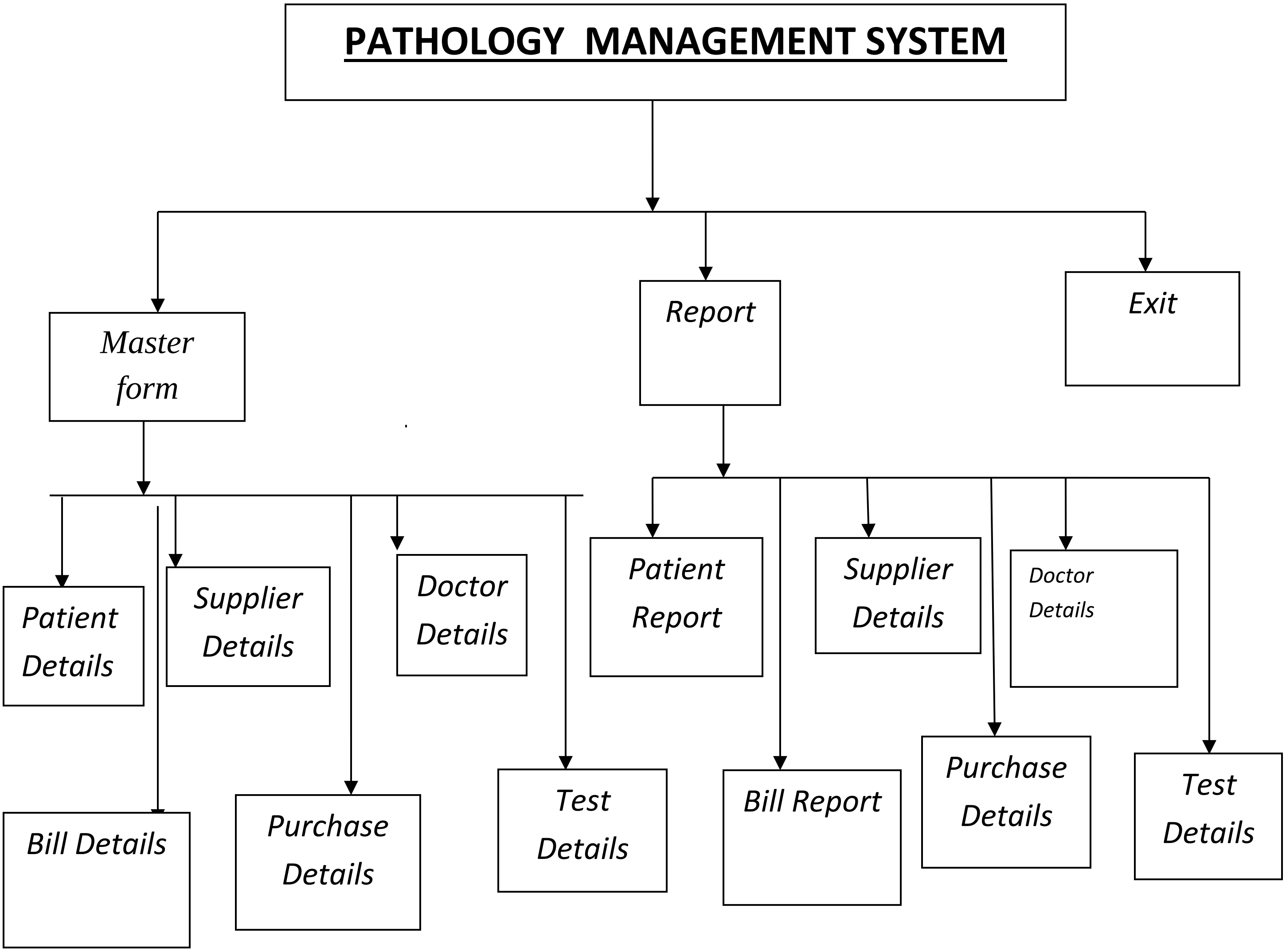


PackageDiagram


Pathology Management System
Patient maintainance details
Doctor maintainance
Test details
Bill maintainance details

System Coding

Menu Tree


PATHOLOGY MANAGEMENT SYSTEM
Master form
Report
Exit
Patient Details
Supplier Details
Doctor Details
Bill Details
Purchase Details
Test Details
Patient Report
Supplier Details
Doctor Details
Bill Report
Purchase Details
Test Details

ListofTableswithConstraints

1] DOCTOR_ DETAILS:-

FIELD_NAME	*DATA TYPE*	*FIELD SIZE*	*CONSTRAINTS*
Doctor _Id	*Number*	*50*	*Primary Key*
Doctor _Name	*Test*	*50*	*Foreign key*
Doctor _ address	*Test*	*50*	
Doctor_ Phone no	*Number*	*50*	

2] PATIENT_ DETAILS:-

FIELD_NAME	*DATA TYPE*	*FIELD SIZE*	*CONSTRAINTS*
Patient _Id	*Number*	*50*	*Primary Key*
Patient _Name	*Text*	*50*	*Foreign Key*
Patient _Address	*Text*	*50*	
Patient _Phone no	*Number*	*50*	

3] TEST_DETAILS:-

FIELD_NAME	*DATA TYPE*	*FIELD SIZE*	*CONSTRAINTS*
Test _Name	*Text*	*50*	*Primary Key*
Test _Report	*Text*	*50*	
Patient _Id	*Number*	*50*	
Test _Sample	*Text*	*50*	

4)BILL_DETAILS:-

FIELD_NAME	DATA TYPE	FIELD SIZE	CONSTRAINTS
Bill _no	Number	50	Primary Key
Bill _Amount	Number	50	

5) SUPPLIER_DETAILS

FIELD NAME	DATA TYPE	FIELD SIZE	CONSTRAINTS
Supplier _id	Number	50	
Supplier _name	Text	50	
Supplier _address	Text	50	
Supplier _no	Number	50	

6) PRODUCT_DETAILS

FIELD NAME	DATA TYPE	FIELD SIZE	CONSTRAINTS
Product _id	Number	50	
Product _name	Text	50	
Mfg _date	Datetime	50	
Exp _date	Datetime	50	
Quantity	Number	50	

7) PURCHASE_DETAILS

FIELD NAME	DATA TYPE	FIELD SIZE	CONSTRAINTS
Purchase _id	Number	50	
Purchase _date	Datetime	50	
Product id	Number	50	
Quantity	Number	50	
Price	Number	50	
Total	Number	50	

ProgramList

FROM NAME	***PURPOSE***
1. Login	*It's Username and Password Property enables to supports Security issues.*
2. MDI Form	*It is Main Form having menu Which Contains further involves forms for Processing.*
3. Supplier _form	*To produce the Supplier details*
4. Patient _form	*To add or Update Patient Record of Patient Details.*
5. Doctor _form	*To add or Update Doctor Record of Doctor Details Form.*
6. Bill _form	*To add or Update Product Stock Record of Stock Master Form.*
7. Product _form	*To check the product.*
8. Purchase _form	*To see the purchase details.*

ReportList

SR. NUMBER	***NAME***	***PURPOSE***
1.	***Product Report***	*This Report will Print the Product which is Ordered.*
2.	***Supplier Report***	*This Report will Print and Store the Supplier Details.*
3.	***Doctor Report***	*This Report will Print and Store the Doctor Details.*
4.	***Patient Report***	*This Report will Print and Maintain the Patient Details.*
5.	***Bill Report***	*This Report will Print and Maintain the Bill Details.*
6.	***Purchase Report***	*This Report Will Print Product Order Which Order to the Supplier.*

ProgramDescription WithNamingConventions

- ***PRODUCT MASTER***

VARIABLE NAME	***DATA TYPE***	***DESCRIPTION***
cn	*Connection*	*It contains Sql Connection.* *It contain records of Products used in Product Master.*
ds	*DataSet*	*To hold all of information from the pathologyDatabase*
da	*DataAdapter*	*Acting as a go-between for the Connection Object and the Data Set.*
maxrow	*Integer*	*It contains Integer values for navigation purpose.*
i	*Integer*	*It contains Integer values.*
j	*Integer*	*It contains Integer values.*
str	*String*	*It contains String values.*

- ***DOCTOR MASTER***

VARIABLE NAME	***DATA TYPE***	***DESCRIPTION***
cn	*Connection*	*It contains Sql Connection.* *It contain records of Distributer used in Distributer Master.*
ds	*DataSet*	*To hold all of information from the pathologyDatabase.*
da	*DataAdapter*	*Acting as a go-between for the Connection Object and the Data Set.*
maxrow	*Integer*	*It contains Integer values for navigation purpose.*
i	*Integer*	*It contains Integer values.*
j	*Integer*	*It contains Integer values.*
str	*String*	*It contains String values.*

➢ *BILL MASTER*

VARIABLE NAME	*DATA TYPE*	*DESCRIPTION*
cn	*Connection*	*It contains Sql Connection.* *It contain records of Doctor used in Doctor Master.*
ds	*DataSet*	*To hold all of information from the pathology Database*
da	*DataAdapter*	*Acting as a go-between for the Connection Object and the Data Set.*
maxrow	*Integer*	*It contains Integer values for navigation purpose.*
i	*Integer*	*It contains Integer values.*
j	*Integer*	*It contains Integer values.*
str	*String*	*It contains String values.*

➢ *TEST MASTER*

VARIABLE NAME	*DATA TYPE*	*DESCRIPTION*
cn	*Connection*	*It contains Sql Connection.* *It contain records of Available Stock used in Stock Master.*
ds	*DataSet*	*To hold all of information from the pathology Database*
da	*DataAdapter*	*Acting as a go-between for the Connection Object and the Data Set.*
maxrow	*Integer*	*It contains Integer values for navigation purpose.*
i	*Integer*	*It contains Integer values.*
j	*Integer*	*It contains Integer values.*
str	*String*	*It contains String values.*

➢ *PURCHASE MASTER*

VARIABLE NAME	*DATA TYPE*	*DESCRIPTION*
cn	*Connection*	*It contains Sql Connection.* *It contain records of ExpiryStock used in Expiry Master.*
ds	*DataSet*	*To hold all of information from the pathology Database*
da	*DataAdapter*	*Acting as a go-between for the Connection Object and the Data Set.*
maxrow	*Integer*	*It contains Integer values for navigation purpose.*
i	*Integer*	*It contains Integer values.*
j	*Integer*	*It contains Integer values.*
str	*String*	*It contains String values.*

➢ *SUPPLIER MASTER*

VARIABLE NAME	*DATA TYPE*	*DESCRIPTION*
cn	*Connection*	*It contains Sql Connection.* *It contain records of Purchase Order used in Purchase Order.*
ds	*DataSet*	*To hold all of information from the pathology Database*
da	*DataAdapter*	*Acting as a go-between for the Connection Object and the Data Set.*
maxrow	*Integer*	*It contains Integer values for navigation purpose.*
i	*Integer*	*It contains Integer values.*
j	*Integer*	*It contains Integer values.*
str	*String*	*It contains String values.*

TableListwithValidation

1] PRODUCTMASTER:-

FIELD NAME	*DATA TYPE*	*VALIDATION*
Product _id	*Number*	*It accepts only Integer Value.*
Product _name	*Text*	*It accepts only text.*
MFG _Date	*Date/Time*	*It accept only date &Time Format*
Expiry _Date	*Date/Time*	*It accept only date &Time Format*
Quantity	*Number*	*It accepts only Integer Value.*

2] DOCTOR MASTER:-

FIELD NAME	*DATA TYPE*	*VALIDATION*
Test _id	*Number*	*It accepts only Integer Value.*
Test_name	*Text*	*It accepts only Text.*
Address	*Text*	*It accepts only text.*
Age	*Number*	*It accepts only integer value.*
Gender	*Text*	*It accepts only text.*
Patient _no	*Number*	*It accepts only integer value.*
Doctor _name	*Text*	*It accepts only text.*
Patient _name	*Text*	*It accepts only Text.*

3] PATIENT MASTER:-

FIELD NAME	*DATA TYPE*	*VALIDATION*
Patient _id	*Number*	*It accepts only Integer Value.*
Patient _name	*Text*	*It accepts only Text.*
Age	*Number*	*It accepts only Integer Value*
Gender	*Text*	*It accepts only Text.*
Reference	*Text*	*It accepts only Text.*
Address	*Text*	*It accepts only Text.*

4] TEST MASTER:-

FIELD NAME	*DATA TYPE*	*VALIDATION*
Test _id	*Number*	*It accepts only Integer Value.*
Test _name	*Text*	*It accepts only Text.*
Amount	*Number*	*It accepts only Integer Value.*
Category	*Text*	*It accepts only Text.*

5] BILL MASTER:-

FIELD NAME	*DATA TYPE*	*VALIDATION*
Bill _id	*Number*	*It accepts only Integer Value.*
Patient _id	*Number*	*It accepts only Integer Value.*
Patient _name	*Text*	*It accepts only Text.*
Bill _date	*Date/Time*	*It accepts only date and time.*
Test _name	*Text*	*It accepts only Text.*
Price	*Number*	*It accepts only Integer Value.*

6] SUPPLIER MASTER:-

FIELD NAME	*DATA TYPE*	*VALIDATION*
Supplier _id	*Number*	*It accepts only Integer Value.*
Supplier _name	*Text*	*It accepts only Text.*
Supplier _address	*Text*	*It accepts only Text.*
Supplier _no	*Number*	*It accepts only Integer Value*
PurchaseOrdeDate	*Date/Time*	*It accept only date &Time Format*

7] PURCHASE MASTER:-

FIELD NAME	*DATA TYPE*	*VALIDATION*
Purchase _id	*Number*	*It accepts only Integer Value.*
Purchase _date	*Date/Time*	*It accepts only date and time.*
Product _id	*Number*	*It accepts only Integer Value*
Quantity	*Number*	*It accepts only Integer Value.*

Price	*Number*	*It accepts only Integer Value.*
Quantity	*Number*	*It accepts only Integer Value.*
Total	*Number*	*It accepts only Integer Value*

TestCases

1] Login form

Test ID: -01
Test Name: - Unit Testing
Test Purpose: -Login validation
Test Focus: -current username and password entry

Test Process

Initiation: - To start the project, User must enter correct Username & Password in the Login Screen.

Attributes: -

Username: - Must be Text
Password: - Must be Text

Test Case: -

1] Username:-
Accepted: - Cursor must be displayed in the Username textbox
on Character Form not in Digit Form.
Excepted Fail: - Digit data inserted cursor stuck to the Username.

2] Password:-
Accepted: - Cursor must be displayed in the Password textbox
on Character Form not in Digit Form.
Excepted Fail: - Digit data inserted cursor stuck to the Password.

2]Doctor Form

Test ID ***: -02***
Test Name ***: -*** *Unit Testing*
Test Purpose: *- Black Box*
Test Focus ***: -*** *Current Data Entry with Valid*

Test Process

Initiation ***: -****To enter the project goes to the MDI Form to enter Doctor Master form and fill the following attributes.*

Attributes ***: -***

Doctor id:-must be text

Doctor address:-must be text

Doctor phone no:-must be number

Test Case ***: -***

1] Doctor id

Excepted:-Cursor must be displayed in the Doctor id Textbox on digit form not in character form.

Excepted Failure:- Character data inserted cursor stuck to the Doctor id.

2] Doctor Name

Excepted:-Cursor must be displayed in the Doctor name Textbox on digit form not in character form.

Excepted Failure:-Character data inserted cursor stuck to the Doctor name.

3] Doctor Address::-

Excepted: - Cursor must be displayed in the Doctor address Textbox on CharacterForm not in Digit Form.

Excepted Failure: - Digit data inserted cursor stuck to the Doctor address.

4] Doctor phone no:-

Accepted: - Cursor must be displayed in the Doctor phone no Textbox ondigitForm not in character Form.

Excepted Fail: - character data inserted cursor stuck to the Doctor phone no.

3]Patient Form

***Test ID** : -<u>03</u>*
***Test Name** : -Unit Testing*
***Test Purpose:** -Black Box*
***Test Focus** : - Current Data Entry with Valid*

Test Process

***Initiation** : - To enter the project goes to the MDI Form to enter Patient Master form and fill the following attributes.*

***Attributes** : -*

Patient id:-must be number.
Patient name:-must be text.
Patient address:-must be text.
Patient phone no:-must be number.

***Test Case** : -*

1] Patient id

Accepted: - Cursor must be displayed in the Patient id Textbox on Digit Form not in Character Form.

Excepted Fail: - Character data inserted cursor stuck to the Patient Code.

2] Distributer Name:-

Accepted: - Cursor must be displayed in the Doctor Name Textbox on CharacterForm not in Digit Form.

Excepted Fail: - Digit data inserted cursor stuck to the Patient Name.

3] Patient Address:-

Accepted: - Cursor must be displayed in the Patient Address Textbox on Character Form not in Digit Form.

Excepted Fail: - Digit data inserted cursor stuck to the Patient Address.

4] Patient phone no:-Cursor must be displayed in the patient phone no Digit data inserted cursor stuck to the patient phone no.

4] Test Details

Test ID ***: -04***
Test Name ***: -*** *Unit Testing*
Test Purpose: - *Black Box*
Test Focus ***: -*** *Current Data Entry with Valid*

Test Process

Initiation ***: -*** *To enter the project goes to the MDI Form to enter Test Master form and fill the following attributes.*

Attributes ***: -***

*Test id**:** - Must be Number*
Test name: - Must be Text
Test Amount: - Must be Number
Test Category:- Must be text

Test Case ***: -***

1] Test id:-

Accepted: - Cursor must be displayed in the Test id Textbox on Digit Form not in Character Form.

Excepted Fail: - Character data inserted cursor stuck to the Test id.

2] Test Name:-

Accepted: - Cursor must be displayed in the Test Name Textbox on Character Form not in Digit Form.

Excepted Fail: - Digit data inserted cursor stuck to the Test Name.

3] Test Amount:-

Accepted: - Cursor must be displayed in the Test amount Textbox on digit Form not in character Form.

Excepted Fail: - character data inserted cursor stuck to the Test amount.

5] Bill Details

Test ID ***: -05***
Test Name ***: -*** *Unit Testing*
Test Purpose: *- Black Box*
Test Focus ***: -*** *Current Data Entry with Valid*

Test Process

Initiation ***: -*** *To enter the project goes to the MDI Form to enter Stock Master form and fill the following attributes.*

Attributes ***: -***

Bill id:- Must be Number.
Bill date:- Must be date and time.

Test Case ***: -***

1] Bill id:-

Accepted: - Cursor must be displayed in the Bill id Textbox on Digit Form not in Character Form.

Excepted Fail: - Character data inserted cursor stuck to the Bill id.

2] Bill date:-

Accepted: - Cursor must be displayed in the Bill date Textbox on Digit Form not in Character Form.

Excepted Fail: - Character data inserted cursor stuck to the Bill date.

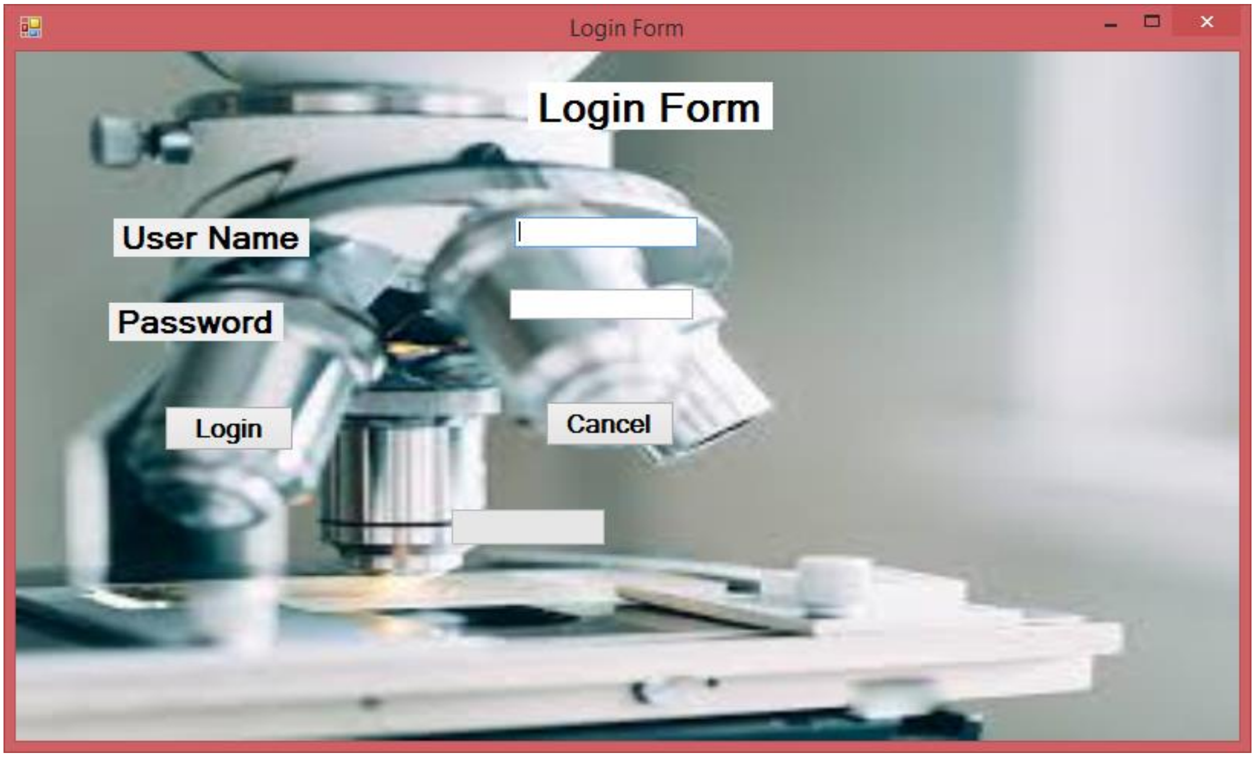
TEST CASE SPECIFICATION

Test Case	Test Data	State	Test I/P Value	Expected Result
1. Username must be contain character only.	Enter character A-Z or a-z	Valid	Neha	It accept username.
	Digit data entered(0-9)	Invalid	Neha12	Message will be displayed character only.
2. Password must be in character format & it should not exceed more than 8 characters.	Enter character A-Z or a-z	Valid	Ghonge	It accept password
	Digit data entered(0-9)	Invalid	5000	Message"Invalid password"

SCREEN

LAYOUT

1] Form Name: - Login Form



TESTDATA

Program ID:- Login_Form

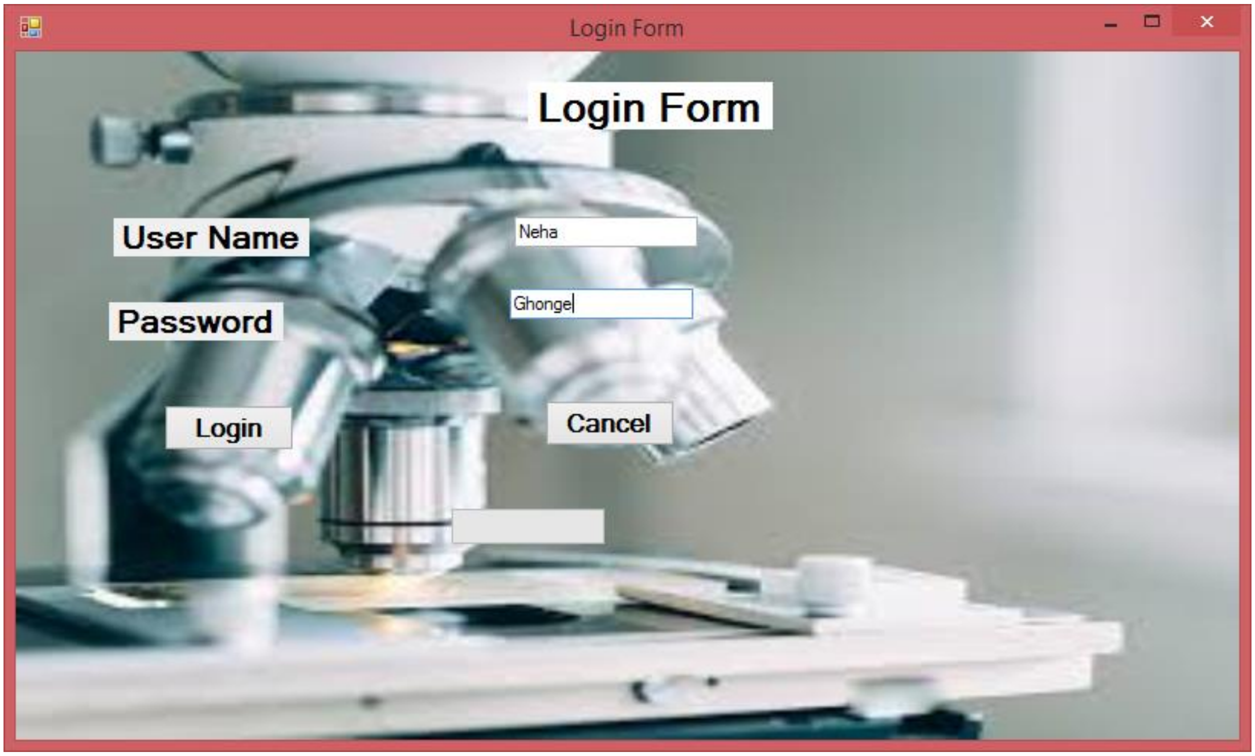
This is used to Login Your Software By using User Name & Password.

PROCEDURE	*DESCRIPTION*
Btnlogin	*If User Name & Password is Correct then Splash Screen is Show* *Else "Wrong User Name or Password" Message Display.*
Btncancel	*Close the Login Form.*

CODING

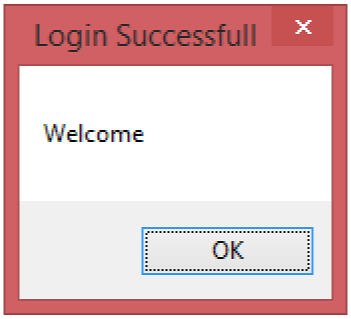
```
Public Class Form1
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        If TextBox1.Text = "Neha" And TextBox2.Text = "Ghonge" Then
            Dim x As Double
            ProgressBar1.Visible = True
            ProgressBar1.Minimum = 1
            ProgressBar1.Maximum = 1000
            ProgressBar1.Value = 1
            ProgressBar1.Step = 1
            For x = 1 To 1000
                ProgressBar1.PerformStep()

            Next x
            MessageBox.Show("Welcome", "Login Successfull")
            MDIParent1.Show()
            Me.Hide()
        Else
            MessageBox.Show("Invalid user")
            Dim cnt As Integer
            cnt = cnt + 1
            If cnt > 3 Then
                MessageBox.Show("Application Ended")
            End If
        End If
    End Sub
    Private Sub Btncancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btncancel.Click
        End
    End Sub

End Class
```

Test Result



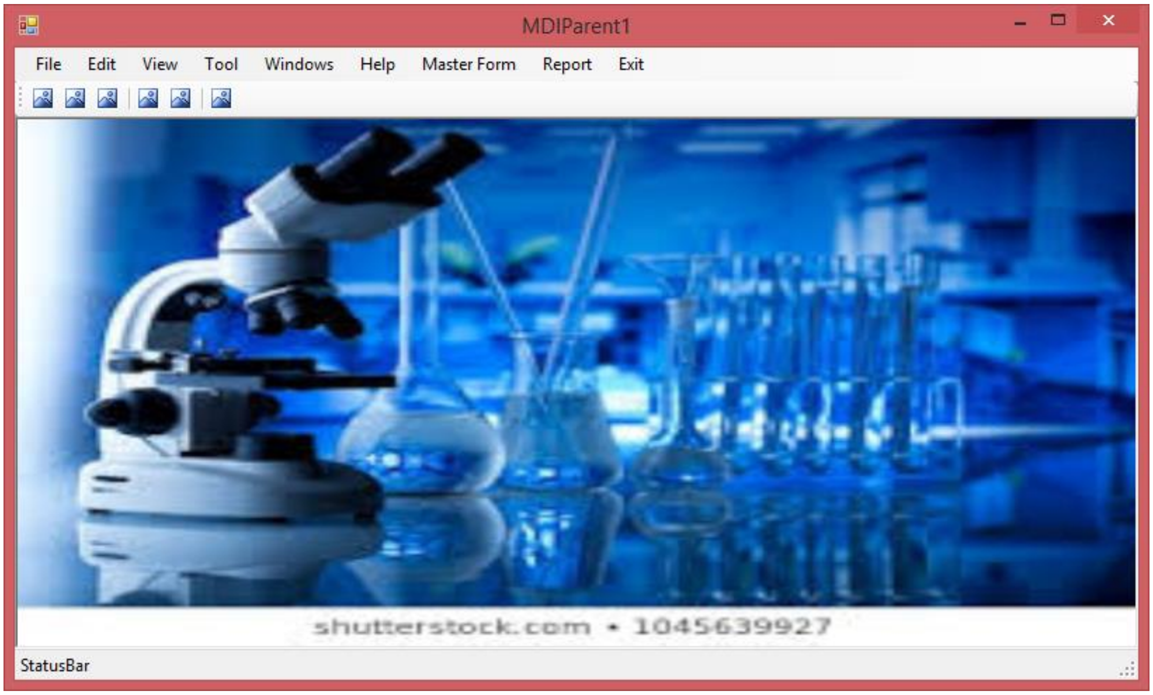
User Name: Neha

Password: Ghonge


Login Successfull

Welcome

2] Form Name: -MDI Form



TESTDATA

Program ID: - MDI

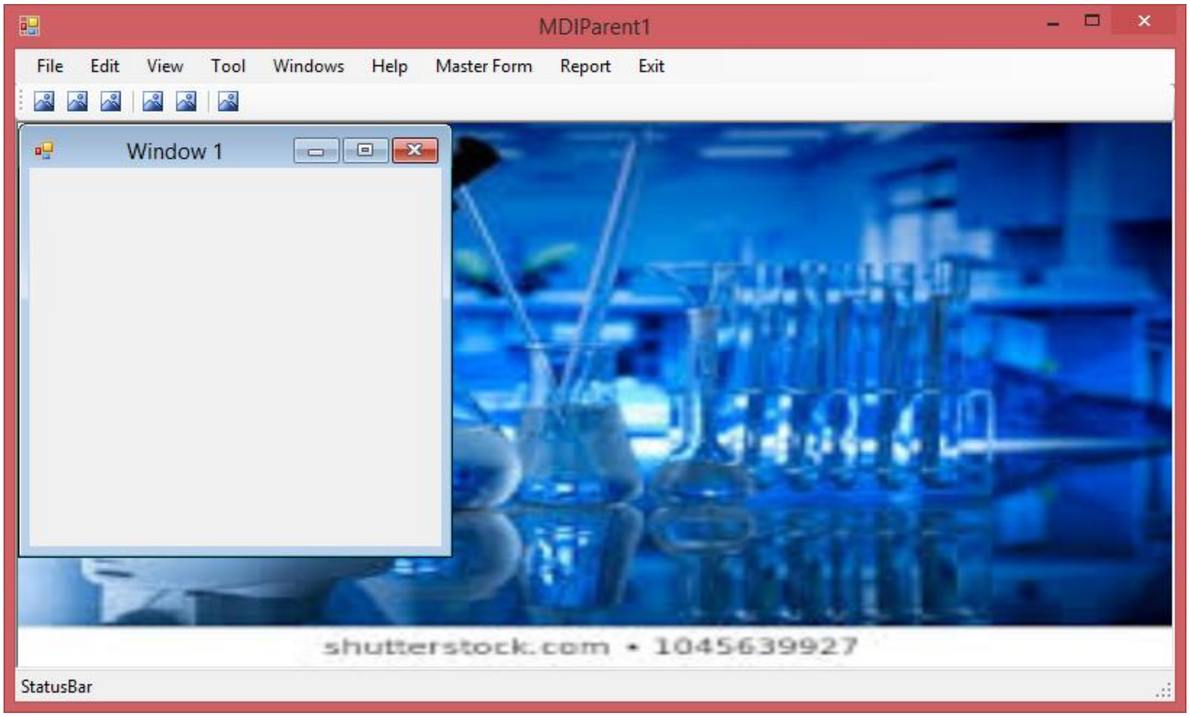
This is your main Form. In that form you can see all the Menu of you program.
All the form is a child of this form.

PROCEDURE	***DESCRIPTION***
Btnfirst	*Go to the first record in the database.*
Btnprev	*Go to the previous record in the database.*
Btnnext	*Go to the next record in the database.*
Btnlast	*Go to the last record in the database.*
Btnadd	*You can add the record.*
Bynsave	*Save all the records in the database.*
Btndelete	*You can delete your record.*
Btncancel	*You can cancel the record.*
Btnsearch	*Go to the search form to search details of particular record*
Btnreport	*Go to the report form to display the records.*
Btnresult	*Move back to the MDI form.*

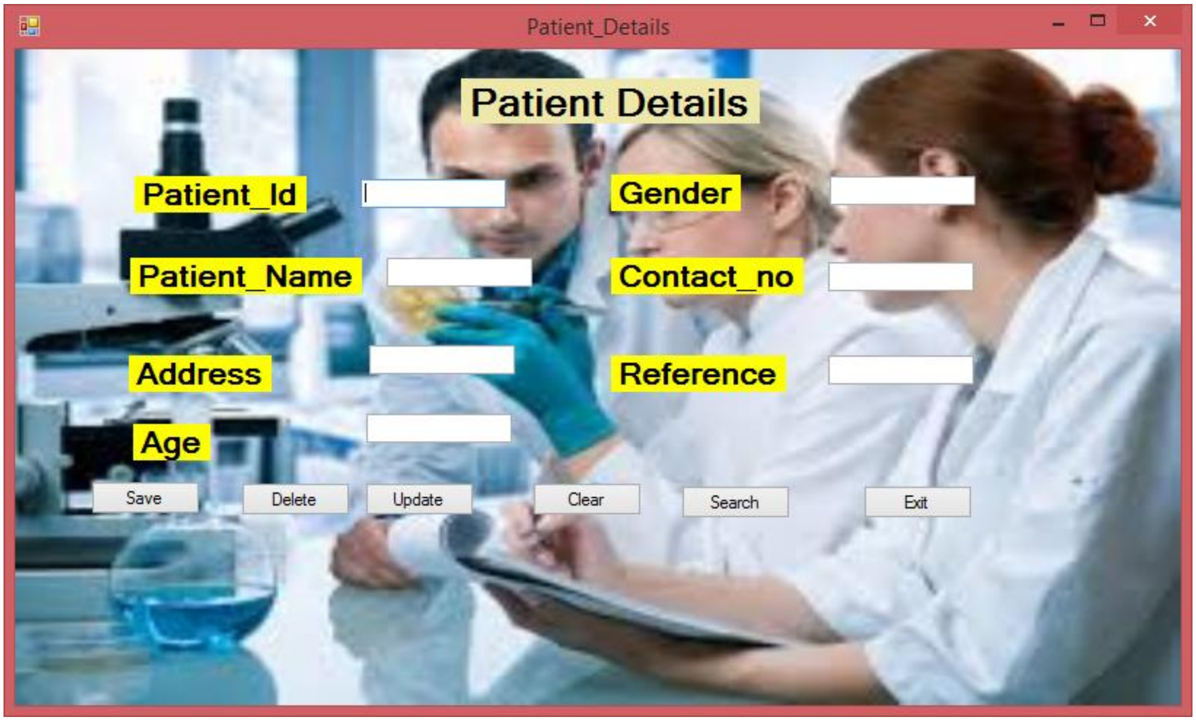
CODING

```
Imports System.Windows.Forms
Public Class MDIParent1

    Private Sub ShowNewForm(ByVal sender As Object, ByVal e As EventArgs) Handles
NewToolStripMenuItem.Click, SaveAs.Click, NewWindowToolStripMenuItem.Click
        ' Create a new instance of the child form.
        Dim ChildForm As New System.Windows.Forms.Form
        ' Make it a child of this MDI form before showing it.
        ChildForm.MdiParent = Me
        m_ChildFormNumber += 1
        ChildForm.Text = "Window " & m_ChildFormNumber

        ChildForm.Show()
    End Sub

    Private Sub OpenFile(ByVal sender As Object, ByVal e As EventArgs) Handles
OpenToolStripMenuItem.Click, OpenToolStripButton.Click
        Dim OpenFileDialog As New OpenFileDialog
        OpenFileDialog.InitialDirectory =
My.Computer.FileSystem.SpecialDirectories.MyDocuments
        OpenFileDialog.Filter = "Text Files (*.txt)|*.txt|All Files (*.*)|*.*"
        If (OpenFileDialog.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then
            Dim FileName As String = OpenFileDialog.FileName
            ' TODO: Add code here to open the file.
        End If
    End Sub

    Private Sub SaveAsToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
Handles SaveAsToolStripMenuItem.Click
        Dim SaveFileDialog As New SaveFileDialog
        SaveFileDialog.InitialDirectory =
My.Computer.FileSystem.SpecialDirectories.MyDocuments
        SaveFileDialog.Filter = "Text Files (*.txt)|*.txt|All Files (*.*)|*.*"

        If (SaveFileDialog.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then
            Dim FileName As String = SaveFileDialog.FileName
            ' TODO: Add code here to save the current contents of the form to a file.
        End If
    End Sub


    Private Sub ExitToolsStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
Handles ExitToolStripMenuItem.Click
        Me.Close()
    End Sub

    Private Sub CutToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
Handles CutToolStripMenuItem.Click
        ' Use My.Computer.Clipboard to insert the selected text or images into the
clipboard
    End Sub

    Private Sub CopyToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
Handles CopyToolStripMenuItem.Click
        ' Use My.Computer.Clipboard to insert the selected text or images into the
clipboard
    End Sub
    Private Sub PasteToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
Handles PasteToolStripMenuItem.Click
        'Use My.Computer.Clipboard.GetText() or My.Computer.Clipboard.GetData to retrieve
information from the clipboard.
    End Sub

    Private Sub ToolBarToolStripMenuItem_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles ToolBarToolStripMenuItem.Click
        Me.ToolStrip.Visible = Me.ToolBarToolStripMenuItem.Checked
    End Sub

    Private Sub StatusBarToolStripMenuItem_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles StatusBarToolStripMenuItem.Click
        Me.StatusStrip.Visible = Me.StatusBarToolStripMenuItem.Checked
    End Sub

    Private Sub CascadeToolStripMenuItem_Click(ByVal sender As Object, ByVal e As
EventArgs)
        Me.LayoutMdi(MdiLayout.Cascade)
    End Sub

    Private Sub TileVerticalToolStripMenuItem_Click(ByVal sender As Object, ByVal e As
EventArgs)
        Me.LayoutMdi(MdiLayout.TileVertical)
    End Sub

    Private Sub TileHorizontalToolStripMenuItem_Click(ByVal sender As Object, ByVal e As
EventArgs)
        Me.LayoutMdi(MdiLayout.TileHorizontal)
    End Sub

    Private Sub ArrangeIconsToolStripMenuItem_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles ArrangeIconsToolStripMenuItem.Click
        Me.LayoutMdi(MdiLayout.ArrangeIcons)
    End Sub

    Private Sub CloseAllToolStripMenuItem_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles CloseAllToolStripMenuItem.Click
        ' Close all child forms of the parent.
        For Each ChildForm As Form In Me.MdiChildren
            ChildForm.Close()
        Next
    End Sub

    Private m_ChildFormNumber As Integer

    Private Sub PatientDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles PatientDetailsToolStripMenuItem.Click
        Patient_Details.Show()
    End Sub

    Private Sub BillDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BillDetailsToolStripMenuItem.Click
        Bill_Details.Show()
    End Sub

    Private Sub DoctorDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles DoctorDetailsToolStripMenuItem.Click
        Doctor_Details.Show()
    End Sub
    Private Sub SupplierDetaildToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles SupplierDetailsToolStripMenuItem.Click
        Supplier_Details.Show()
    End Sub

    Private Sub ProductDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ProductDetailsToolStripMenuItem.Click
        Product_Details.Show()
    End Sub

    Private Sub PurchaseDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles PurchaseDetailsToolStripMenuItem.Click
        Purchase_Details.Show()
    End Sub

    Private Sub ExitToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ExitToolStripMenuItem1.Click
        Me.Close()
    End Sub

    Private Sub TestDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TestDetailsToolStripMenuItem.Click
        Test_Details.Show()
    End Sub

    Private Sub PatientReportToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles PatientReportToolStripMenuItem.Click
        Patient_Report.Show()
    End Sub

    Private Sub TestReportToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TestReportToolStripMenuItem.Click
        Test_Report.Show()
    End Sub

    Private Sub DoctoToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles DoctoToolStripMenuItem.Click
        Doctor_Report.Show()
    End Sub
    Private Sub PurchaseReportToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles PurchaseReportToolStripMenuItem.Click
        Purchase_Report.Show()
    End Sub

    Private Sub BillReportToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BillReportToolStripMenuItem.Click
        Bill_Report.Show()
    End Sub

    Private Sub SupplierReportToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles SupplierReportToolStripMenuItem.Click
        Supplier_Report.Show()
    End Sub

    Private Sub ProductReportToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ProductReportToolStripMenuItem.Click
        Product_Report.Show()
    End Sub

End Class
```

MDIRESULT



3] **Form Name**: -**Patient Details**



TESTDATA

Program ID: - Patient Details

This is used to read & write an information of patient details.

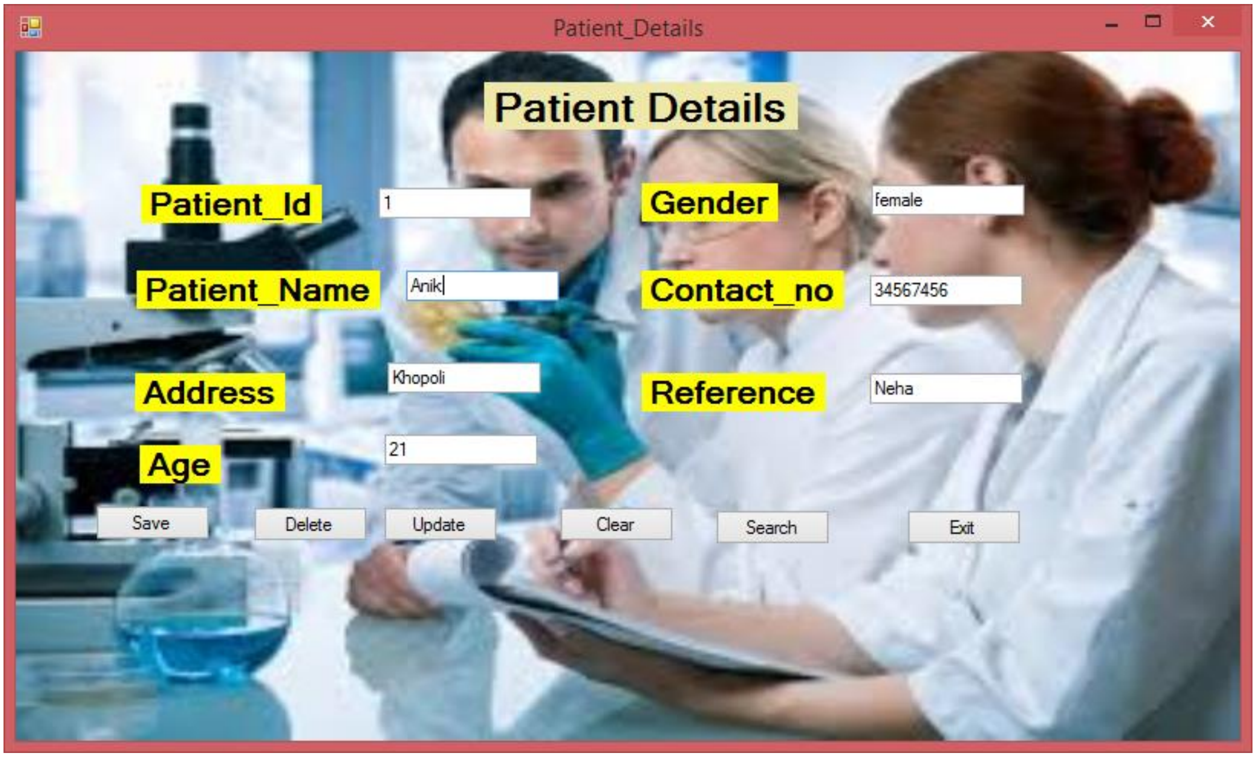
PROCEDURE	***DESCRIPTION***
Btnfirst	*Go to the first record in the database.*
Btnprev	*Go to the previous record in the database.*
Btnnext	*Go to the next record in the database.*
Btnlast	*Go to the last record in the database.*
Btnadd	*You can add the record.*
Bynsave	*Save all the records in the database.*
Btndelete	*You can delete your record.*
Btncancel	*You can cancel the record.*
Btnsearch	*Go to the search form to search details of particular record*
Btnreport	*Go to the report form to display the records.*
Btnexit	*Move back to the MDI form.*

CODING

```
Imports System.Data
Imports System.Data.SqlClient
Public Class Patient_Details
    Dim con1 As New SqlConnection("Data Source=MADHUKAR\SQLEXPRESS;Initial
Catalog=mydb;Integrated Security=True")
    Dim ad As SqlDataAdapter = New SqlDataAdapter("Select * from Patient_info", con1)
    Dim ds As DataSet
    Dim dv As DataView
    Dim abc As Char
    Dim cm As CurrencyManager
    Dim id As Object
    Dim rd, rd1 As SqlDataReader
    Dim cmd As New SqlCommand("Select * from Patient_info", con1)
    Dim cmd1 As SqlCommand
    Dim count As Integer
    Dim i As String
    Private Sub filldata()
        ds = New DataSet
        If con1.State = 1 Then con1.Close()
        con1.Open()
        ad.Fill(ds, "Patient_info")
        dv = New DataView(ds.Tables("Patient_info"))
        cm = CType(Me.BindingContext(dv), CurrencyManager)
    End Sub
    Private Sub bindfield()
        TextBox1.DataBindings.Clear()
        TextBox2.DataBindings.Clear()
        TextBox3.DataBindings.Clear()
        TextBox4.DataBindings.Clear()
        TextBox5.DataBindings.Clear()
        TextBox6.DataBindings.Clear()
        TextBox7.DataBindings.Clear()
        TextBox1.DataBindings.Add("text", dv, "Pat_id")
        TextBox2.DataBindings.Add("text", dv, "Pat_nm")
        TextBox3.DataBindings.Add("text", dv, "Addr")
        TextBox4.DataBindings.Add("text", dv, "Age")
        TextBox5.DataBindings.Add("text", dv, "Gender")
        TextBox6.DataBindings.Add("text", dv, "Contact_no")
        TextBox7.DataBindings.Add("text", dv, "Reference")
    End Sub
    Private Sub Patient_Details_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        filldata()
        bindfield()
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox4.Text
= "" Or TextBox5.Text = "" Or TextBox6.Text = "" Or TextBox7.Text = "" Then
            MsgBox("Please fill all fields with appropriate dada", MsgBoxStyle.Information,
"Save")
        Else
```

```
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            cmd.CommandText = "Insert into patient_info
values(@Pat_id,@Pat_nm,@Addr,@Age,@Gender,@Contact_no,@Reference)"
            cmd.Parameters.AddWithValue("@Pat_id", TextBox1.Text)
            cmd.Parameters.AddWithValue("@Pat_nm", TextBox2.Text)
            cmd.Parameters.AddWithValue("@Addr", TextBox3.Text)
            cmd.Parameters.AddWithValue("@Age", TextBox4.Text)
            cmd.Parameters.AddWithValue("@Gender", TextBox5.Text)
            cmd.Parameters.AddWithValue("@Contact_no", TextBox6.Text)
            cmd.Parameters.AddWithValue("@Reference", TextBox7.Text)
            Try
                cmd.ExecuteNonQuery()
                MsgBox("Record Saved Successfully", MsgBoxStyle.Information, "Save")
            Catch ex As Exception
                MsgBox("Error:" & ex.Source & ":" & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error!!")
            End Try
            con1.Close()
        End If
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        If TextBox1.Text = "" Then
            MsgBox("Please enter Patient's id which you want to delete",
MsgBoxStyle.Information, "Delete")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            If MsgBox("Are you sure?", MsgBoxStyle.YesNo Or MsgBoxStyle.Question, "Delete")
= MsgBoxResult.Yes Then
                cmd.CommandText = "Delete from Patient_info where Pat_id like @Pat_id"
                cmd.Parameters.AddWithValue("@Pat_id", TextBox1.Text)
                cmd.Parameters.AddWithValue("@Pat_nm", TextBox2.Text)
                cmd.Parameters.AddWithValue("@Addr", TextBox3.Text)
                cmd.Parameters.AddWithValue("@Age", TextBox4.Text)
                cmd.Parameters.AddWithValue("@Gender", TextBox5.Text)
                cmd.Parameters.AddWithValue("@Contact_no", TextBox6.Text)
                cmd.Parameters.AddWithValue("@Reference", TextBox7.Text)
                Try
                    cmd.ExecuteNonQuery()
                    MsgBox("Records Deleted Successfully", MsgBoxStyle.Information,
"Delete")
                Catch ex As Exception
                    MessageBox.Show(ex.Message)
                End Try
            End If
            con1.Close()
        End If
    End Sub
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        If TextBox1.Text = "" Then
            MsgBox("Please enter Patient's id which you want to modify",
MsgBoxStyle.Information, "Update")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            cmd.CommandText = "Update Patient_info set
Pat_nm=@Pat_nm,Addr=@Addr,Age=@Age,Gender=@Gender,Contact_no=@Contact_no,Reference=@Referen
ce where Pat_id like @Pat_id"
            cmd.Parameters.AddWithValue("@Pat_id", TextBox1.Text)
            cmd.Parameters.AddWithValue("@Pat_nm", TextBox2.Text)
            cmd.Parameters.AddWithValue("@Addr", TextBox3.Text)
            cmd.Parameters.AddWithValue("@Age", TextBox4.Text)
            cmd.Parameters.AddWithValue("@Gender", TextBox5.Text)
            cmd.Parameters.AddWithValue("@Contact_no", TextBox6.Text)
            cmd.Parameters.AddWithValue("@Reference", TextBox7.Text)
            Try
                cmd.ExecuteNonQuery()
                MsgBox("Records Updated Successfully", MsgBoxStyle.Information, "Update")
            Catch ex As Exception
                MessageBox.Show(ex.Message)
            End Try
            con1.Close()
        End If
    End Sub
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
    End Sub
    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
        If con1.State = 1 Then con1.Close()
        con1.Open()
        Dim cmd As SqlCommand = New SqlCommand("select * from patient_info where Pat_id=" &
TextBox1.Text & "", con1)
        Dim rd As SqlDataReader = cmd.ExecuteReader()
        If Not rd.HasRows Then
            MsgBox("The Data does not exist")
        Else
            MsgBox("The Record is exist")
            rd.Read()

            TextBox1.Text = rd.Item(0)
            TextBox2.Text = rd.Item(1)
            TextBox3.Text = rd.Item(2)
            TextBox4.Text = rd.Item(3)
            TextBox5.Text = rd.Item(4)
            TextBox6.Text = rd.Item(5)
            TextBox7.Text = rd.Item(6)
            rd.Close()
        End If
        con1.Close()
    End Sub
    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click
        Me.Dispose()
    End Sub
    Private Sub TextBox1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
TextBox1.Click
        TextBox1.Clear()
        If con1.State = 1 Then con1.Close()
        con1.Open()
        rd = cmd.ExecuteReader()
        con1.Close()
    End Sub
    Private Sub TextBox2_KeyPress(ByVal eventSender As System.Object, ByVal eventArgs As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress
        Dim KeyAscii As Short = Asc(eventArgs.KeyChar)
        If Not ((KeyAscii < 9 And KeyAscii > 7) Or (KeyAscii >= 65 And KeyAscii <= 90) Or
(KeyAscii >= 97 And KeyAscii <= 122) Or (KeyAscii > 31 And KeyAscii < 33)) Then
            MsgBox("Sorry, Only Alphabets allowed", MsgBoxStyle.Critical, "Name")
            KeyAscii = 0
        End If
        eventArgs.KeyChar = Chr(KeyAscii)
        If KeyAscii = 0 Then
            eventArgs.Handled = True
        End If
    End Sub
    Private Sub TextBox6_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox6.KeyPress
        If Char.IsPunctuation(e.KeyChar) = True Then
            e.KeyChar = ""
            MessageBox.Show("Only Numbers Allowed", "Invalid Data", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
            Exit Sub
        End If
        If Char.IsControl(e.KeyChar) = False Then
            If Char.IsDigit(e.KeyChar) = False Then
                e.KeyChar = ""
                MessageBox.Show("Only Numbers Allowed", "Invalid Data",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

            End If
        End If
    End Sub

End Class
```

TESTRESULT



Patient Details

Patient_Id 1

Gender female

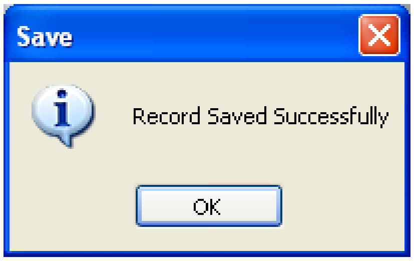
Patient_Name Anik

Contact_no 34567456

Address Khopoli

Reference Neha

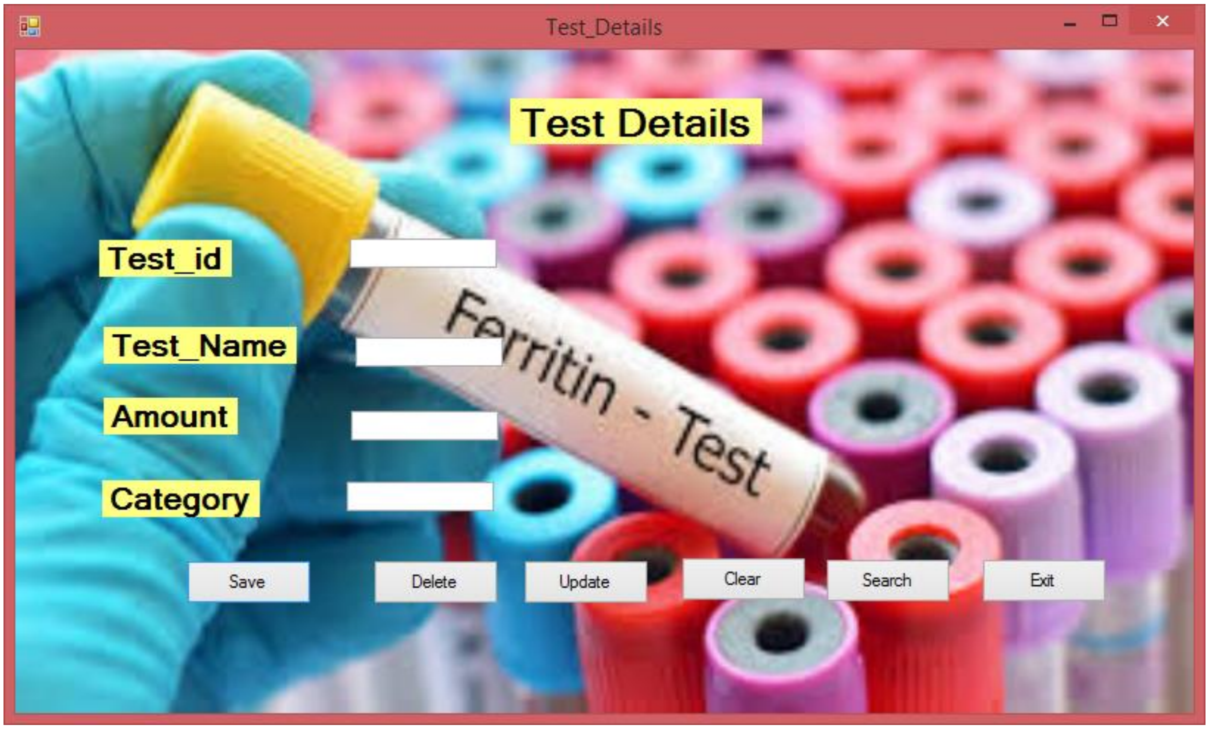
Age 21



Save

Record Saved Successfully

4] Form Name: -Test Details



TESTDATA

Program ID:- Test Form

This is used to read & write an information of test details.

PROCEDURE	*DESCRIPTION*
Btnfirst	*Go to the first record in the database.*
Btnprev	*Go to the previous record in the database.*
Btnnext	*Go to the next record in the database.*
Btnlast	*Go to the last record in the database.*
Btnadd	*You can add the new product record.*
Btnsave	*Save the all record in the database.*
Btndelete	*You can delete the record.*
Btncancel	*You can cancel the record.*
Btngo	*Go to the Search form to search details of particular Product record.*
Btnreport	*Go to the Report form to display the records.*
Btnexit	*Move back to the MDI Form.*

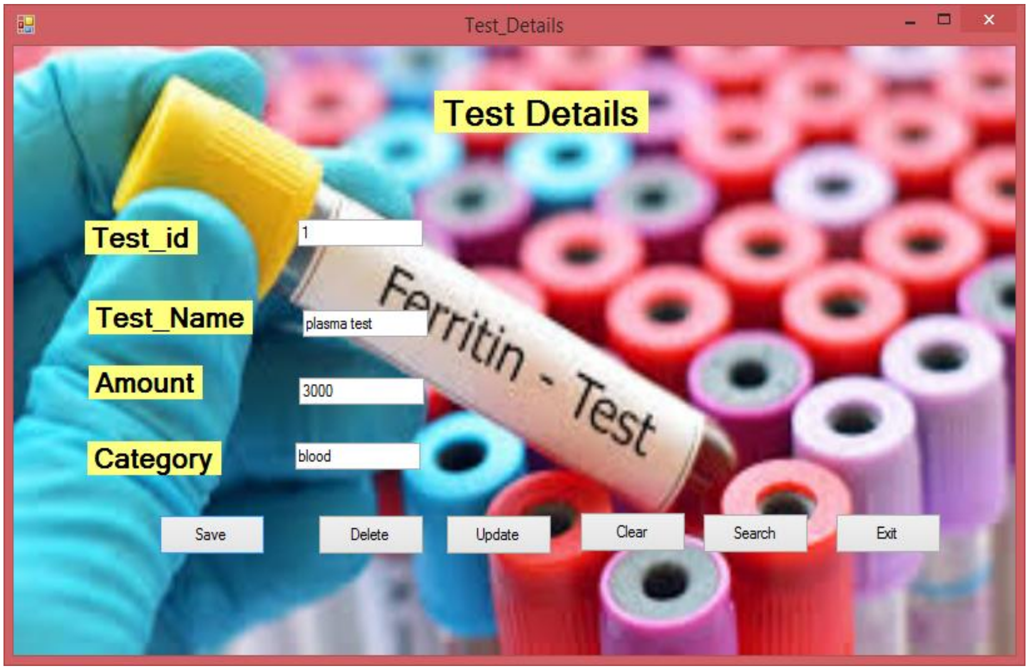
CODING

```
Imports System.Data
Imports System.Data.SqlClient
Public Class Test_Details
    Dim con1 As New SqlConnection("Data Source=MADHUKAR\SQLEXPRESS;Initial
Catalog=mydb;Integrated Security=True")
    Dim ad As SqlDataAdapter = New SqlDataAdapter("Select * from Test_info", con1)
        Dim ds As DataSet
        Dim dv As DataView
        Dim abc As Char
        Dim cm As CurrencyManager
        Dim id As Object
        Dim rd, rd1 As SqlDataReader
        Dim cmd As New SqlCommand("Select * from Test_info", con1)
        Dim cmd1 As SqlCommand
        Dim count As Integer
        Dim i As String
        Private Sub bindfield()
            TextBox1.DataBindings.Clear()
            TextBox2.DataBindings.Clear()
            TextBox3.DataBindings.Clear()
            TextBox4.DataBindings.Clear()

            TextBox1.DataBindings.Add("text", dv, "Test_id")
            TextBox2.DataBindings.Add("text", dv, "Test_nm")
            TextBox3.DataBindings.Add("text", dv, "Amt")
            TextBox4.DataBindings.Add("text", dv, "Cat")

        End Sub
    Private Sub filldata()
        ds = New DataSet
        If con1.State = 1 Then con1.Close()
        con1.Open()
        ad.Fill(ds, "Test_info")
        dv = New DataView(ds.Tables("Test_info"))
        cm = CType(Me.BindingContext(dv), CurrencyManager)
    End Sub
    Private Sub Test_Details_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        filldata()
        bindfield()
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
    End Sub


    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
            If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or
TextBox4.Text = "" Then
                MsgBox("Please fill all fields with appropriate data",
MsgBoxStyle.Information, "Save")
            Else
                If con1.State = 1 Then con1.Close()
                con1.Open()
                Dim cmd As SqlCommand = New SqlCommand
                cmd.Connection = con1
                cmd.CommandText = "Insert into Test_info
values(@Test_id,@Test_nm,@Amt,@Cat)"
                cmd.Parameters.AddWithValue("@Test_id", TextBox1.Text)
                cmd.Parameters.AddWithValue("@Test_nm", TextBox2.Text)
                cmd.Parameters.AddWithValue("@Amt", TextBox3.Text)
                cmd.Parameters.AddWithValue("@Cat", TextBox4.Text)
                Try
                    cmd.ExecuteNonQuery()
                    MsgBox("Records Saved Successfully", MsgBoxStyle.Information, "Save")
                Catch ex As Exception
                    MsgBox("Error:" & ex.Source & ":" & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error!!")
                End Try
                con1.Close()
            End If
        End Sub
        Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
            If TextBox1.Text = "" Then
                MsgBox("Please enter Test id which you want to delete",
MsgBoxStyle.Information, "Delete")
            Else
                If con1.State = 1 Then con1.Close()
                con1.Open()
                Dim cmd As SqlCommand = New SqlCommand
                cmd.Connection = con1
                If MsgBox("Are you sure?", MsgBoxStyle.YesNo Or MsgBoxStyle.Question,
"Delete") = MsgBoxResult.Yes Then
                    cmd.CommandText = "Delete from Test_info where Test_id like@Test_id"
                    cmd.Parameters.AddWithValue("@Test_id", TextBox1.Text)
                    cmd.Parameters.AddWithValue("@Test_nm", TextBox2.Text)
                    cmd.Parameters.AddWithValue("@Amt", TextBox3.Text)
                    cmd.Parameters.AddWithValue("@Cat", TextBox4.Text)

                    Try
                        cmd.ExecuteNonQuery()
                        MsgBox("Records Saved Successfully", MsgBoxStyle.Information,
"Delete")
                    Catch ex As Exception
                        MessageBox.Show(ex.Message)
                    End Try
                End If
                con1.Close()
            End If
        End Sub
        Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
            If TextBox1.Text = "" Then
                MsgBox("Please enter Test_id which you want to modify",
MsgBoxStyle.Information, "Update")
            Else
                If con1.State = 1 Then con1.Close()
                con1.Open()
                Dim cmd As SqlCommand = New SqlCommand
                cmd.Connection = con1
                cmd.CommandText = "Update Test_info set Test_nm=@Test_nm,Amt=@Amt,Cat=@Cat
where Test_id like @Test_id"
                cmd.Parameters.AddWithValue("@Test_id", TextBox1.Text)
                cmd.Parameters.AddWithValue("@Test_nm", TextBox2.Text)
                cmd.Parameters.AddWithValue("@Amt", TextBox3.Text)
```

```
                cmd.Parameters.AddWithValue("@Cat", TextBox4.Text)
                Try
                    cmd.ExecuteNonQuery()
                    MsgBox("Records Saved Successfully", MsgBoxStyle.Information, "Update")
                Catch ex As Exception
                    MessageBox.Show(ex.Message)
                End Try
                con1.Close()
            End If
        End Sub
        Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click

            TextBox1.Text = ""
            TextBox2.Text = ""
            TextBox3.Text = ""
            TextBox4.Text = ""
        End Sub
        Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand("Select * from Test_info where Test_id="
& TextBox1.Text & "", con1)
            Dim rd As SqlDataReader = cmd.ExecuteReader
            If Not rd.HasRows Then
                MsgBox("The Data does not exist")
            Else
                MsgBox("The Record is exist")
                rd.Read()
                TextBox1.Text = rd.Item(0)
                TextBox2.Text = rd.Item(1)
                TextBox3.Text = rd.Item(2)
                TextBox4.Text = rd.Item(3)
                rd.Close()
            End If
            con1.Close()
        End Sub
        Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
            Me.Dispose()
        End Sub
        Private Sub TextBox2_KeyPress(ByVal eventSender As System.Object, ByVal eventArgs
As System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress
            Dim KeyAscii As Short = Asc(eventArgs.KeyChar)
            If Not ((KeyAscii < 9 And KeyAscii > 7) Or (KeyAscii >= 65 And KeyAscii <= 90)
Or (KeyAscii >= 97 And KeyAscii <= 122) Or (KeyAscii >= 31 And KeyAscii < 33)) Then
                MsgBox("Sorry,Only Alphabets Allowed", MsgBoxStyle.Critical, "Name")
                KeyAscii = 0
            End If
            eventArgs.KeyChar = Chr(KeyAscii)
            If KeyAscii = 0 Then
                eventArgs.Handled = True
            End If
        End Sub
        Private Sub TextBox3_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox3.KeyPress
            If Char.IsPunctuation(e.KeyChar) = True Then 'numbers validation'
                e.KeyChar = ""
                MessageBox.Show("Only Numbers Allowed", "Invalid Data",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
```

```
                Exit Sub
            End If
            If Char.IsControl(e.KeyChar) = False Then
                If Char.IsDigit(e.KeyChar) = False Then
                    e.KeyChar = ""
                    MessageBox.Show("Only Numbers Allowed", "Invalid Data",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
                End If
            End If

        End Sub
End Class
```

TESTRESULT



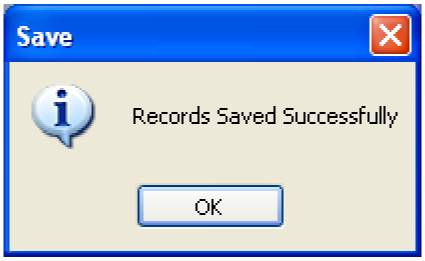
Test Details

Test_id: 1

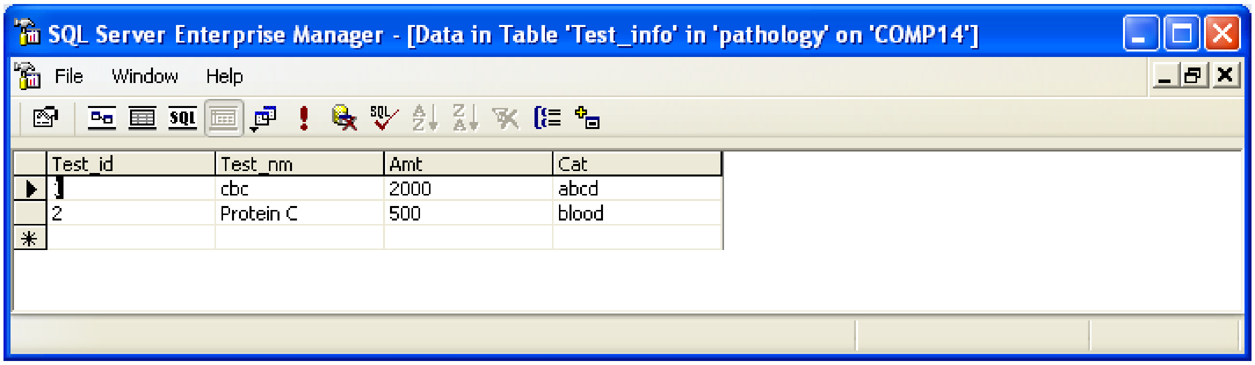
Test_Name: plasma test

Amount: 3000

Category: blood



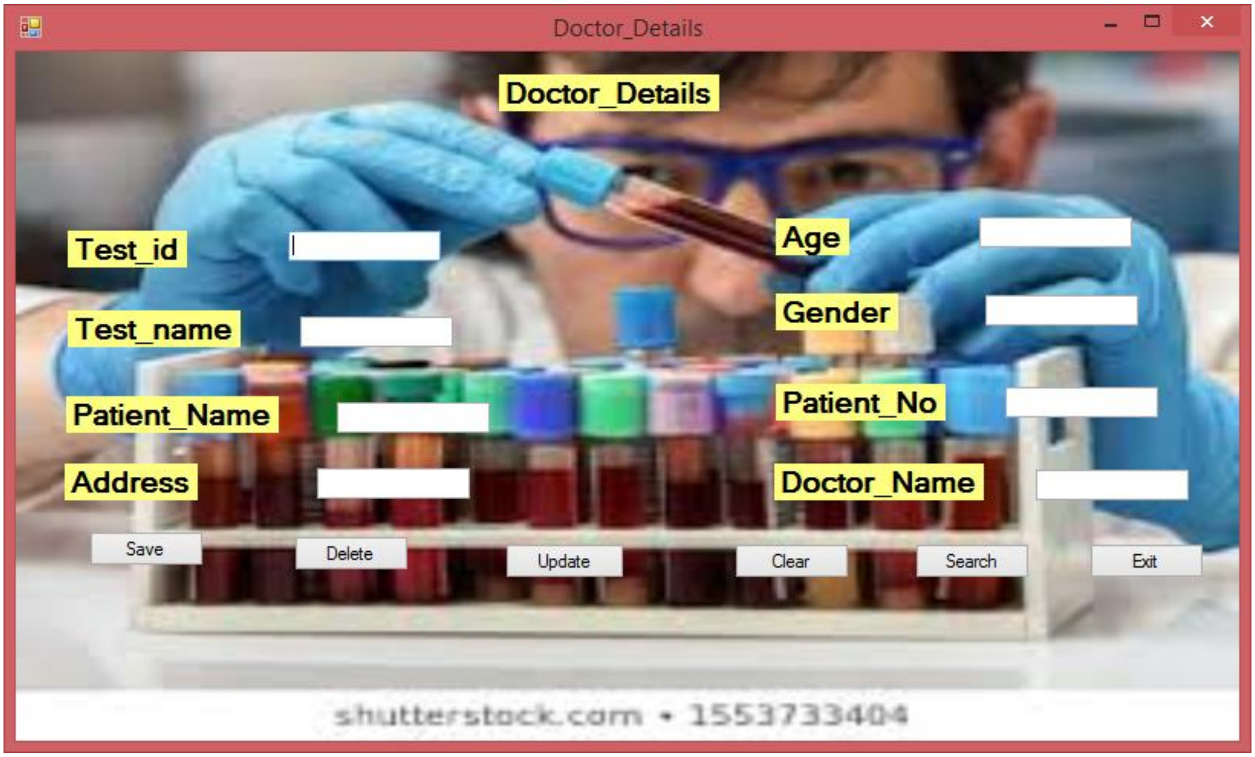
Records Saved Successfully



SQL Server Enterprise Manager - [Data in Table 'Test_info' in 'pathology' on 'COMP14']

Test_id	Test_nm	Amt	Cat
1	cbc	2000	abcd
2	Protein C	500	blood

5] Form Name: - Doctor Details



TESTDATA

Program ID:- Doctor Form

This is used to read & write an information of Doctor Details.

PROCEDURE	*DESCRIPTION*
Btnfirst	*Go to the first record in the database.*
Btnprev	*Go to the previous record in the database.*
Btnnext	*Go to the next record in the database.*
Btnlast	*Go to the last record in the database.*
Btnadd	*You can add the new Distributer record.*
Btnsave	*Save the all record in the database.*
Btndelete	*You can delete the record.*
Btncancel	*You can cancel the record.*
Btnexit	*Move back to the MDI Form.*

CODING

```
Imports System.Data
Imports System.Data.SqlClient
Public Class Doctor_Details


  Dim con1 As New SqlConnection("Data Source=MADHUKAR\SQLEXPRESS;Initial
Catalog=mydb;Integrated Security=True")
  Dim ad As SqlDataAdapter = New SqlDataAdapter("Select * from Doctor_info", con1)
  Dim ds As DataSet
  Dim dv As DataView
  Dim abc As Char
  Dim cm As CurrencyManager
  Dim id As Object
  Dim rd, rd1 As SqlDataReader
  Dim cmd As New SqlCommand("Select * from Doctor_info", con1)
  Dim cmd1 As SqlCommand
  Dim count As Integer
  Dim i As String
  Private Sub filldata()
    ds = New DataSet
    If con1.State = 1 Then con1.Close()
    con1.Open()
    ad.Fill(ds, "Doctor_info")
    dv = New DataView(ds.Tables("Doctor_info"))
    cm = CType(Me.BindingContext(dv), CurrencyManager)
  End Sub
  Private Sub bindfield()
    TextBox1.DataBindings.Clear()
    TextBox2.DataBindings.Clear()
    TextBox3.DataBindings.Clear()
    TextBox4.DataBindings.Clear()
    TextBox5.DataBindings.Clear()
    TextBox6.DataBindings.Clear()
```

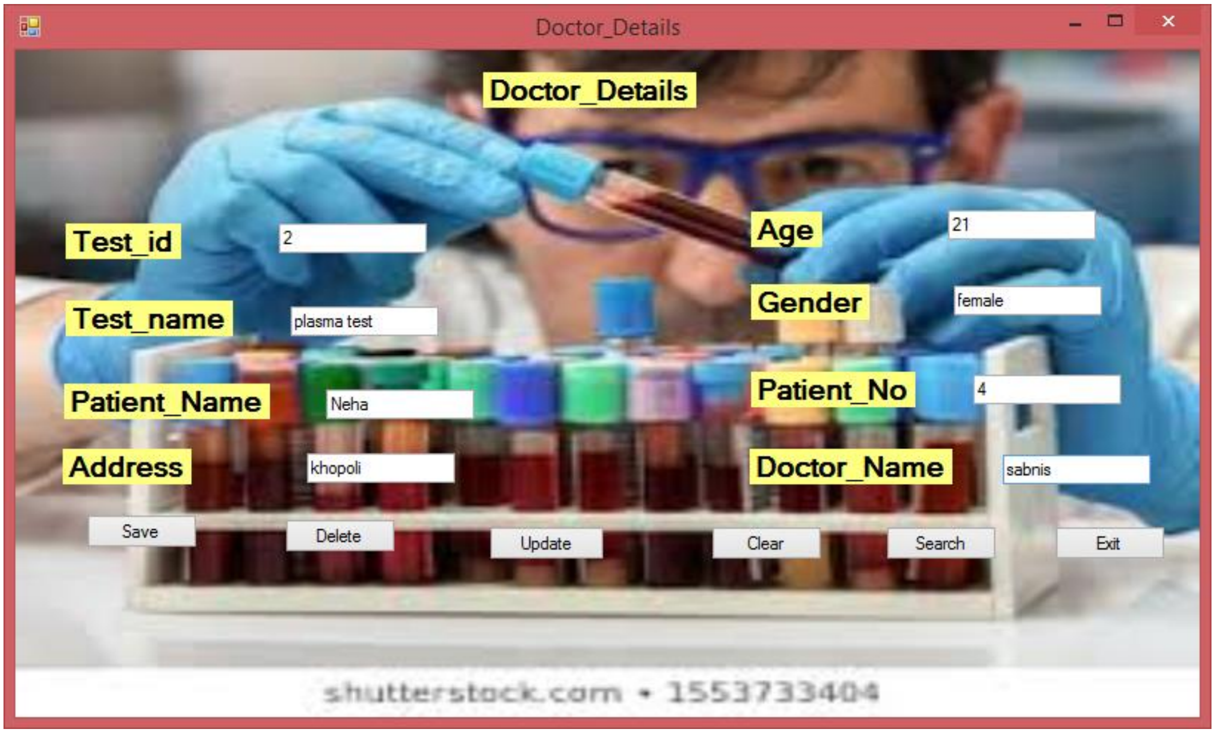
```
        TextBox7.DataBindings.Clear()
        TextBox8.DataBindings.Clear()
        TextBox1.DataBindings.Add("text", dv, "Test_id")
        TextBox2.DataBindings.Add("text", dv, "Test_nm")
        TextBox3.DataBindings.Add("text", dv, "Patient_nm")
        TextBox4.DataBindings.Add("text", dv, "Addr")
        TextBox5.DataBindings.Add("text", dv, "Age")
        TextBox6.DataBindings.Add("text", dv, "Gender")
        TextBox7.DataBindings.Add("text", dv, "Pat_no")
        TextBox8.DataBindings.Add("text", dv, "Doc_nm")
    End Sub
    Private Sub Doctor_Details_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        filldata()
        bindfield()
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
        TextBox8.Text = ""
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox4.Text = "" Or TextBox5.Text = "" Or TextBox6.Text = "" Or TextBox7.Text = "" Or TextBox8.Text = "" Then
            MsgBox("Please fill all fields with appropriate data", MsgBoxStyle.Information, "Save")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            cmd.CommandText = "Insert into Doctor_info values(@Test_id,@Test_nm,@Patient_nm,@Addr,@age,@Gender,@Pat_no,@Doc_nm)"
            cmd.Parameters.AddWithValue("@Test_id", TextBox1.Text)
            cmd.Parameters.AddWithValue("@Test_nm", TextBox2.Text)
            cmd.Parameters.AddWithValue("@Patient_nm", TextBox3.Text)
            cmd.Parameters.AddWithValue("@Addr", TextBox4.Text)
            cmd.Parameters.AddWithValue("@Age", TextBox5.Text)
            cmd.Parameters.AddWithValue("@Gender", TextBox6.Text)
            cmd.Parameters.AddWithValue("@Pat_no", TextBox7.Text)
            cmd.Parameters.AddWithValue("@Doc_nm", TextBox8.Text)
            Try
                cmd.ExecuteNonQuery()
                MsgBox("Records Saved Successfully", MsgBoxStyle.Information, "Save")
            Catch ex As Exception
                MsgBox("Error:" & ex.Source & ":" & ex.Message, MsgBoxStyle.OkOnly, "Connection
Error!!")
            End Try
            con1.Close()
        End If
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        If TextBox1.Text = "" Then
            MsgBox("Please enter Test's id which you want to delete", MsgBoxStyle.Information,
"Delete")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            If MsgBox("Are you sure", MsgBoxStyle.YesNo Or MsgBoxStyle.Question, "Delete") =
MsgBoxResult.Yes Then
                cmd.CommandText = "Delete from Doctor_info where Test_id like @Test_id"
                cmd.Parameters.AddWithValue("@Test_id", TextBox1.Text)
                cmd.Parameters.AddWithValue("@Test_nm", TextBox2.Text)
                cmd.Parameters.AddWithValue("@Patient_nm", TextBox3.Text)
                cmd.Parameters.AddWithValue("@Addr", TextBox4.Text)
                cmd.Parameters.AddWithValue("@Age", TextBox5.Text)
                cmd.Parameters.AddWithValue("@Gender", TextBox6.Text)
                cmd.Parameters.AddWithValue("@Pat_no", TextBox7.Text)
                cmd.Parameters.AddWithValue("@Doc_nm", TextBox8.Text)
                Try
```

```
                    cmd.ExecuteNonQuery()
                    MsgBox("Records Deleted Successfully", MsgBoxStyle.Information, "Delete")
                Catch ex As Exception
                    MessageBox.Show(ex.Message)
                End Try
            End If
        End If
    End Sub
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        If TextBox1.Text = "" Then
            MsgBox("Please enter Test's id which you want to modify", MsgBoxStyle.Information,
"Update")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            If MsgBox("Are you sure", MsgBoxStyle.YesNo Or MsgBoxStyle.Question, "Delete") =
MsgBoxResult.Yes Then
                cmd.CommandText = "Update Doctor_info set
Test_nm=@Test_nm,Patient_nm=@Patient_nm,Addr=@Addr,Age=@Age,Gender=@Gender,Pat_
no=@Pat_no,Doc_nm=@Doc_nm where Test_id like @Test_id"
                cmd.Parameters.AddWithValue("@Test_id", TextBox1.Text)
                cmd.Parameters.AddWithValue("@Test_nm", TextBox2.Text)
                cmd.Parameters.AddWithValue("@Patient_nm", TextBox3.Text)
                cmd.Parameters.AddWithValue("@Addr", TextBox4.Text)
                cmd.Parameters.AddWithValue("@Age", TextBox5.Text)
                cmd.Parameters.AddWithValue("@Gender", TextBox6.Text)
                cmd.Parameters.AddWithValue("@Pat_no", TextBox7.Text)
                cmd.Parameters.AddWithValue("@Doc_nm", TextBox8.Text)
                Try
                    cmd.ExecuteNonQuery()
                    MsgBox("Records Updated Successfully", MsgBoxStyle.Information, "Update")
                Catch ex As Exception
                    MessageBox.Show(ex.Message)
                End Try
            End If
            con1.Close()
        End If
    End Sub
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
        TextBox8.Text = ""
    End Sub
    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
        If con1.State = 1 Then con1.Close()
        con1.Open()
        Dim cmd As SqlCommand = New SqlCommand("select * from Doctor_info where Test_id=" &
TextBox1.Text & "", con1)
        Dim rd As SqlDataReader = cmd.ExecuteReader
        If Not rd.HasRows Then
            MsgBox("The Data does not exist")
        Else
            MsgBox("The Record is exist")
            rd.Read()
            TextBox1.Text = rd.Item(0)
            TextBox2.Text = rd.Item(1)
            TextBox3.Text = rd.Item(2)
            TextBox4.Text = rd.Item(3)
            TextBox5.Text = rd.Item(4)
            TextBox6.Text = rd.Item(5)
            TextBox7.Text = rd.Item(6)
            TextBox8.Text = rd.Item(7)
            rd.Close()
        End If
        con1.Close()
    End Sub
```

```
  Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click
    Me.Dispose()
  End Sub
  Private Sub TextBox2_KeyPress(ByVal eventSender As System.Object, ByVal eventArgs As System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress
    Dim KeyAscii As Short = Asc(eventArgs.KeyChar)
    If Not ((KeyAscii < 9 And KeyAscii > 7) Or (KeyAscii >= 65 And KeyAscii <= 90) Or (KeyAscii >= 97 And KeyAscii <= 122) Or (KeyAscii > 31 And KeyAscii < 33)) Then
      MsgBox("Sorry, Only Alphabets allowed", MsgBoxStyle.Critical, "Name")
      KeyAscii = 0
    End If
    eventArgs.KeyChar = Chr(KeyAscii)
    If KeyAscii = 0 Then
      eventArgs.Handled = True
    End If
  End Sub
  Private Sub TextBox3_KeyPress(ByVal eventSender As System.Object, ByVal eventArgs As System.Windows.Forms.KeyPressEventArgs) Handles TextBox3.KeyPress
    Dim KeyAscii As Short = Asc(eventArgs.KeyChar)
    If Not ((KeyAscii < 9 And KeyAscii > 7) Or (KeyAscii >= 65 And KeyAscii <= 90) Or (KeyAscii >= 97 And KeyAscii <= 122) Or (KeyAscii > 31 And KeyAscii < 33)) Then
      MsgBox("Sorry, Only Alphabets allowed", MsgBoxStyle.Critical, "Name")
      KeyAscii = 0
    End If
    eventArgs.KeyChar = Chr(KeyAscii)
    If KeyAscii = 0 Then
      eventArgs.Handled = True
    End If
  End Sub
  Private Sub TextBox7_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox7.KeyPress
    If Char.IsPunctuation(e.KeyChar) = True Then
      e.KeyChar = ""
      MessageBox.Show("Only Numbers Allowed", "Invalid Data", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
      Exit Sub
    End If
```

```
        If Char.IsControl(e.KeyChar) = False Then
            If Char.IsDigit(e.KeyChar) = False Then
                e.KeyChar = ""
                MessageBox.Show("Only Numbers Allowed", "Invalid Data", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)

            End If
        End If
    End Sub
End Class
```

TESTRESULT



Doctor_Details

Test_id: 2

Test_name: plasma test

Patient_Name: Neha

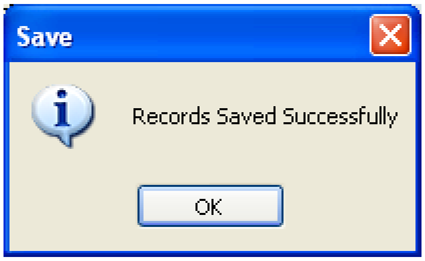
Address: khopoli

Age: 21

Gender: female

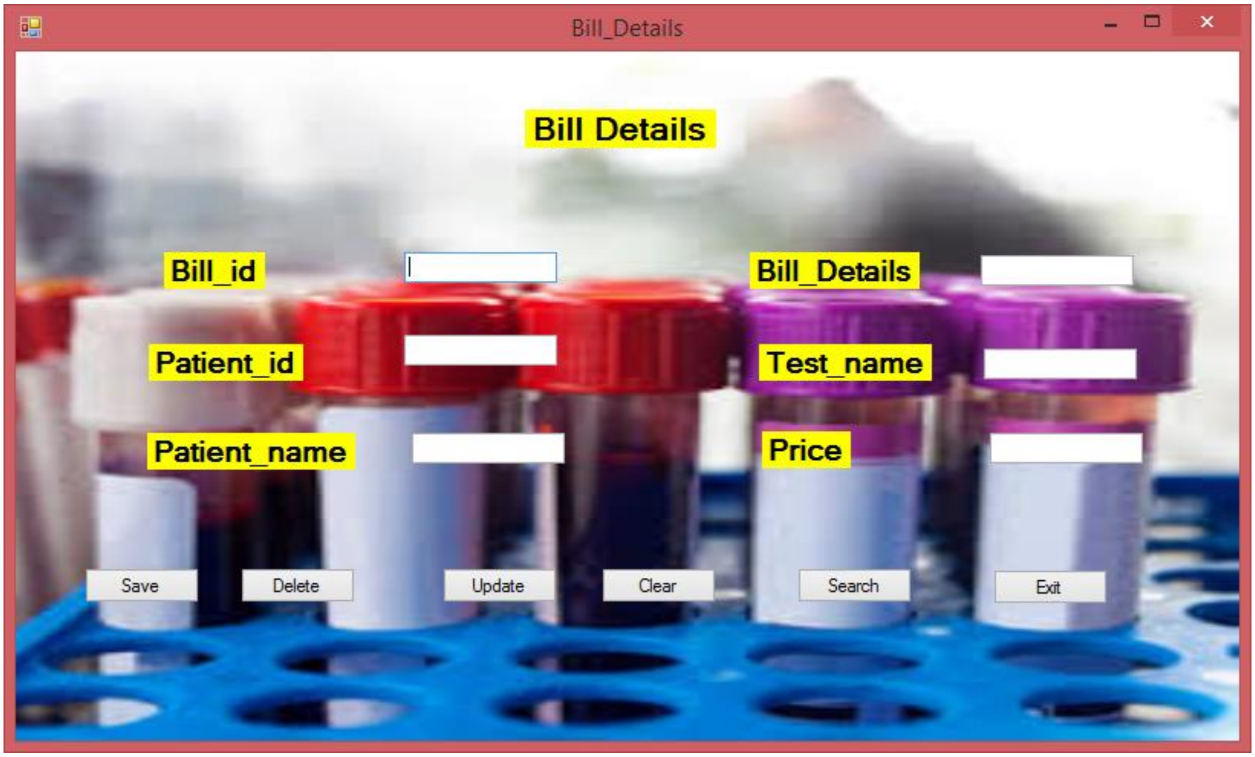
Patient_No: 4

Doctor_Name: sabnis



Records Saved Successfully

6] Form Name: - Bill Details



TESTDATA

Program ID:- Bill Form

This is used to read & write an information of Bill details.

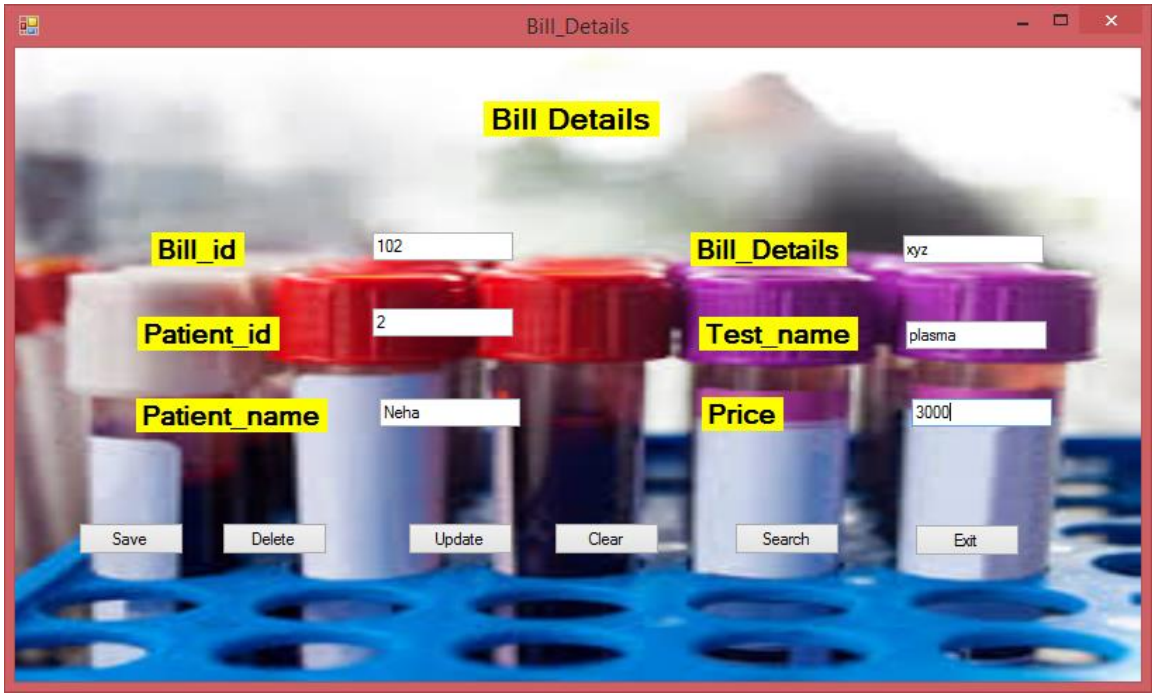
PROCEDURE	*DESCRIPTION*
Btnfirst	*Go to the first record in the database.*
Btnprev	*Go to the previous record in the database.*
Btnnext	*Go to the next record in the database.*
Btnlast	*Go to the last record in the database.*
Btnadd	*You can add the new Doctor record.*
Btnsave	*Save the all record in the database.*
Btndelete	*You can delete the record.*
Btncancel	*You can cancel the record.*
Btnexit	*Move back to the MDI Form.*

CODING

```
Imports System.Data
Imports System.Data.SqlClient
Public Class Bill_Details
    Dim con1 As New SqlConnection("Data Source=MADHUKAR\SQLEXPRESS;Initial
Catalog=mydb;Integrated Security=True")
    Dim ad As SqlDataAdapter = New SqlDataAdapter("select * from Bill_info", con1)
    Dim ds As DataSet
    Dim dv As DataView
    Dim abc As Char
    Dim cm As CurrencyManager
    Dim id As Object
    Dim rd, rd1 As SqlDataReader
    Dim cmd As New SqlCommand("select * from Bill_info", con1)
    Dim cmd1 As SqlCommand
    Dim count As Integer
    Dim i As String
    Private Sub filldata()
        ds = New DataSet
        If con1.State = 1 Then con1.Close()
        con1.Open()
        ad.Fill(ds, "Bill_info")
        dv = New DataView(ds.Tables("Bill_info"))
        cm = CType(Me.BindingContext(dv), CurrencyManager)
    End Sub
    Private Sub bindfield()
        TextBox1.DataBindings.Clear()
        TextBox2.DataBindings.Clear()
        TextBox3.DataBindings.Clear()
        TextBox4.DataBindings.Clear()
        TextBox5.DataBindings.Clear()
        TextBox6.DataBindings.Clear()
        TextBox1.DataBindings.Add("text", dv, "Bill_id")
        TextBox2.DataBindings.Add("text", dv, "Pat_id")
        TextBox3.DataBindings.Add("text", dv, "Pat_nm")
        TextBox4.DataBindings.Add("text", dv, "Bill_date")
        TextBox5.DataBindings.Add("text", dv, "Test_nm")
        TextBox6.DataBindings.Add("text", dv, "Price")
    End Sub
    Private Sub Bill_Details_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        filldata()
        bindfield()
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox4.Text
= "" Or TextBox5.Text = "" Or TextBox6.Text = "" Then
            MsgBox("Please fill all fields with appropriate data", MsgBoxStyle.Information,
"Save")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            cmd.CommandText = "Insert into Bill_info 
values(@Bill_id,@Pat_id,@Pat_nm,@Bill_date,@Test_nm,@Price)"
            cmd.Parameters.AddWithValue("@Bill_id", TextBox1.Text)
            cmd.Parameters.AddWithValue("@Pat_id", TextBox2.Text)
            cmd.Parameters.AddWithValue("@pat_nm", TextBox3.Text)
            cmd.Parameters.AddWithValue("@Bill_date", TextBox4.Text)
            cmd.Parameters.AddWithValue("@Test_nm", TextBox5.Text)
            cmd.Parameters.AddWithValue("@Price", TextBox6.Text)
            Try
                cmd.ExecuteNonQuery()
                MsgBox("Records Saved Successfully", MsgBoxStyle.Information, "Save")
            Catch ex As Exception
                MsgBox("Error:" & ex.Source & ":" & ex.Message, MsgBoxStyle.OkOnly, 
"Connection Error!!")
            End Try
            con1.Close()
        End If
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) 
Handles Button2.Click
        If TextBox1.Text = "" Then
            MsgBox("Please enter Bill id which you want to delete", 
MsgBoxStyle.Information, "Delete")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            If MsgBox("Are you sure?", MsgBoxStyle.YesNo Or MsgBoxStyle.Question, "Delete") 
= MsgBoxResult.Yes Then
                cmd.CommandText = "Delete from Bill_info where Bill_id like @Bill_id"
                cmd.Parameters.AddWithValue("@Bill_id", TextBox1.Text)
                cmd.Parameters.AddWithValue("@Pat_id", TextBox2.Text)
                cmd.Parameters.AddWithValue("@Pat_nm", TextBox3.Text)
                cmd.Parameters.AddWithValue("@Bill_date", TextBox4.Text)
                cmd.Parameters.AddWithValue("@Test_nm", TextBox5.Text)
                cmd.Parameters.AddWithValue("@Price", TextBox6.Text)

                Try
                    cmd.ExecuteNonQuery()
                    MsgBox("Records Deleted Successfully", MsgBoxStyle.Information, 
"Delete")
                Catch ex As Exception
                    MessageBox.Show(ex.Message)
                End Try
            End If
            con1.Close()
        End If
    End Sub
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) 
Handles Button3.Click
        If TextBox1.Text = "" Then
            MsgBox("Please enter Bill id which you want to modify", 
MsgBoxStyle.Information, "Update")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            cmd.CommandText = "Update Bill_info set
Pat_id=@Pat_id,Pat_nm=@Pat_nm,Bill_date=@Bill_date,Test_nm=@Test_nm,Price=@Price where
Bill_id like @Bill_id"
            cmd.Parameters.AddWithValue("@Bill_id", TextBox1.Text)
            cmd.Parameters.AddWithValue("@Pat_id", TextBox2.Text)
            cmd.Parameters.AddWithValue("@Pat_nm", TextBox3.Text)
            cmd.Parameters.AddWithValue("@Bill_date", TextBox4.Text)
            cmd.Parameters.AddWithValue("@Test_nm", TextBox5.Text)
            cmd.Parameters.AddWithValue("@Price", TextBox6.Text)

            Try
                cmd.ExecuteNonQuery()
                MsgBox("Records Updated Successfully", MsgBoxStyle.Information, "Update")
            Catch ex As Exception
                MessageBox.Show(ex.Message)
            End Try
            con1.Close()
        End If
    End Sub
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""

    End Sub
    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
        If con1.State = 1 Then con1.Close()
        con1.Open()
        Dim cmd As SqlCommand = New SqlCommand("select * from Bill_info where Bill_id=" &
TextBox1.Text & "", con1)
        Dim rd As SqlDataReader = cmd.ExecuteReader
        If Not rd.HasRows Then
            MsgBox("The Data does not exist")
        Else
            MsgBox("The Record is exist")
            rd.Read()

            TextBox1.Text = rd.Item(0)
            TextBox2.Text = rd.Item(1)
            TextBox3.Text = rd.Item(2)
            TextBox4.Text = rd.Item(3)
            TextBox5.Text = rd.Item(4)
            TextBox6.Text = rd.Item(5)

            rd.Close()
        End If
        con1.Close()
    End Sub
    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click
        Me.Dispose()
    End Sub

    Private Sub TextBox1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles TextBox1.Click
```

```
        TextBox1.Clear()
        If con1.State = 1 Then con1.Close()
        con1.Open()
        rd = cmd.ExecuteReader()
        con1.Close()
    End Sub


    Private Sub TextBox3_KeyPress(ByVal eventSender As System.Object, ByVal eventArgs As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox3.KeyPress
        Dim KeyAscii As Short = Asc(eventArgs.KeyChar)
        If Not ((KeyAscii < 9 And KeyAscii > 7) Or (KeyAscii >= 65 And KeyAscii <= 90) Or
(KeyAscii >= 97 And KeyAscii <= 122) Or (KeyAscii > 31 And KeyAscii < 33)) Then
            MsgBox("Sorry, Only Alphabets allowed", MsgBoxStyle.Critical, "Name")
            KeyAscii = 0
        End If
        eventArgs.KeyChar = Chr(KeyAscii)
        If KeyAscii = 0 Then
            eventArgs.Handled = True

        End If
    End Sub
End Class
```

TESTRESULT



Bill_Details

Bill Details

Bill_id: 102

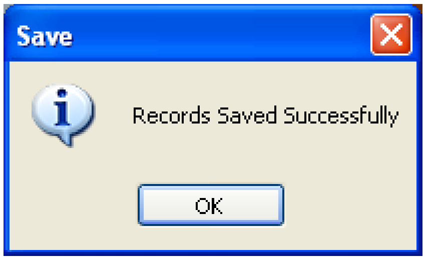
Patient_id: 2

Patient_name: Neha

Bill_Details: xyz

Test_name: plasma

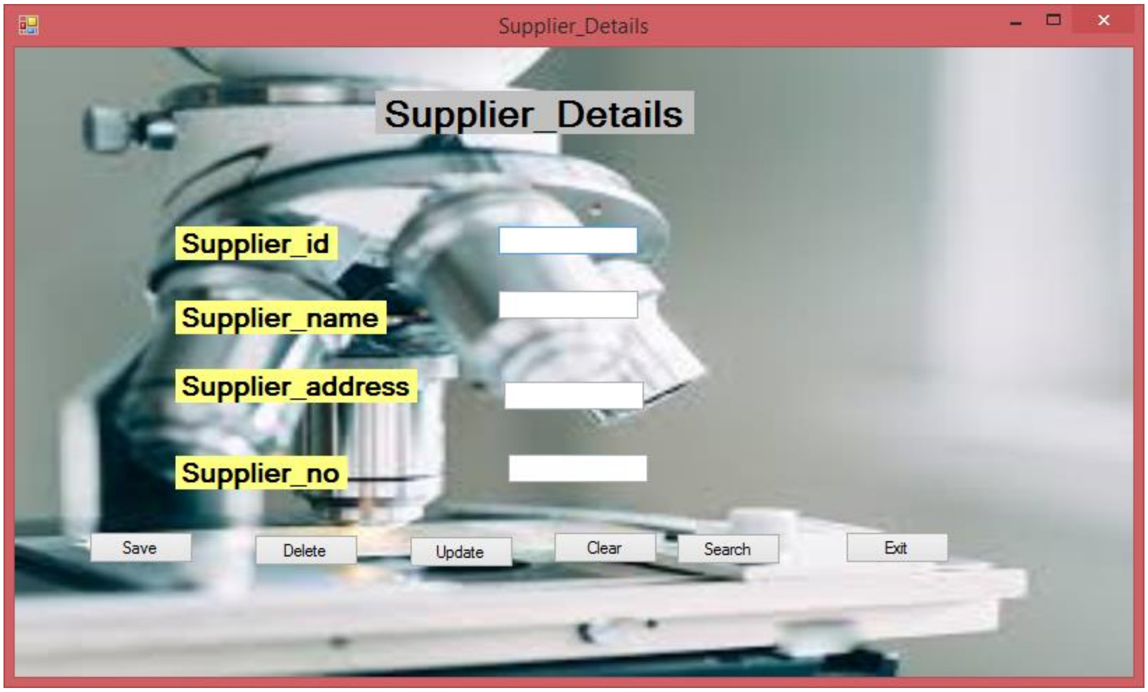
Price: 3000



Save

Records Saved Successfully

7] Form Name: - Supplier Details



TESTDATA

Program ID:- Supplier Form

This is used to read & write an information of Supplier details.

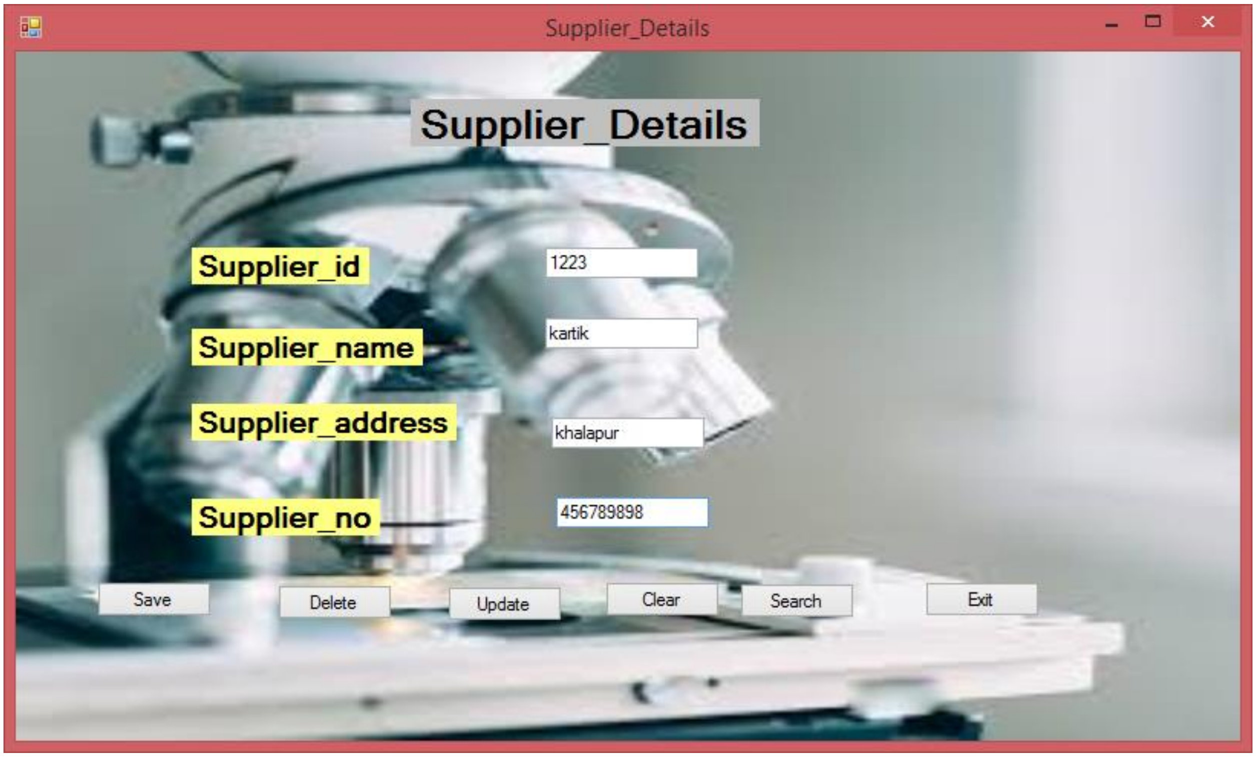
PROCEDURE	*DESCRIPTION*
Btnfirst	*Go to the first record in the database.*
Btnprev	*Go to the previous record in the database.*
Btnnext	*Go to the next record in the database.*
Btnlast	*Go to the last record in the database.*
Btnadd	*You can add the new Stock record.*
Btnsave	*Save the all record in the database.*
Btndelete	*You can delete the record.*
Btncancel	*You can cancel the record.*
Btnsearch	*Go to the Search form to search details of particular Stock record.*
Btnreport	*Go to the Report form to display the records.*
Btnexit	*Move back to the MDI Form.*

CODING

```
Imports System.Data
Imports System.Data.SqlClient
Public Class Supplier_Details
    Dim con1 As New SqlConnection("Data Source=MADHUKAR\SQLEXPRESS;Initial
Catalog=mydb;Integrated Security=True")
    Dim ad As SqlDataAdapter = New SqlDataAdapter("select * from Supplier_info", con1)
    Dim ds As DataSet
    Dim dv As DataView
    Dim abc As Char
    Dim cm As CurrencyManager
    Dim id As Object
    Dim rd, rd1 As SqlDataReader
    Dim cmd As New SqlCommand("select * from Supplier_info", con1)
    Dim cmd1 As SqlCommand
    Dim count As Integer
    Dim i As String
    Private Sub filldata()
        ds = New DataSet
        If con1.State = 1 Then con1.Close()
        con1.Open()
        ad.Fill(ds, "Supplier_info")
        dv = New DataView(ds.Tables("Supplier_info"))
        cm = CType(Me.BindingContext(dv), CurrencyManager)
    End Sub
    Private Sub bindfield()
        TextBox1.DataBindings.Clear()
        TextBox2.DataBindings.Clear()
        TextBox3.DataBindings.Clear()
        TextBox4.DataBindings.Clear()

        TextBox1.DataBindings.Add("text", dv, "Supplier_id")
        TextBox2.DataBindings.Add("text", dv, "Supplier_nm")
        TextBox3.DataBindings.Add("text", dv, "Supplier_addr")
        TextBox4.DataBindings.Add("text", dv, "Supplier_no")
    End Sub
    Private Sub Supplier_Details_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        filldata()
        bindfield()
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox4.Text
= "" Then
            MsgBox("Please fill all fields with appropriate data", MsgBoxStyle.Information,
"Save")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            cmd.CommandText = "Insert into Supplier_info 
values(@Supplier_id,@Supplier_nm,@Supplier_addr,@Supplier_no)"
            cmd.Parameters.AddWithValue("@Supplier_id", TextBox1.Text)
            cmd.Parameters.AddWithValue("@Supplier_nm", TextBox2.Text)
            cmd.Parameters.AddWithValue("@Supplier_addr", TextBox3.Text)
            cmd.Parameters.AddWithValue("@Supplier_no", TextBox4.Text)

            Try
                cmd.ExecuteNonQuery()
                MsgBox("Records Saved Successfully", MsgBoxStyle.Information, "Save")
            Catch ex As Exception
                MsgBox("Error:" & ex.Source & ":" & ex.Message, MsgBoxStyle.OkOnly, 
"Connection Error!!")
            End Try
            con1.Close()
        End If
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) 
Handles Button2.Click
        If TextBox1.Text = "" Then
            MsgBox("Please enter Supplier id which you want to delete", 
MsgBoxStyle.Information, "Delete")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            If MsgBox("Are you sure?", MsgBoxStyle.YesNo Or MsgBoxStyle.Question, "Delete") 
= MsgBoxResult.Yes Then
                cmd.CommandText = "Delete Supplier_info where Supplier_id like 
@Supplier_id"
                cmd.Parameters.AddWithValue("@Supplier_id", TextBox1.Text)
                cmd.Parameters.AddWithValue("@Supplier_nm", TextBox2.Text)
                cmd.Parameters.AddWithValue("@Supplier_addr", TextBox3.Text)
                cmd.Parameters.AddWithValue("@Supplier_no", TextBox4.Text)


                Try
                    cmd.ExecuteNonQuery()
                    MsgBox("Records Deleted Successfully", MsgBoxStyle.Information, 
"Delete")
                Catch ex As Exception
                    MessageBox.Show(ex.Message)
                End Try
            End If
            con1.Close()
        End If
    End Sub
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) 
Handles Button3.Click
        If TextBox1.Text = "" Then
            MsgBox("Please enter Supplier id which you want to modify", 
MsgBoxStyle.Information, "Update")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            cmd.CommandText = "Update Supplier_info set 
Supplier_nm=@Supplier_nm,Supplier_addr=@Supplier_addr,Supplier_no=@Supplier_no, where 
Supplier_id like @Supplier_id"
```

```
            cmd.Parameters.AddWithValue("@Supplier_id", TextBox1.Text)
            cmd.Parameters.AddWithValue("@Supplier_nm", TextBox2.Text)
            cmd.Parameters.AddWithValue("@Supplier_addr", TextBox3.Text)
            cmd.Parameters.AddWithValue("@Supplier_no", TextBox4.Text)


            Try
                cmd.ExecuteNonQuery()
                MsgBox("Records Updated Successfully", MsgBoxStyle.Information, "Update")
            Catch ex As Exception
                MessageBox.Show(ex.Message)
            End Try
            con1.Close()
        End If
    End Sub
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
    End Sub
    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
        If con1.State = 1 Then con1.Close()
        con1.Open()
        Dim cmd As SqlCommand = New SqlCommand("select * from Supplier_info where
Supplier_id=" & TextBox1.Text & "", con1)
        Dim rd As SqlDataReader = cmd.ExecuteReader
        If Not rd.HasRows Then
            MsgBox("The Data does not exist")
        Else
            MsgBox("The Record is exist")
            rd.Read()

            TextBox1.Text = rd.Item(0)
            TextBox2.Text = rd.Item(1)
            TextBox3.Text = rd.Item(2)
            TextBox4.Text = rd.Item(3)

            rd.Close()
        End If
        con1.Close()
    End Sub
    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click
        Me.Dispose()
    End Sub

    Private Sub TextBox2_KeyPress(ByVal eventSender As System.Object, ByVal eventArgs As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress
        Dim KeyAscii As Short = Asc(eventArgs.KeyChar)
        If Not ((KeyAscii < 9 And KeyAscii > 7) Or (KeyAscii >= 65 And KeyAscii <= 90) Or
(KeyAscii >= 97 And KeyAscii <= 122) Or (KeyAscii > 31 And KeyAscii < 33)) Then
            MsgBox("Sorry, Only Alphabets allowed", MsgBoxStyle.Critical, "Name")
            KeyAscii = 0
        End If
        eventArgs.KeyChar = Chr(KeyAscii)
        If KeyAscii = 0 Then
            eventArgs.Handled = True
        End If
    End Sub
    Private Sub TextBox3_KeyPress(ByVal eventSender As System.Object, ByVal eventArgs As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox3.KeyPress
        Dim KeyAscii As Short = Asc(eventArgs.KeyChar)
        If Not ((KeyAscii < 9 And KeyAscii > 7) Or (KeyAscii >= 65 And KeyAscii <= 90) Or
(KeyAscii >= 97 And KeyAscii <= 122) Or (KeyAscii > 31 And KeyAscii < 33)) Then
            MsgBox("Sorry, Only Alphabets allowed", MsgBoxStyle.Critical, "Name")
            KeyAscii = 0
        End If
        eventArgs.KeyChar = Chr(KeyAscii)
        If KeyAscii = 0 Then
            eventArgs.Handled = True
        End If
    End Sub
End Class
```

TESTRESULT



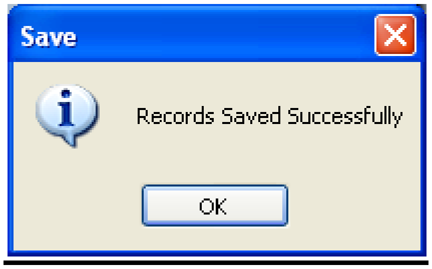
Supplier_Details

Supplier_id 1223

Supplier_name kartik

Supplier_address khalapur

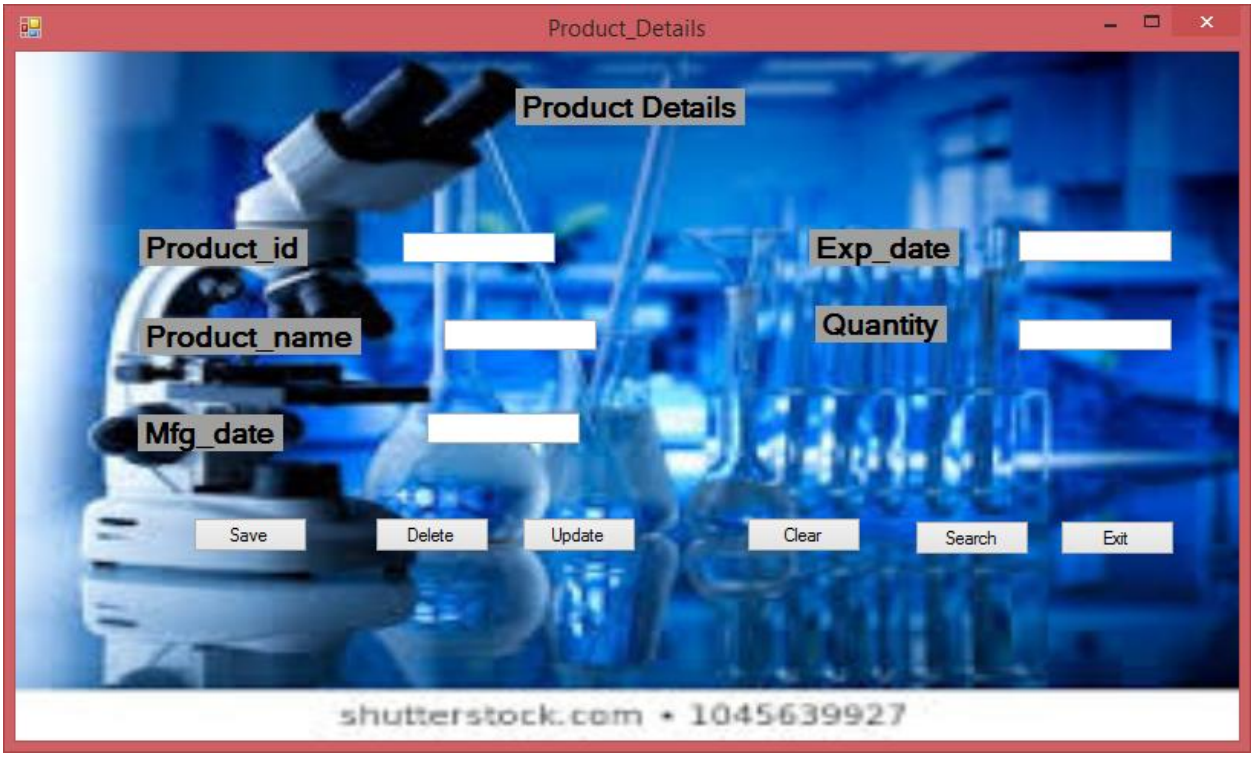
Supplier_no 456789898



Save

Records Saved Successfully

8] Form Name: -Product Details



TESTDATA

Program ID:- Product Form

This is used to read & write an information of Product details.

PROCEDURE	*DESCRIPTION*
Btnfirst	*Go to the first record in the database.*
Btnprev	*Go to the previous record in the database.*
Btnnext	*Go to the next record in the database.*
Btnlast	*Go to the last record in the database.*
Btnadd	*You can add the new Expiry Product record.*
Btnsave	*Save the all record in the database.*
Btndelete	*You can delete the record.*
Btncancel	*You can cancel the record.*
Btnsearch	*Go to the Search form to search details of particular Expiry Product record.*
Btnreport	*Go to the Report form to display the records.*
Btnexit	*Move back to the MDI Form.*

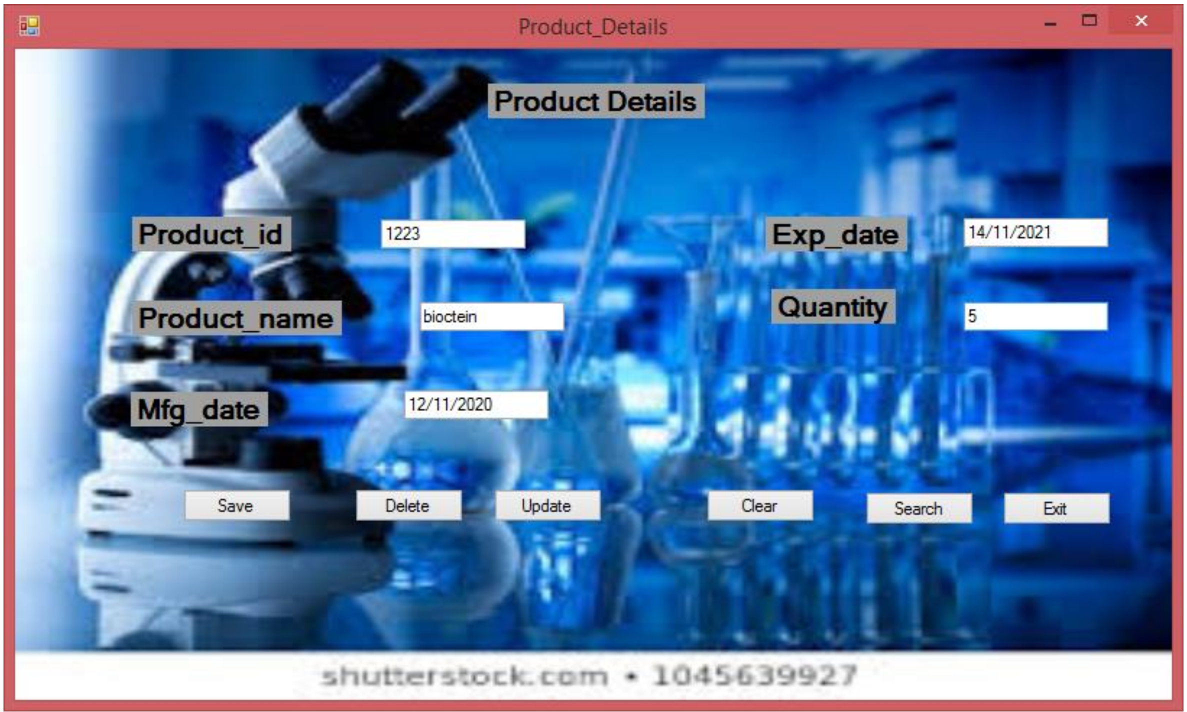
CODING

```
Imports System.Data
Imports System.Data.SqlClient
Public Class Product_Details
    Dim con1 As New SqlConnection("Data Source=MADHUKAR\SQLEXPRESS;Initial
Catalog=mydb;Integrated Security=True")
    Dim ad As SqlDataAdapter = New SqlDataAdapter("select * from Product_info", con1)
    Dim ds As DataSet
    Dim dv As DataView
    Dim abc As Char
    Dim cm As CurrencyManager
    Dim id As Object
    Dim rd, rd1 As SqlDataReader
    Dim cmd As New SqlCommand("select * from Product_info", con1)
    Dim cmd1 As SqlCommand
    Dim count As Integer
    Dim i As String
    Private Sub filldata()
        ds = New DataSet
        If con1.State = 1 Then con1.Close()
        con1.Open()
        ad.Fill(ds, "Product_info")
        dv = New DataView(ds.Tables("Product_info"))
        cm = CType(Me.BindingContext(dv), CurrencyManager)
    End Sub
    Private Sub bindfield()
        TextBox1.DataBindings.Clear()
        TextBox2.DataBindings.Clear()
        TextBox3.DataBindings.Clear()
        TextBox4.DataBindings.Clear()
        TextBox5.DataBindings.Clear()

        TextBox1.DataBindings.Add("text", dv, "Product_id")
        TextBox2.DataBindings.Add("text", dv, "Product_nm")
        TextBox3.DataBindings.Add("text", dv, "Mfg_date")
        TextBox4.DataBindings.Add("text", dv, "Exp_date")
        TextBox5.DataBindings.Add("text", dv, "Quantity")
    End Sub
    Private Sub Product_Details_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        filldata()
        bindfield()
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox4.Text
= "" Or TextBox5.Text = "" Then
            MsgBox("Please fill all fields with appropriate data", MsgBoxStyle.Information,
"Save")
        Else
```

```
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            cmd.CommandText = "Insert into Product_info
values(@Product_id,@Product_nm,@Mfg_date,@Exp_date,@Quantity)"
            cmd.Parameters.AddWithValue("@Product_id", TextBox1.Text)
            cmd.Parameters.AddWithValue("@Product_nm", TextBox2.Text)
            cmd.Parameters.AddWithValue("@Mfg_date", TextBox3.Text)
            cmd.Parameters.AddWithValue("@Exp_date", TextBox4.Text)
            cmd.Parameters.AddWithValue("@Quantity", TextBox5.Text)
            Try
                cmd.ExecuteNonQuery()
                MsgBox("Records Saved Successfully", MsgBoxStyle.Information, "Save")
            Catch ex As Exception
                MsgBox("Error:" & ex.Source & ":" & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error!!")
            End Try
            con1.Close()
        End If
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        If TextBox1.Text = "" Then
            MsgBox("Please enter Product id which you want to delete",
MsgBoxStyle.Information, "Delete")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            If MsgBox("Are you sure?", MsgBoxStyle.YesNo Or MsgBoxStyle.Question, "Delete")
= MsgBoxResult.Yes Then
                cmd.CommandText = "Delete from Product_info where Product_id like
@Product_id"
                cmd.Parameters.AddWithValue("@Product_id", TextBox1.Text)
                cmd.Parameters.AddWithValue("@Product_nm", TextBox2.Text)
                cmd.Parameters.AddWithValue("@Mfg_date", TextBox3.Text)
                cmd.Parameters.AddWithValue("@Exp_date", TextBox4.Text)
                cmd.Parameters.AddWithValue("@Quantity", TextBox5.Text)

                Try
                    cmd.ExecuteNonQuery()
                    MsgBox("Records Saved Successfully", MsgBoxStyle.Information, "Delete")
                Catch ex As Exception
                    MessageBox.Show(ex.Message)
                End Try
            End If
            con1.Close()
        End If
    End Sub
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        If TextBox1.Text = "" Then
            MsgBox("Please enter Product_id which you want to modify",
MsgBoxStyle.Information, "Update")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            cmd.CommandText = "Update Product_info set
Product_nm=@Product_nm,Mfg_date=@Mfg_date,Exp_date=@Exp_date,Quantity=@Quantity where
Product_id like @Product_id"
            cmd.Parameters.AddWithValue("@Product_id", TextBox1.Text)
            cmd.Parameters.AddWithValue("@Product_nm", TextBox2.Text)
            cmd.Parameters.AddWithValue("@Mfg_date", TextBox3.Text)
            cmd.Parameters.AddWithValue("@Exp_date", TextBox4.Text)
            cmd.Parameters.AddWithValue("@Quantity", TextBox5.Text)
            Try
                cmd.ExecuteNonQuery()
                MsgBox("Records Saved Successfully", MsgBoxStyle.Information, "Update")
            Catch ex As Exception
                MessageBox.Show(ex.Message)
            End Try
            con1.Close()
        End If
    End Sub
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
        If con1.State = 1 Then con1.Close()
        con1.Open()
        Dim cmd As SqlCommand = New SqlCommand("select * from Product_info where
Product_id=" & TextBox1.Text & "", con1)
        Dim rd As SqlDataReader = cmd.ExecuteReader
        If Not rd.HasRows Then
            MsgBox("The Data does not exist")
        Else
            MsgBox("The Record is exist")
            rd.Read()
            TextBox1.Text = rd.Item(0)
            TextBox2.Text = rd.Item(1)
            TextBox3.Text = rd.Item(2)
            TextBox4.Text = rd.Item(3)
            TextBox5.Text = rd.Item(4)
            rd.Close()
        End If
        con1.Close()
    End Sub
    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click
        Me.Dispose()
    End Sub
    Private Sub TextBox1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles TextBox1.Click
        TextBox1.Clear()
        If con1.State = 1 Then con1.Close()
        con1.Open()
        rd = cmd.ExecuteReader()
        con1.Close()
    End Sub
    Private Sub TextBox2_KeyPress(ByVal eventSender As System.Object, ByVal eventArgs As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress
```

```
        Dim KeyAscii As Short = Asc(eventArgs.KeyChar)
        If Not ((KeyAscii < 9 And KeyAscii > 7) Or (KeyAscii >= 65 And KeyAscii <= 90) Or
(KeyAscii >= 97 And KeyAscii <= 122) Or (KeyAscii >= 31 And KeyAscii < 33)) Then
            MsgBox("Sorry,Only Alphabets Allowed", MsgBoxStyle.Critical, "Name")
            KeyAscii = 0
        End If
        eventArgs.KeyChar = Chr(KeyAscii)
        If KeyAscii = 0 Then
            eventArgs.Handled = True
        End If
    End Sub
End Class
```

TEST RESULT



Product_Details

Product Details

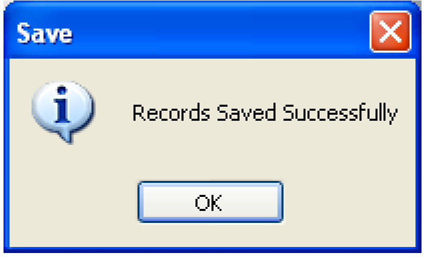
Product_id 1223

Product_name bioctein

Mfg_date 12/11/2020

Exp_date 14/11/2021

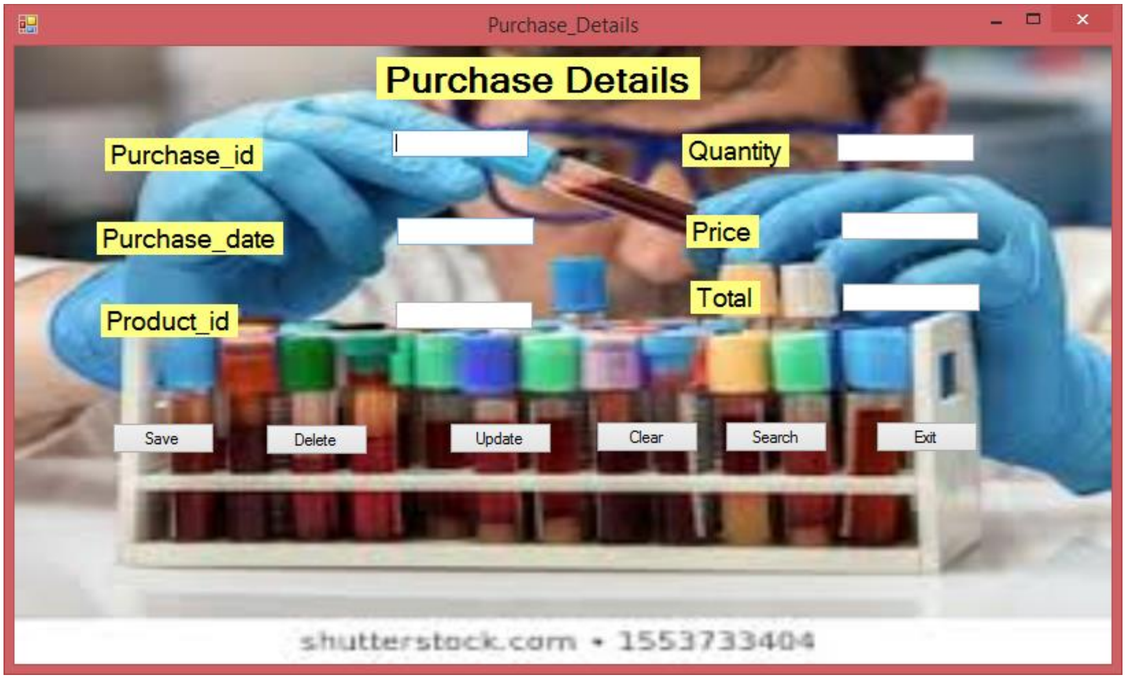
Quantity 5



Save

Records Saved Successfully

9] Form Name: - PURCHASE DETAILS



TESTDATA

Program ID:- Purchase Form

This is used to read & write an information of purchase details.

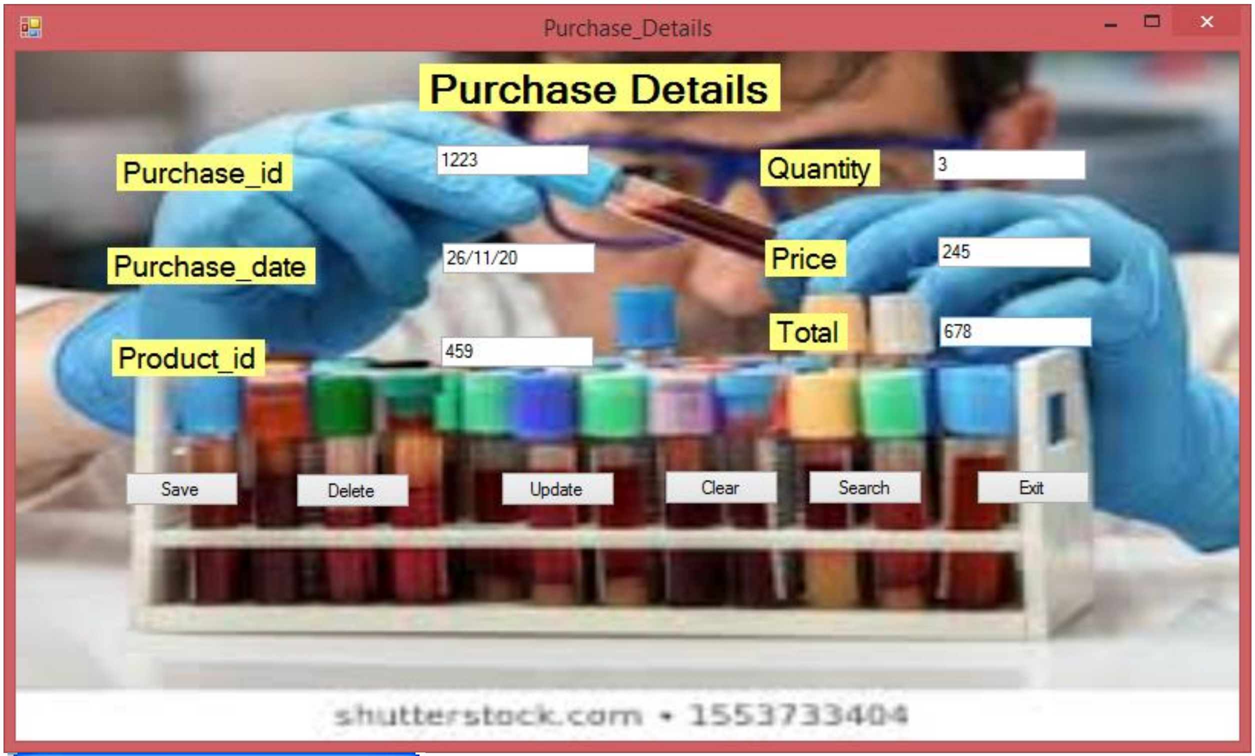
PROCEDURE	*DESCRIPTION*
Btnfirst	*Go to the first record in the database.*
Btnprev	*Go to the previous record in the database.*
Btnnext	*Go to the next record in the database.*
Btnlast	*Go to the last record in the database.*
Btnadd	*You can add the new Purchase Product record.*
Btnsave	*Save the all record in the database.*
Btndelete	*You can delete the record.*
Btncancel	*You can cancel the record.*
Btnsearch	*Go to the Search form to search details of particular Purchase Product record.*
Btnreport	*Go to the Report form to display the records.*
Btnexit	*Move back to the MDI Form.*

CODING

```
Imports System.Data
Imports System.Data.SqlClient
Public Class Purchase_Details
    Dim con1 As New SqlConnection("Data Source=MADHUKAR\SQLEXPRESS;Initial
Catalog=mydb;Integrated Security=True")
    Dim ad As SqlDataAdapter = New SqlDataAdapter("select * from Purchase_info", con1)
    Dim ds As DataSet
    Dim dv As DataView
    Dim abc As Char
    Dim cm As CurrencyManager
    Dim id As Object
    Dim rd, rd1 As SqlDataReader
    Dim cmd As New SqlCommand("select * from Purchase_info", con1)
    Dim cmd1 As SqlCommand
    Dim count As Integer
    Dim i As String
    Private Sub filldata()
        ds = New DataSet
        If con1.State = 1 Then con1.Close()
        con1.Open()
        ad.Fill(ds, "Purchase_info")
        dv = New DataView(ds.Tables("Purchase_info"))
        cm = CType(Me.BindingContext(dv), CurrencyManager)
    End Sub
    Private Sub bindfield()
        TextBox1.DataBindings.Clear()
        TextBox2.DataBindings.Clear()
        TextBox3.DataBindings.Clear()
        TextBox4.DataBindings.Clear()
        TextBox5.DataBindings.Clear()
        TextBox6.DataBindings.Clear()

        TextBox1.DataBindings.Add("text", dv, "Purchase_id")
        TextBox2.DataBindings.Add("text", dv, "Purchase_date")
        TextBox3.DataBindings.Add("text", dv, "Product_nm")
        TextBox4.DataBindings.Add("text", dv, "Quantity")
        TextBox5.DataBindings.Add("text", dv, "Price")
        TextBox6.DataBindings.Add("text", dv, "Total")
    End Sub
    Private Sub Purchase_Details_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        filldata()
        bindfield()
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""

    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox4.Text
= "" Or TextBox5.Text = "" Or TextBox6.Text = "" Then
            MsgBox("Please fill all fields with appropriate data", MsgBoxStyle.Information,
"Save")
```

```
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            cmd.CommandText = "Insert into Purchase_info
values(@Purchase_id,@Purchase_date,@Product_nm,@Quantity,@Price,@Total)"
            cmd.Parameters.AddWithValue("@Purchase_id", TextBox1.Text)
            cmd.Parameters.AddWithValue("@Purchase_date", TextBox2.Text)
            cmd.Parameters.AddWithValue("@Product_nm", TextBox3.Text)
            cmd.Parameters.AddWithValue("@Quantity", TextBox4.Text)
            cmd.Parameters.AddWithValue("@Price", TextBox5.Text)
            cmd.Parameters.AddWithValue("@Total", TextBox6.Text)

            Try
                cmd.ExecuteNonQuery()
                MsgBox("Records Saved Successfully", MsgBoxStyle.Information, "Save")
            Catch ex As Exception
                MsgBox("Error:" & ex.Source & ":" & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error!!")
            End Try
            con1.Close()
        End If
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        If TextBox1.Text = "" Then
            MsgBox("Please enter Purchase id which you want to delete",
MsgBoxStyle.Information, "Delete")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            If MsgBox("Are you sure?", MsgBoxStyle.YesNo Or MsgBoxStyle.Question, "Delete")
= MsgBoxResult.Yes Then
                cmd.CommandText = "Delete from Purchase_info where Purchase_id like
@Purchase_id"
                cmd.Parameters.AddWithValue("@Purchase_id", TextBox1.Text)
                cmd.Parameters.AddWithValue("@Purchase_date", TextBox2.Text)
                cmd.Parameters.AddWithValue("@Product_nm", TextBox3.Text)
                cmd.Parameters.AddWithValue("@Quantity", TextBox4.Text)
                cmd.Parameters.AddWithValue("@Price", TextBox5.Text)
                cmd.Parameters.AddWithValue("@Total", TextBox6.Text)
                Try
                    cmd.ExecuteNonQuery()
                    MsgBox("Records Saved Successfully", MsgBoxStyle.Information, "Delete")
                Catch ex As Exception
                    MessageBox.Show(ex.Message)
                End Try
            End If
            con1.Close()
        End If
    End Sub
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        If TextBox1.Text = "" Then
            MsgBox("Please enter Purchase_id which you want to modify",
MsgBoxStyle.Information, "Update")
        Else
            If con1.State = 1 Then con1.Close()
            con1.Open()
            Dim cmd As SqlCommand = New SqlCommand
            cmd.Connection = con1
            cmd.CommandText = "Update Purchase_info set 
Purchase_date=@Purchase_date,Product_nm=@Product_nm,Quantity=@Quantity,Price=@Price,Total=@
Total where Purchase_id like @Purchase_id"
            cmd.Parameters.AddWithValue("@Purchase_id", TextBox1.Text)
            cmd.Parameters.AddWithValue("@Purchase_date", TextBox2.Text)
            cmd.Parameters.AddWithValue("@Product_nm", TextBox3.Text)
            cmd.Parameters.AddWithValue("@Quantity", TextBox4.Text)
            cmd.Parameters.AddWithValue("@Price", TextBox5.Text)
            cmd.Parameters.AddWithValue("@Total", TextBox6.Text)
            Try
                cmd.ExecuteNonQuery()
                MsgBox("Records Saved Successfully", MsgBoxStyle.Information, "Update")
            Catch ex As Exception
                MessageBox.Show(ex.Message)
            End Try
            con1.Close()
        End If
    End Sub
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) 
Handles Button4.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
    End Sub
    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) 
Handles Button5.Click
        If con1.State = 1 Then con1.Close()
        con1.Open()
        Dim cmd As SqlCommand = New SqlCommand("select * from Purchase_info where 
Purchase_id=" & TextBox1.Text & "", con1)
        Dim rd As SqlDataReader = cmd.ExecuteReader
        If Not rd.HasRows Then
            MsgBox("The Data does not exist")
        Else
            MsgBox("The Record is exist")
            rd.Read()
            TextBox1.Text = rd.Item(0)
            TextBox2.Text = rd.Item(1)
            TextBox3.Text = rd.Item(2)
            TextBox4.Text = rd.Item(3)
            TextBox5.Text = rd.Item(4)
            TextBox6.Text = rd.Item(5)
            rd.Close()
        End If
        con1.Close()
    End Sub
    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) 
Handles Button6.Click
        Me.Dispose()
    End Sub
    Private Sub TextBox1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) 
Handles TextBox1.Click
        TextBox1.Clear()
        If con1.State = 1 Then con1.Close()
        con1.Open()
        rd = cmd.ExecuteReader()
        con1.Close()
    End Sub

    Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox3.TextChanged

    End Sub

End class
```

TESTRESULT



Purchase_Details

Purchase Details

Purchase_id 1223

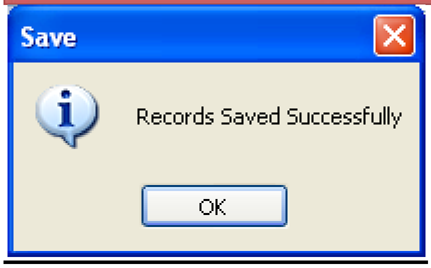
Purchase_date 26/11/20

Product_id 459

Quantity 3

Price 245

Total 678



Save

Records Saved Successfully

System Implementation

1] Install visual studio 2019 and Sql server 2008

2] Copy project files on the machine.

3] Execute the project.

FutureEnhancements

- ✓ *Multiuser Functionality*
- ✓ *Alarm System for Medicines Expired*
- ✓ *Online Update of Medicine*
- ✓ *Nearby Pathology Lab Tracker*

Conclusion

A] System help to all the data about supplier to lab stored in computer . there is no need to do paper work

B] Data is going to be preserved carefully for longer period ,hence proper backup is required otherwise there is chance of losing enterprise.

C] There are also chances that calculation error can occur which can huge problem.

D] The system allows the user to add new supplier, patient , test results to record.

E] This information can be modified , deleted as required.

References and Bibliography

1. ***Mastering Micosoft visual Basic 2008***
   Author:- *Evangelos Petroutsos*
   Publisher: *Sybex*

2. ***Murach's visual basic 2008***
   Author:- *Anne Boehm*
   Publisher:- *Murach*

3. ***Mastering visual studio 2010***
   Author: *Joe Mayo*
   Publisher: *MCGraw*

4. ***Mastering visual studio 2019***
   Developer*: Microsoft*