1 INTRODUCTION

1.1 PROJECT DESCRIPTION:

Dairy management system is application software to maintain day to day transaction in milk dairy. This software helps to register all supplier, buyer details, purchaser and sales details. It became tough for dairy owner to manage all dairy work manually. To reduce manual work, a dairy management system can help to make day to day dairy related activities easier. The project entitled dairy management system has been developing with the aim of providing a tool, which will prove efficient enough of milk and save organization time as well resource. This project gives new the user which helps to save their time.

1.2 PROBLEM STATEMENT:

As we know manual system are quite tedious and time consuming and less effective and also not accurate in comparison to computerised system. Manual system cannot be accessed anywhere and anytime which is a big problem to overcome and mismatch in demand and supply some time demand for the product is more as compared to supply. In India, only of 15 % dairy management the organization in are organized and other 85% are unorganised. Long wasting time and inefficiency is the biggest problem sometimes dairy is not managed properly 50 the farmer has to wait for a long time for selling their product.

1.3 EXISTING SYSTEM:

Manual systems consumes more time where it also does. Not provide 24/7 service.

Limitations:

- Time Consuming.
- Less accurate.
- Less efficient
- Slow data processing.
- Lot of paper work.
- Not user friendly

1.4 PROPOSED SYSTEM:

This system helps in managing data so, it makes the maintenance easy. The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitation of existing system. The system provides security and reduces the manual work and less time consuming. This also helps to record the total milk supplied by each supplier is recorded separately and recognize all the customers and suppliers etc.

1.5 BENEFITS OF PROPOSED SYSTEM:

- Fast and accurate.
- Backup is maintained.
- High data processing.
- Good level of security.
- It can be accessed at anytime and anywhere.

1.6 OBJECTIVES:

- Reduce the errors done through manual system.
- Maintain daily inventory report and records.
- Day to day update of dairy. Of members.
- Quick access to all records.
- Identify top performing assets.
- Data can be easily updated or altered.
- User friendly GUI.

1.7 MODULES:

1. Admin
2. User

Admin

Admin plays an important role in dairy management system. The admin has the unique login and password which completely controls the working. He will verify the details given by the user in the register form. Admin can make payments to the user for his product or item. He can modify the details of the customers.

User

The user can easily register a fresh form in the dairy management system. He can update his form for the admin to recognize by providing his name, contact, address etc along with the dairy profiles. They will be having a unique id for each user. He can view the bill generated by the admin.

2 SYSTEM SPECIFICATION

2.1 HARDWRE SPECIFICATION:

Computer Processor	: Dual Core
Processor Speed	: 2.8 GHz Processor
Hard Disk	: 40GB
RAM	: min 745MB
Monitor	: 14” colour or black and white

2.2 SOFTWARE SPECIFICATUION:

Operating System	: window 10 enterprise/XP/vista/seven
Front End	: Visual Basic 6.0
Back End	: Microsoft SQL Server
Language Used	: Visual Basic
Documentation	: Microsoft Office 2007

3 TOOLS

3.1 FRONT END

Visual Basic

Visual basic is a third-generation event-driven programming language from Microsoft known for its Component Object Model (COM) programming model first released in 1991 and declared legacy during 2008. Microsoft intended Visual Basic to be relatively easy to learn and use.

All the controls in the Toolbox except the Pointer are objects in Visual Basic. These objects have associated properties, methods and events.

Real world objects are loaded with properties. For example, a flower is loaded certain colour, shape and fragrance. Similarly programming objects are loaded with properties. A property is a named attribute of a programming object. Properties define the characteristics of an object such as Size, Color etc. or sometimes the way in which it behaves. For example, a Textbox accepts properties such as Enabled, Font, Multiline, Text, Visible, Width, etc.

- Enables property allows the Textbox to be enabled or disabled at run time depending on the condition set to True or False.
- Font property sets a particular font in the Textbox.
- Multiline property allows the Textbox to accept and display multiple lines at run time.
- Text property of the Textbox control sets a particular text in the control.
- Visible property is used to hide the object at run time.
- Width property sets the Textbox to the desired width at design time.

The properties that are discussed above are design-time properties that can be set at the design time by selecting the Properties Window. But certain properties cannot be set at design time. For example, the Current x and Current y properties of a Form cannot be set at the design time.

A method is an action that can be performed on objects. For example, a cat is an object. Its properties might include long white hair, blue eyes, 3 pounds weight etc. A complete definition of cat must only encompass on its looks, but should also include a complete itemization of its activities. Therefore, a cat's methods might be move, jump, play, breath etc.

Similarly, in object-oriented programming, a method is a connected or built-in procedure, a block of code that can be invoked to impart some action on a particular object. A method requires an object to provide them with a context.

Characteristics of visual basic

- Modern,
- General purpose.
- Object oriented.
- Component oriented.
- Easy to learn.
- Structured language.
- It produces efficient programs.
- It can be compiled on a variety of computer platforms.

The Visual Basic program also includes features like "IntelliSense" and "Code Snippets," which automatically generate code for visual objects added by the programmer.

Event Driven Programming

Visual Basic programs are built around events. Events are various things that can happen in a program. This will become clearer when studied in contrast to procedural programming. In procedural languages, an application is written is executed by checking for the program logically through the program statements, one after another. For a temporary phase, the control may be transferred to some other point in a program. While in an event driven application, the program statements are executed only when a particular event calls a specific part of the code that is assigned to the event.

Let us consider a Textbox control and a few of its associated events to understand the concept of event driven programming. The Textbox control supports various events such as Change, Click, Mouse Move and many more that will be listed in the Properties dropdown list in the code window for the Textbox control. We will look into a few of them as given below.

The code entered in the Change event fires when there is a change in the contents of the Textbox. The Click event fires when the Textbox control is clicked.

3.2 BACK END

SQL Server Management Studio (SSMS) is a software application first launched with Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components within Microsoft SQL Server. It is the successor to the Enterprise Manager in SQL 2000 or before. The tool includes both script editors and graphical tools which work with objects and features of the server.

Earlier version of Microsoft SQL Server Management Studio showing a query, the results of the query and the Object Explorer pane while connected to a SQL Server database engine. SSMS is one of the SQL Server management tools, regardless of your location, used for designing queries and managing databases and data warehouses via personal computer or Cloud.

In reality, SSMS is an integrated environment that provides tools for configuring, monitoring, and administering SQL Server instances and databases.

A central feature of SSMS is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server.[5] It also shipped a separate Express edition that could be freely downloaded, however recent versions of SSMS are fully capable of connecting to and manage any SQL Server Express instance. Microsoft also incorporated backwards compatibility for older versions of SQL Server thus allowing a newer version of SSMS to connect to older versions of SQL Server instances. It also comes with Microsoft SQL Server Express 2012, or users can download it separately.

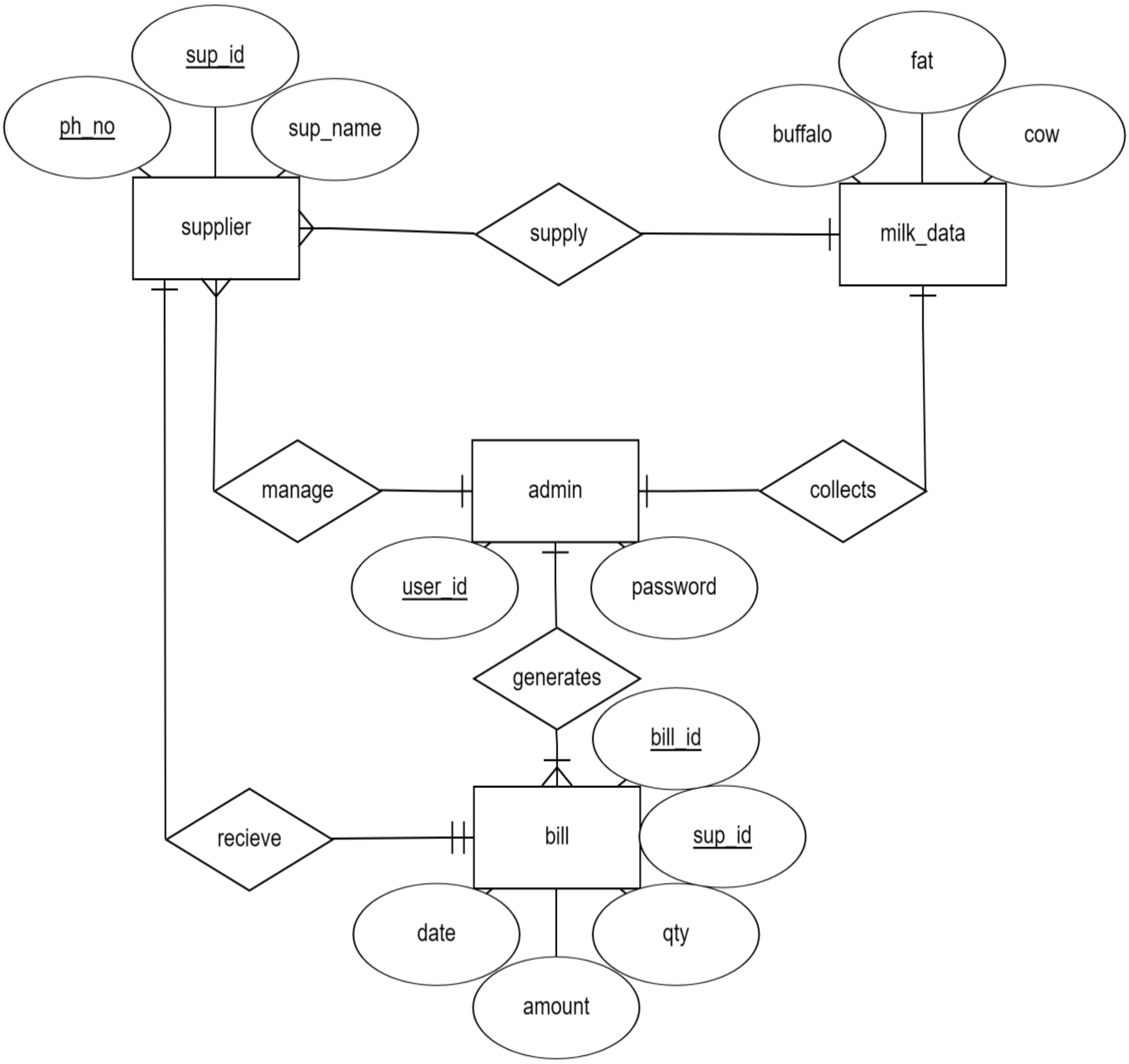
Starting from version 11, the application was based on the Visual Studio 2010 shell, using WPF for the user interface. Versions 18 and after are based on the Visual Studio 2017 Isolated Shell.

In June 2015, Microsoft announced their intention to release future versions of SSMS independently of SQL Server database engine releases.

4 DESIGN SPECIFICATION

4.1 ENTITY RELATIONSHIP DIAGRAM

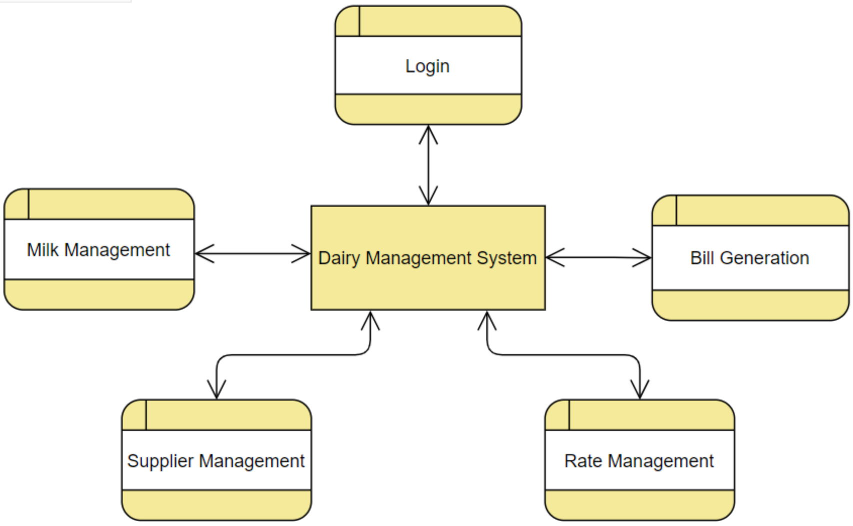
It is a graphical representation of the entities and the relationships between them. ER diagram are a useful medium to achieve a common understanding of data among user and application developers.



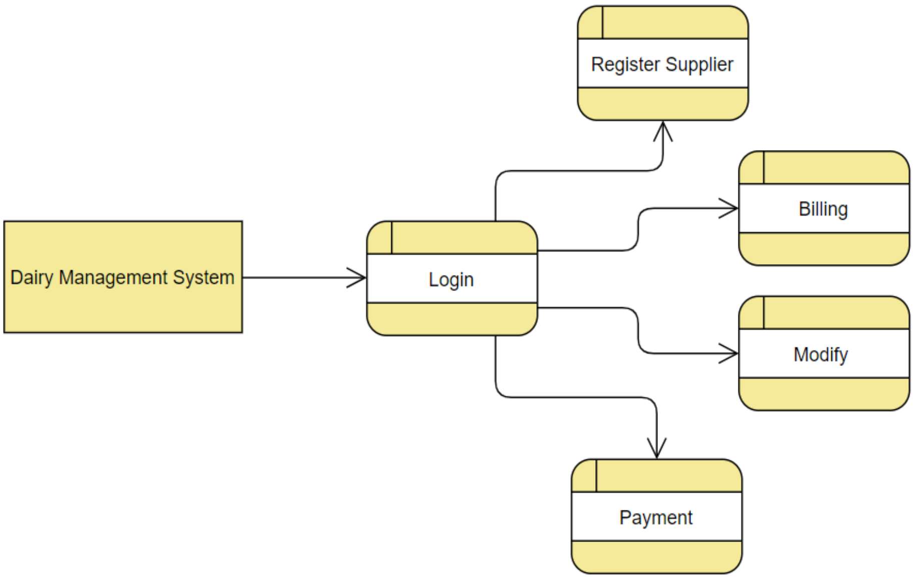
4.2 DATA FLOW DIAGRAM

The data flow diagrams are pictorial or graphical representation of the outline of the system study.

DFD Level 0:



DFD Level 1:



4.3 DATA DICTIONARY

A data dictionary is a file or a set of files that contains a database's metadata. The data dictionary contains records about other objects in the database, such as data ownership, data relationships to other objects and other data.

The data dictionary is a crucial component of any relational database, ironically, because of its importance; it is invisible to most database users. Typically, only database administrators interact with the data dictionary.

Supplier Login

Column Name	Data Type
username	varchar()
password	nvarchar()

Register

Column Name	Data Type
u_name	varchar()
u_id	nvarchar()
cadd	varchar()
cpno	nvarchar()
milk	varchar()
pass	nvarchar()
acne	nvarchar()

Billing

Column_Name	Data Type
Bill_id	nvarchar
u_id	nvarchar()
date	Date
milk	nvarchar()

quantity	Int
fat	int
rate	int
acne	nvarchar()
tot	Int

Admin Login

Column Name	Data Type
Admin_name	varchar()
password	nvarchar()

5 CODING

5.1 WELCOME FORM:

```
Private Sub admin_Click ()

admin_login.Show

WindowState = 2

End Sub

Private Sub user_Click ()

user_login.Show

WindowState = 2

End Sub
```

5.2 ADMIN FORM:

```
Private Sub login_Click ()

On Error GoTo errmsg

Adodc1.Refresh

If Adodc1.Recordset.EOF Then

admin_name.SetFocus

Adodc1.Recordset.AddNew

End If

If admin_name.Text = "admin" And pass.Text = "12345" Then

MsgBox ("Login Successfull")

admin_page.Show

Else
```

```
MsgBox "Sorry Incorrect Admin name or Password"

End If

If (admin_name.Text = "" Or pass.Text = "") Then

MsgBox "Please fill the empty fields"

Else

Adodc1.Recordset.Fields ("username") = admin_name.Text

Adodc1.Recordset.Fields ("password") = pass.Text

Adodc1.Recordset.Update

End If

Exit Sub

errmsg:

MsgBox Err.Description

End Sub

Private Sub exit_Click ()

Unload Me

WindowState = 2

Welcome.Show

End Sub

Private Sub forgot_Click ()

forgot1.Show

WindowState = 2

End Sub
```

5.3 USER LOGIN

```
Private Sub exit_Click()

Unload Me

Welcome.Show

WindowState = 2

End Sub

Private Sub form_load()

Txtuser = ""

txtpass = ""

End Sub

Private Sub login_Click ()

Adodc1.Refresh

On Error GoTo errmsg

With Adodc1.Recordset

.MoveFirst

.Find "u_name='" & u_name.Text & "'"

If .EOF = True Then

MsgBox "INCORRECT USERNAME OR PASSWORD!!"

ElseIf (pass.Text = Adodc1.Recordset.Fields ("pass")) Then

MsgBox "YOU ARE SUCCESSFULLY LOGGED IN!!"

Unload Me

End If

Exit Sub
```

```
errmsg:
MsgBox Err.Description
End Sub
Private Sub registe_Click()
Unload Me
register.Show
WindowState = 2
End Sub
```

5.4 REGISTER FORM

```
Private Sub cmdadd_Click ()
Dim id As Integer
Dim id1 As String
u_name.Enabled = True
u_id.Enabled = True
cadd.Enabled = True
cpno.Enabled = True
Selec.Enabled = True
pass.Enabled = True
acno.Enabled = True
u_name = ""
u_id = ""
cadd = ""
cpno = ""
```

```
pass = ""

acne = ""

On Error GoTo errmsg

Adodc1.Refresh

If Adodc1.Recordset.EOF Then

u_id = "C101"

Else

Adodc1.Recordset.MoveLast

id1 = Adodc1.Recordset ("u_id")

id = Mid(id1, 2, 4) + 1

u_id = "C" & id

End If

u_name.SetFocus

Adodc1.Recordset.AddNew

Exit Sub

errmsg:

MsgBox Err.Description

End Sub

Private Sub cmdexit_Click()

Adodc1.Recordset.Close

Unload Me

admin_page.Show

End Sub
```

```
Private Sub cmdsave_Click ()

On Error GoTo errmsg

If Not Len (cpno) = 10 Then

cpno = ""

MsgBox "Enter a valid phone number"

End If

 If (u_name.Text = "" Or u_id.Text = "" Or cadd.Text = "" Or cpno.Text = "" Or pass.Text = "" Or acno.Text = "") Then

 MsgBox "please fill the empty fields"

Else

Adodc1.Recordset.Fields ("u_name") = u_name.Text

Adodc1.Recordset.Fields ("u_id") = u_id.Text

Adodc1.Recordset.Fields ("cadd") = cadd.Text

Adodc1.Recordset.Fields ("cpno") = cpno.Text

Adodc1.Recordset.Fields ("milk") = Selec.Text

Adodc1.Recordset.Fields ("pass") = pass.Text

Adodc1.Recordset.Fields ("acne") = acno.Text

Adodc1.Recordset.Update

MsgBox "New Supplier details added"

End If

Exit Sub

errmsg:

MsgBox Err.Description
```

```
End Sub
Private Sub form_load ()
u_name = ""
u_id = ""
cadd = ""
cpno = ""
pass = ""
acne = ""
u_name.Enabled = False
u_id.Enabled = False
cadd.Enabled = False
cpno.Enabled = False
Selec.Enabled = False
pass.Enabled = False
acno.Enabled = False
End Sub
Private Sub cpno_Click ()
If Not Len (cpno) = 10 Then
MsgBox "enter a valid phone no"
cpno = ""
End If
End Sub
Private Sub u_name_keypress (keyascii As Integer)
```

```
If keyascii >= 65 And keyascii <= 90 Or keyascii >= 97 And keyascii >= 65 And keyascii <=
122 Or keyascii = 8 Or keyascii = 32 Then

Else

Keyascii = 0

MsgBox "Dear Admin Please Enter Alphabets Only"

End If

End Sub

Private Sub cpno_keypress(keyascii As Integer)

If keyascii >= 48 And keyascii <= 57 Or keyascii = 8 Then

Else

Keyascii = 0

MsgBox "Dear Admin Please Enter Digits Only"

End If

End Sub

Private Sub Selec_Change ()

combo1.clear

On Error GoTo errmsg

Adodc1.Refresh

With Adodc1.Recordset

.MoveFirst

While Not .EOF

combo1.AddItem .Fields ("milk")

.MoveNext
```

```
Wend
End With
Exit Sub
errmsg:
MsgBox Err.Description
End Sub
Private Sub acno_keypress (keyascii As Integer)
If keyascii >= 48 And keyascii <= 57 Or keyascii = 8 Then
Else
keyascii = 0
MsgBox "Dear Admin Please Enter Digits Only"
End If
End Sub
Private Sub acno_Click()
If Not Len(acne) = 10 Then
MsgBox "enter a valid account no"
acne = ""
End If
End Sub
```

5.5 ADMIN PAGE

```
Private Sub Command1_Click()
Unload Me
rate.Show
```

```
WindowState = 2

End Sub

Private Sub Command2_Click()

Unload Me

register.Show

WindowState = 2

End Sub

Private Sub deposit_Click()

Unload Me

Billing.Show

WindowState = 2

End Sub

Private Sub exit_Click()

Unload Me

Welcome.Show

WindowState = 2

End Sub

Private Sub report_Click()

Unload Me

payment.Show

WindowState = 2

End Sub
```

```
Private Sub modify_Click()

Unload Me

modify.Show

WindowState = 2

End Sub

Private Sub mod_Click()

Unload Me

modify.Show

WindowState = 2

End Sub

Private Sub payment_Click()

Unload Me

payment1.Show

WindowState = 2

End Sub
```

5.6 MODIFY FORM

```
Private Sub cmdadd_Click()

Dim id As Integer

Dim id1 As String

u_name.Enabled = True

u_id.Enabled = True

cadd.Enabled = True

cpno.Enabled = True
```

```
Selec.Enabled = True

pass.Enabled = True

acno.Enabled = True

u_name = ""

u_id = ""

cadd = ""

cpno = ""

pass = ""

acne = ""

u_name.SetFocus

Adodc1.Recordset.AddNew

Exit Sub

errmsg:

MsgBox Err.Description

End Sub

Private Sub cmdexit_Click()

Unload Me

admin_page.Show

End Sub

Private Sub cmdfind_Click()

Dim id1 As String

Adodc1.Refresh

id1 = InputBox("Enter the Supplier ID to be modified")
```

```
Adodc1.Recordset.Find "u_id='" & id1 & "'"

If Adodc1.Recordset.EOF Then

MsgBox " Customer id not found"

Else

u_name = Adodc1.Recordset("u_name")

u_id = Adodc1.Recordset("u_id")

cadd = Adodc1.Recordset("cadd")

cpno = Adodc1.Recordset("cpno")

Selec = Adodc1.Recordset("milk")

pass = Adodc1.Recordset("pass")

acne = Adodc1.Recordset("acne")

End If

Exit Sub

errmsg:

MsgBox Err.Description

End Sub

Private Sub cpno_Click()

If Not Len(cpno) = 10 Then

MsgBox "enter a valid phone no"

cpno = ""

End If

End Sub
```

```
Private Sub cmdsave_Click()
On Error GoTo errmsg
If Not Len(cpno) = 10 Then
MsgBox "enter a valid phone no"
cpno = ""
End If
If (u_name.Text = "" Or u_id.Text = "" Or cadd.Text = "" Or cpno.Text = "" Or Selec.Text = "" Or pass.Text = "" Or acno.Text = "") Then
MsgBox "Please fill the empty fields"
Else
Adodc1.Recordset.Fields ("u_name") = u_name.Text
Adodc1.Recordset.Fields ("u_id") = u_id.Text
Adodc1.Recordset.Fields ("cadd") = cadd.Text
Adodc1.Recordset.Fields ("cpno") = cpno.Text
Adodc1.Recordset.Fields ("milk") = Selec.Text
Adodc1.Recordset.Fields ("pass") = pass.Text
Adodc1.Recordset.Fields ("acne") = acno.Text
Adodc1.Recordset.Update
MsgBox "Supplier details modified sucessfully"
End If
Exit Sub
errmsg:
MsgBox Err.Description
```

```
End Sub

Private Sub delete_Click()

On Error GoTo errmsg

Dim confirm As Integer

Adodc1.Refresh

Dim bno As String

bno = u_id.Text

confirm = MsgBox("Do you really want to delete this supplier y/n?", vbYesNo + vbInformation)

If confirm = vbYes Then

Adodc1.Recordset.MoveFirst

Adodc1.Recordset.Find "u_id='" & bno & "'"

Adodc1.Recordset.delete

DataGrid1.Refresh

MsgBox "This supplier is deleted successfully"

Else

MsgBox "Sorry wrong supplier id"

End If

Exit Sub

errmsg:

MsgBox Err.Description

End Sub
```

```
Private Sub form_load()

u_name = ""

u_id = ""

cadd = ""

cpno = ""

pass = ""

acne = ""

u_name.Enabled = False

u_id.Enabled = False

cadd.Enabled = False

cpno.Enabled = False

Selec.Enabled = False

pass.Enabled = False

acno.Enabled = False

End Sub

Private Sub cpno_keypress(keyascii As Integer)

If keyascii >= 48 And keyascii <= 57 Or keyascii = 8 Then

Else

keyascii = 0

MsgBox " Dear Admin Please Enter Digits Only"

End If

End Sub
```

```
Private Sub u_name_keypress(keyascii As Integer)

If keyascii >= 65 And keyascii <= 90 Or keyascii >= 97 And keyascii >= 65 And keyascii <= 122 Or keyascii = 8 Or keyascii = 32 Then

Else

keyascii = 0

MsgBox " Enter only alphabets"

End If

End Sub

Private Sub Selec_Change()

combo1.clear

On Error GoTo errmsg

Adodc1.Refresh

With Adodc1.Recordset

.MoveFirst

While Not .EOF

combo1.AddItem .Fields("milk")

.MoveNext

Wend

End With

Exit Sub

errmsg:

MsgBox Err.Description

End Sub
```

5.7 BILLING FORM

```
Private Sub cmdfind_Click()

u_id.Enabled = False

Selec.Enabled = False

acno.Enabled = False

Dim id1 As String

Adodc2.Refresh

Adodc2.Recordset.MoveFirst

id1 = InputBox("Enter the Supplier ID")

Adodc2.Recordset.Find "u_id='" & id1 & "'"

If Adodc2.Recordset.EOF Then

MsgBox "Supplier id not found"

Else

u_id = Adodc2.Recordset("u_id")

Selec = Adodc2.Recordset("milk")

acne = Adodc2.Recordset("acne")

End If

Exit Sub

errmsg:

MsgBox Err.Description

End Sub

Private Sub cmdadd_Click()

Dim id As Integer
Dim id1 As String

bill.Enabled = True

u_id.Enabled = True

dat.Enabled = True

Selec.Enabled = True

quantity.Enabled = True

fat.Enabled = True

rate.Enabled = True

acno.Enabled = True

tot.Enabled = True

bill = ""

u_id = ""

dat = ""

fat = ""

rate = ""

acne = ""

tot = ""

On Error GoTo errmsg

Adodc1.Refresh

If Adodc1.Recordset.EOF Then

bill = "B101"

Else

Adodc1.Recordset.MoveLast
```

```
id1 = Adodc1.Recordset("Bill_id")

id = Mid(id1, 2, 4) + 1

bill = "B" & id

End If

u_id.SetFocus

Adodc1.Recordset.AddNew

Exit Sub

errmsg:

MsgBox Err.Description

End Sub

Private Sub rate_click()

Dim id1 As String

Adodc3.Refresh

Adodc3.Recordset.MoveFirst

id1 = InputBox("Enter the code")

If id1 = "1" Then

rate = Adodc3.Recordset("1")

ElseIf id1 = "2" Then

rate = Adodc3.Recordset("2")

Else

errmsg:

MsgBox "not a valid code"

End If
```

```
Exit Sub

rate.Enabled = False

End Sub

Private Sub dat_Click()

dat = Date

End Sub

Private Sub exit_Click()

Unload Me

admin_page.Show

End Sub

Private Sub form_load()

bill = ""

u_id = ""

dat = ""

quantity = ""

fat = ""

rate = ""

acne = ""

tot = ""

bill.Enabled = False

u_id.Enabled = False

dat.Enabled = False

Selec.Enabled = False
```

```
quantity.Enabled = False
fat.Enabled = False
rate.Enabled = False
acno.Enabled = False
tot.Enabled = False
End Sub
Private Sub selec_dropdown()
Selec.clear
While Not Adodc1.Recordset.EOF
Selec.AddItem Adodc1.Recordset.Fields(1)
Adodc1.Recordset.MoveNext
Wend
Selec.Text = ""
Adodc1.Recordset.Closec
End Sub
Private Sub tot_Click()
tot = Val(rate) * Val(quantity)
End Sub
```

5.8 Rate Form

```
Private Sub cmdsav_Click()
On Error GoTo errmsg
If (cow.Text = "" Or buffalo.Text = "") Then
MsgBox "Please fill the empty fields"
```

```
Else

Adodc1.Recordset.Fields("1") = cow.Text

Adodc1.Recordset.Fields("2") = buffalo.Text

MsgBox ("Rates Updated")

End If

Exit Sub

errmsg:

MsgBox Err.Description

End Sub

Private Sub Command1_Click()

cow.Enabled = True

buffalo.Enabled = True

cow = ""

buffalo = ""

On Error GoTo errmsg

If Adodc1.Recordset.EOF Then

Adodc1.Recordset.AddNew

Else

Adodc1.Recordset.delete

Adodc1.Recordset.AddNew

MsgBox ("Rates deleted ")

End If

Exit Sub
```

```
errmsg:
MsgBox Err.Description
End Sub
Private Sub Command2_Click()
Unload Me
admin_page.Show
WindowState = 2
End Sub
Private Sub form_load()
cow.Enabled = False
buffalo.Enabled = False
cow = Adodc1.Recordset("1")
buffalo = Adodc1.Recordset("2")
End Sub
```

5.9 Payment Details Form

```
 Private Sub cmdfind_Click()
On Error GoTo errmsg
u_id.Enabled = False
bill.Enabled = False
acno.Enabled = False
amt.Enabled = False
Dim id1 As String
Adodc1.Refresh
```

```
id1 = InputBox("Enter the Supplier ID")

Adodc1.Recordset.Find "u_id='" & id1 & "'"

If Adodc1.Recordset.EOF Then

MsgBox "Supplier id not found"

Else

u_id = Adodc1.Recordset("u_id")

bill = Adodc1.Recordset("Bill_id")

acne = Adodc1.Recordset("acne")

amt = Adodc1.Recordset("tot")

End If

Exit Sub

MsgBox Err.Description

End Sub

Private Sub exit_Click()

Unload Me

admin_page.Show

WindowState = 2

End Sub

Private Sub form_load()

u_id.Enabled = False

acno.Enabled = False

amt.Enabled = False

End Sub
```

6 IMPLEMENTATION AND SCREENSHOTS

6.1 IMPLEMENTATION

Modules:

1 Admin
2 Supplier

Module Functionalities

1 Admin:

- Login
- Register Supplier
- Modify Supplier
- Delete Supplier
- Billing
- Modify Milk rate
- Payment Details

2 Supplier:

- Login
- View Profile

6.2 SCREENSHOTS

Home Page

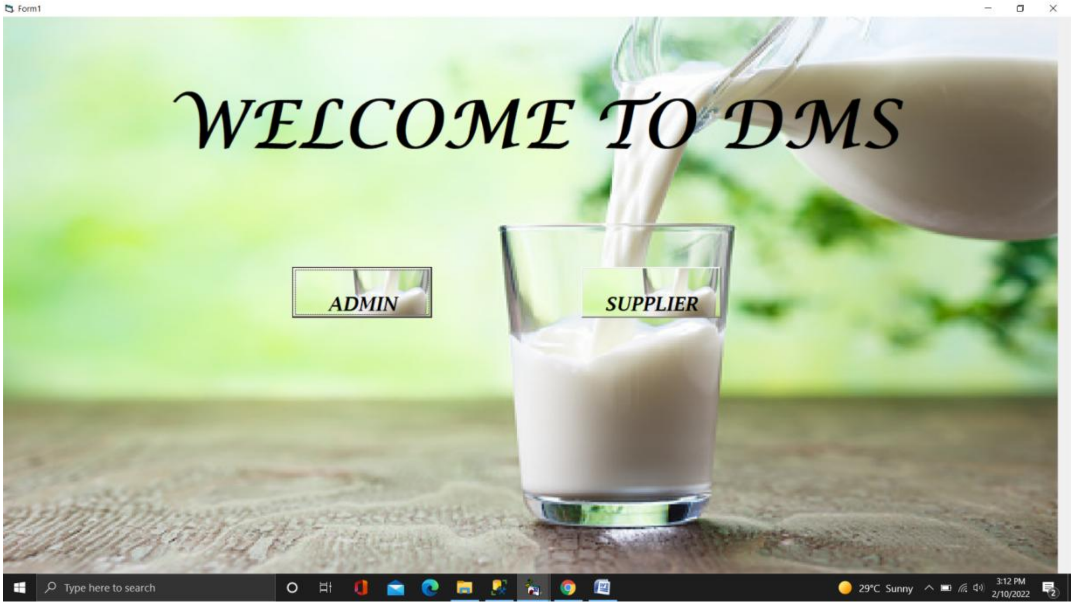


Fig.1. Welcome page

Here the choice is given to the user to select whether the user is admin or supplier.

Supplier Login Page

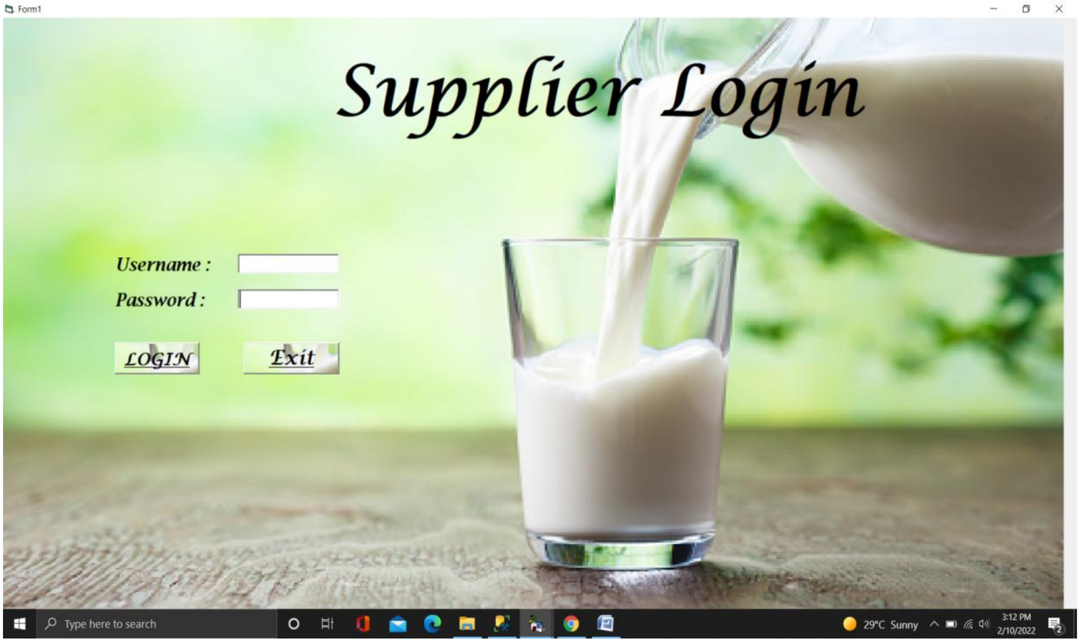
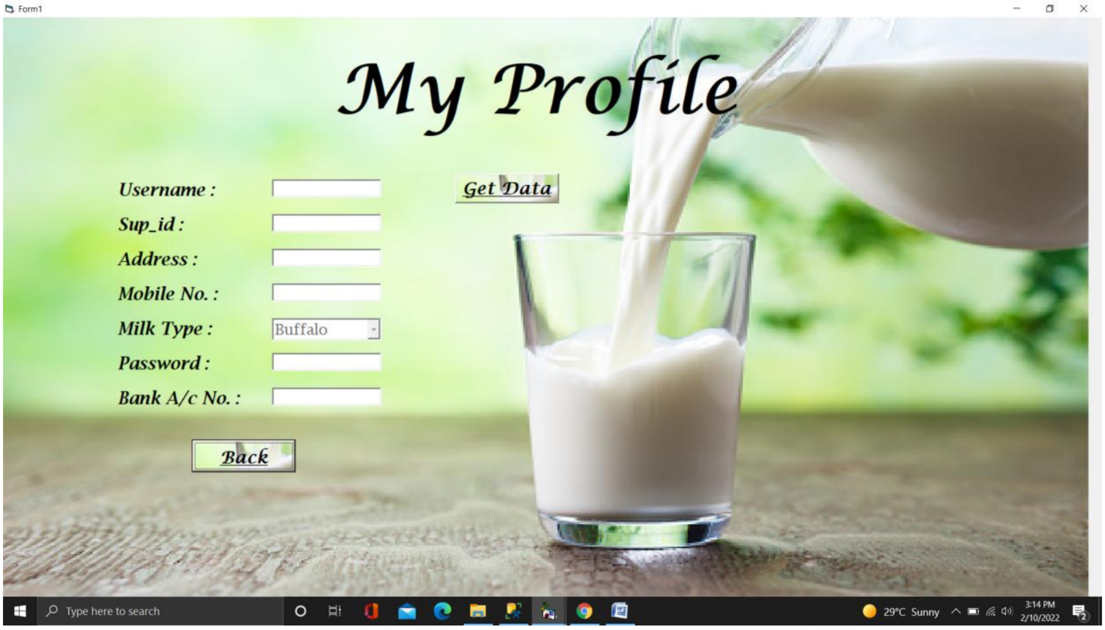


Fig.2.Supplier Login

Here the supplier has to login to access the application by filling in their credentials.

Profile

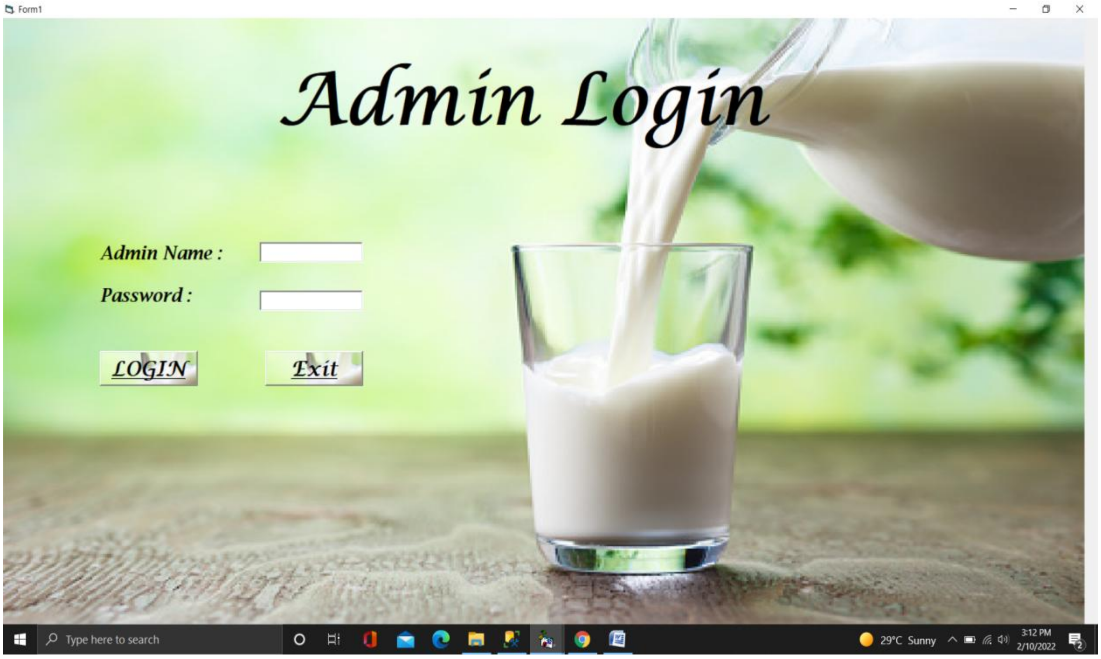


Username :
Sup_id :
Address :
Mobile No. :
Milk Type : Buffalo
Password :
Bank A/c No. :

Fig.3.Profile

Here the details of the supplier are retrieved and displayed.

Admin Login Page



Admin Name :
Password :

Fig.4.Admin Login

Here the admin has to login to access the application by filling in the credentials.

Admin Page

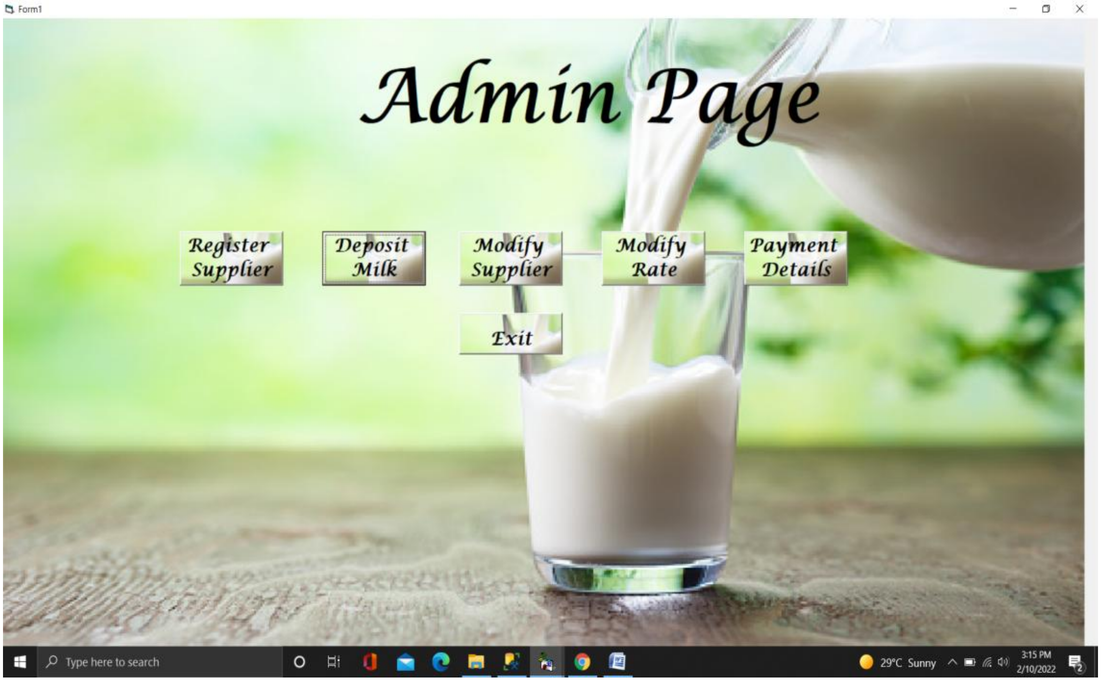


Fig.5.Admin Page

Here the admin has access to select/choose what operation should be done.

Registration Form

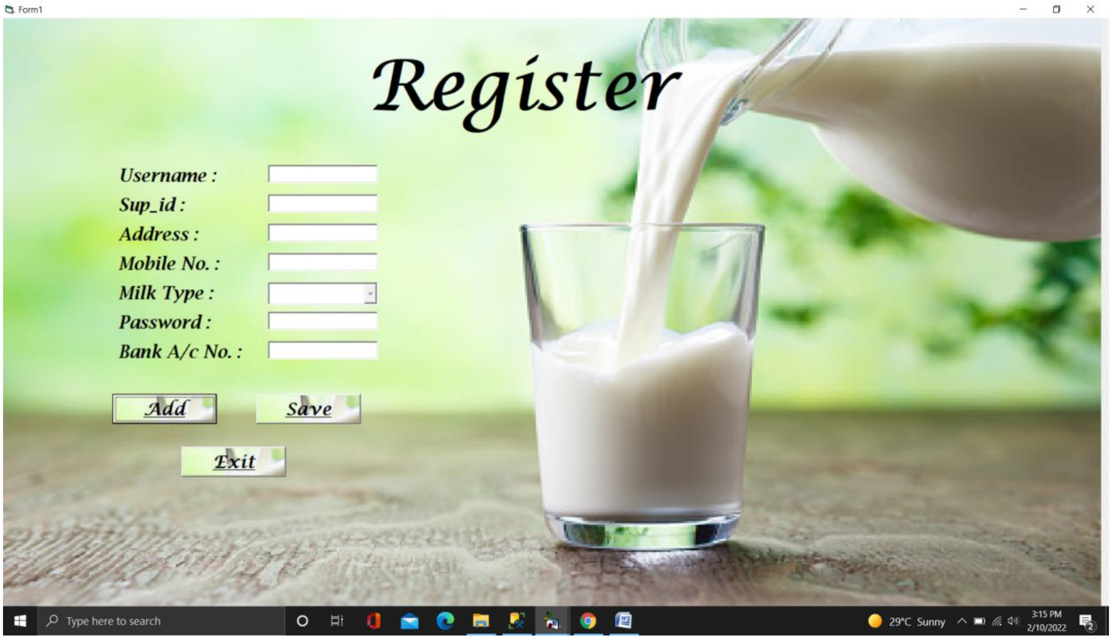


Fig.6.Registration

Here admin can add new supplier details.

Billing Form

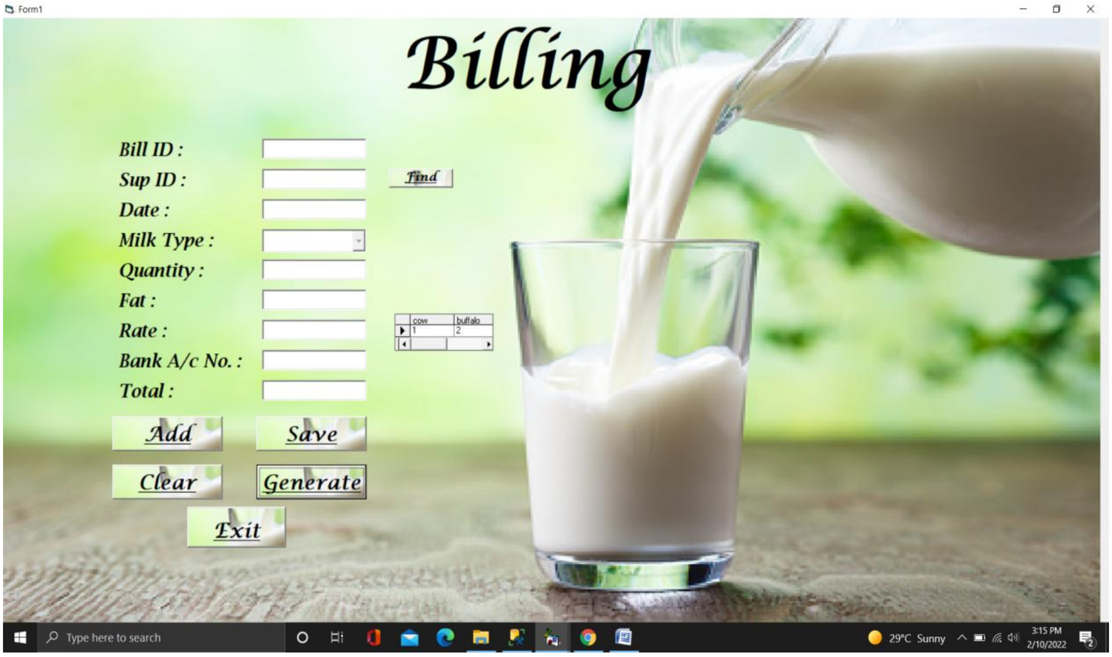


Fig.7.Billing

Here admin can generate bill to the supplier.

Modify Details

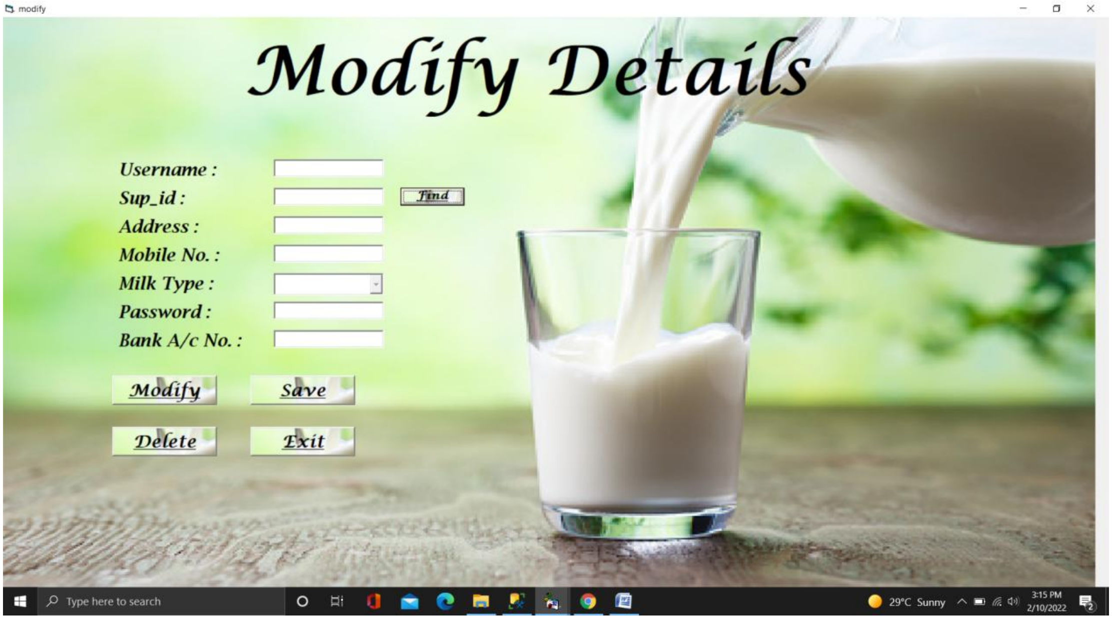


Fig.8.Modify Supplier details

Here admin can alter or modify the details of existing suppliers.

Modify milk rate

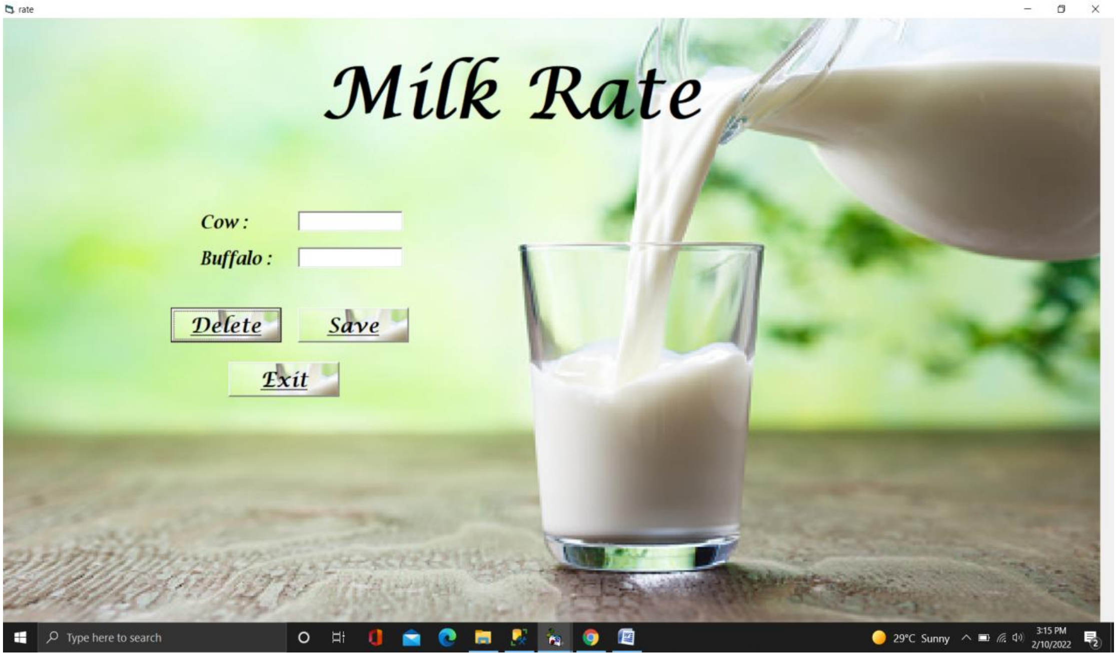


Fig.9.Modify milk rate

Here admin can delete or alter the rates of milk.

Payment Details

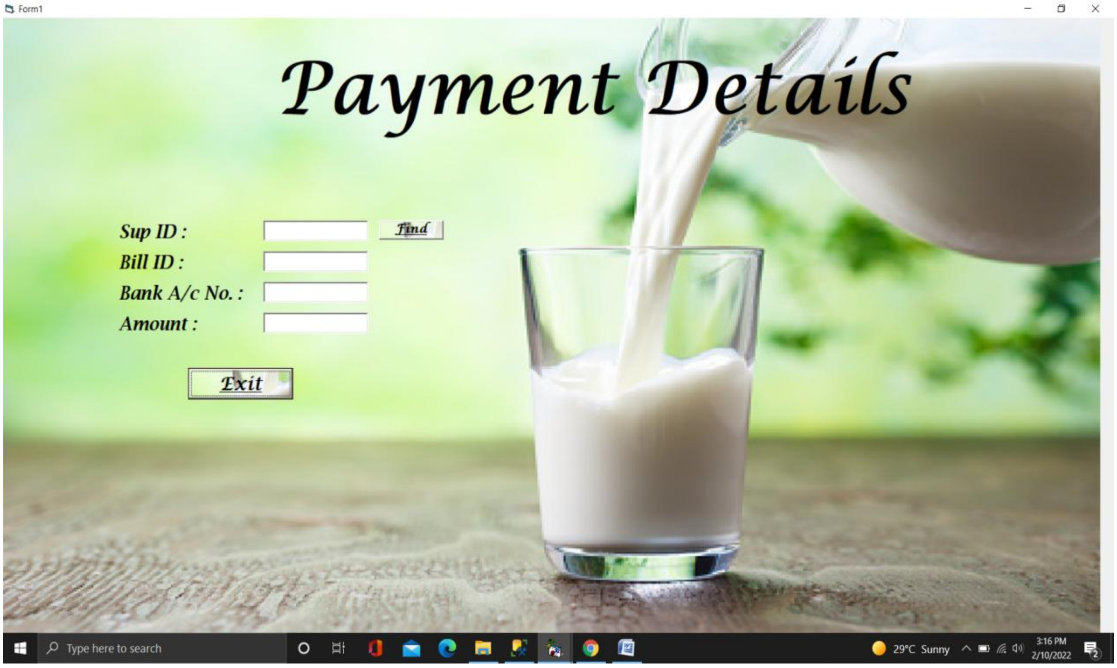


Fig.10.Payment Details

Here admin can view details of payment done in past.

7 TESTING

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Some prefer saying Software testing as a White Box and Black Box Testing. In simple terms, Software Testing means the verification of Application Under Test (AUT). This tutorial introduces testing software to the audience and justifies its importance.

7.1 UNIT TESTING

Unit Testing is a software testing technique by means of which individual units of software i.e. group of computer program modules, usage procedures and operating procedures are tested to determine whether they are suitable for use or not. It is a testing method using which every independent module are tested to determine if there are any issue by the developer himself. It is correlated with functional correctness of the independent modules.

Unit testing is defined as a type of software testing where individual components of software are tested.

Unit testing of software product is carried out during the development of an application. An individual component may be either an individual function or a procedure. Unit testing is typically performed by the developer.

In SDLC or V Model, Unit testing is first level of testing done before integration testing. Unit testing is such type of testing technique that is usually performed by the developers. Although due to reluctance of developers to tests, quality assurance engineers also do unit testing.

7.2 ITEGRATION TESTING:

Integration testing is performed to test individual components to check how they function together. In other words, it is performed to test the modules which are working fine individually and do not show bugs when integrated. It is the most common functional testing type and performed as automated testing.

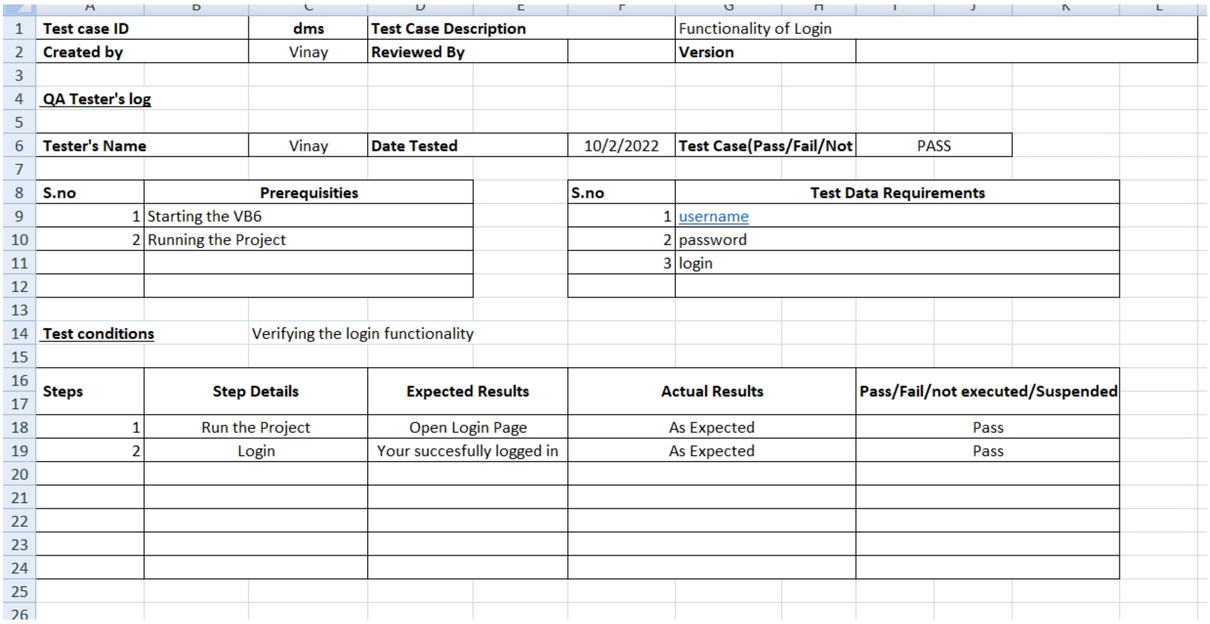
Generally, developers build different modules of the system/software simultaneously and don’t focus on others. They perform extensive black and white box functional verification, commonly known as unit tests, on the individual modules. Integration tests cause data and operational commands to flow between a module which means that they have to act as parts of a whole system rather than individual components. This typically uncovers issues with UI operations, data formats, operation timing, API calls, and database access and user interface operation.

7.3 SYSTEM TESTING:

System testing is testing conducted on a complete, integrated system to evaluate its compliance with the specified requirements.

After the completion of the integration testing, the product is passed for system testing. System testing is undertaken by independent testers who haven’t played a role in developing the program. This testing is performed in an environment that closely mirrors production. System Testing is very important because it verifies that the application meets the technical, functional, and business requirements that were set by the stakeholder.

7.4 Test case 1 :



Test case ID	**dms**	**Test Case Description**		Functionality of Login	
Created by	Vinay	**Reviewed By**		**Version**	

QA Tester's log

Tester's Name	Vinay	**Date Tested**	10/2/2022	**Test Case(Pass/Fail/Not**	PASS

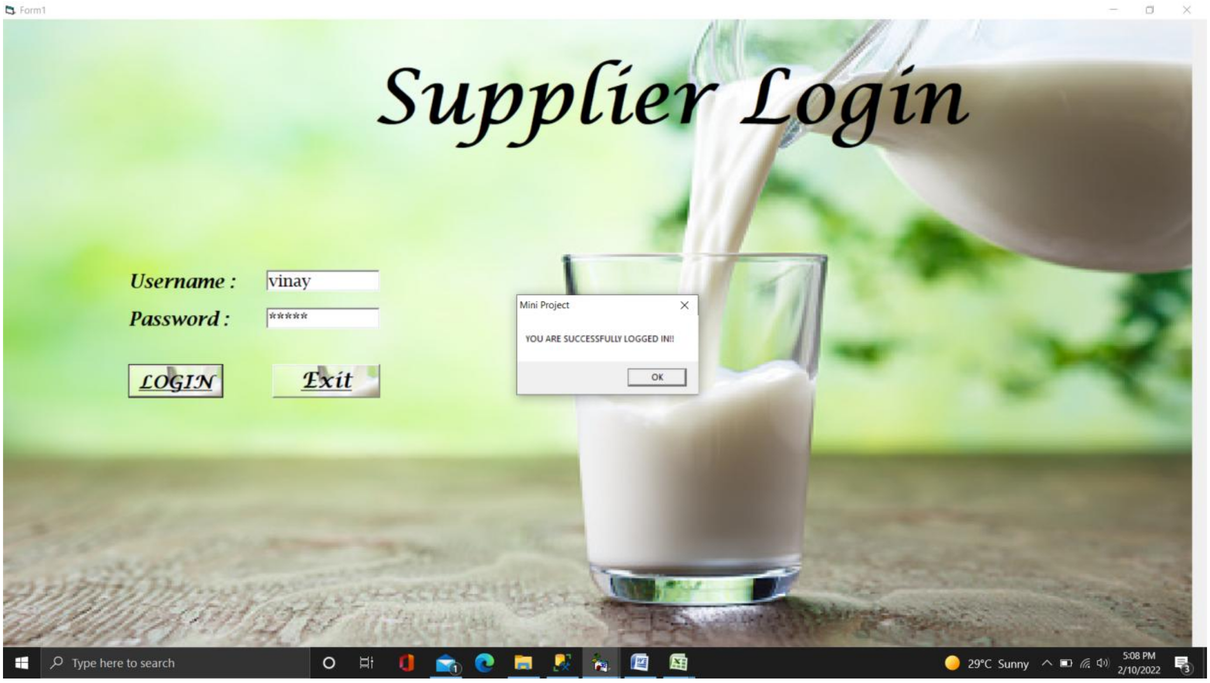
S.no	**Prerequisities**
1	Starting the VB6
2	Running the Project

S.no	**Test Data Requirements**
1	username
2	password
3	login

Test conditions Verifying the login functionality

Steps	**Step Details**	**Expected Results**	**Actual Results**	**Pass/Fail/not executed/Suspended**
1	Run the Project	Open Login Page	As Expected	Pass
2	Login	Your succesfully logged in	As Expected	Pass

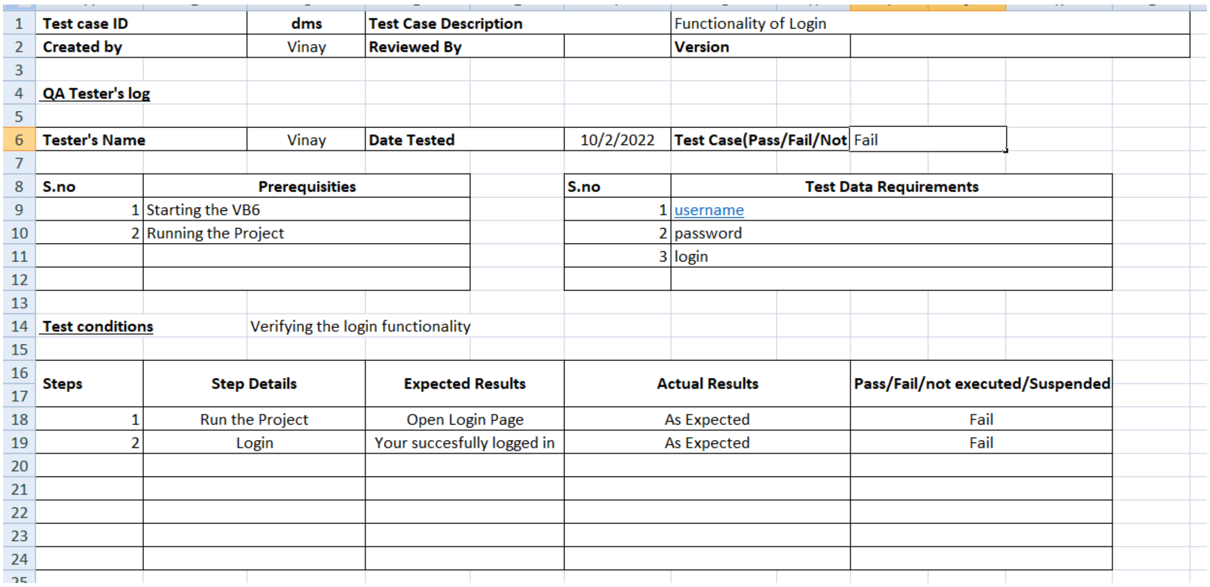
Here we test the login functionality, on running the project login page appears as expected and on entering valid credentials supplier is logged in.



Username : vinay

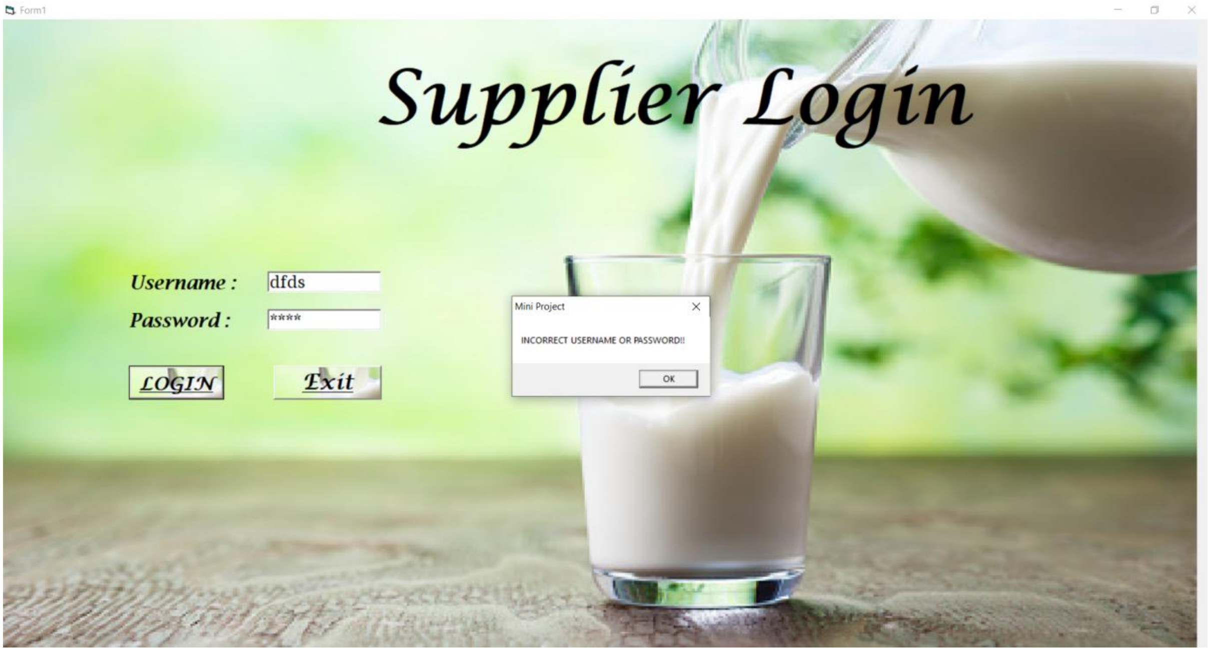
Password : *****

YOU ARE SUCCESSFULLY LOGGED IN!!



Test case ID	dms	Test Case Description		Functionality of Login	
Created by	Vinay	Reviewed By		Version	
QA Tester's log					
Tester's Name	Vinay	Date Tested	10/2/2022	Test Case(Pass/Fail/Not	Fail
S.no	Prerequisities		S.no	Test Data Requirements	
1	Starting the VB6		1	username	
2	Running the Project		2	password	
			3	login	
Test conditions	Verifying the login functionality				
Steps	Step Details	Expected Results	Actual Results		Pass/Fail/not executed/Suspended
1	Run the Project	Open Login Page	As Expected		Fail
2	Login	Your succesfully logged in	As Expected		Fail

In scenario 2 of the login functionality, on running the project login page appears and on entering username and on giving wrong password it still accepts and checks for the record making it a fail case. After alteration, i.e, after giving the validation the problem is solved.

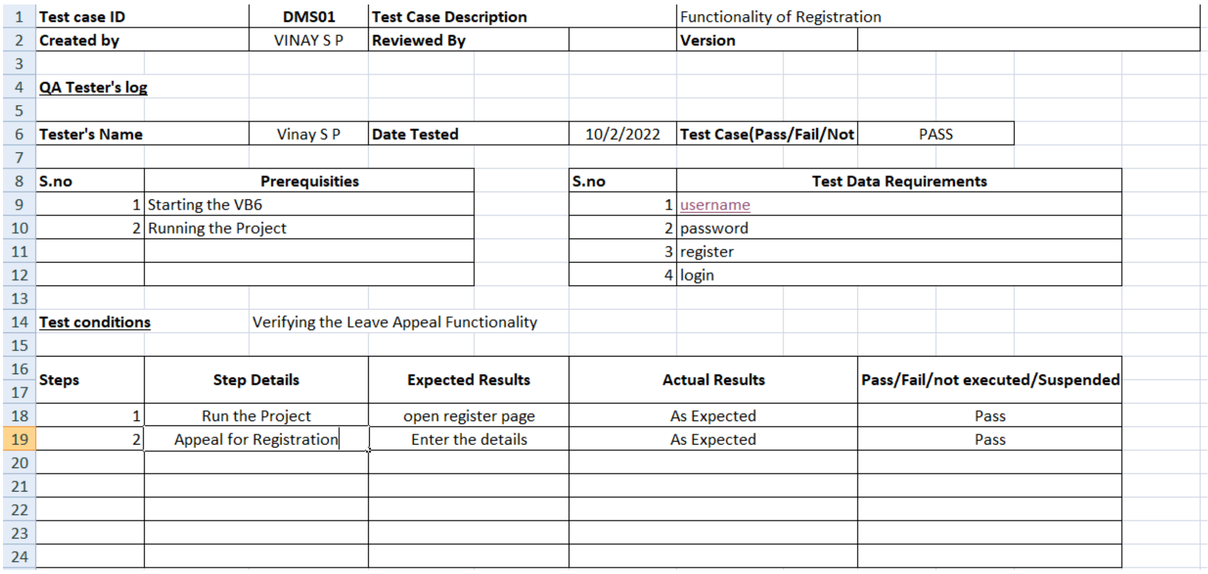


Username : dfds

Password : ****

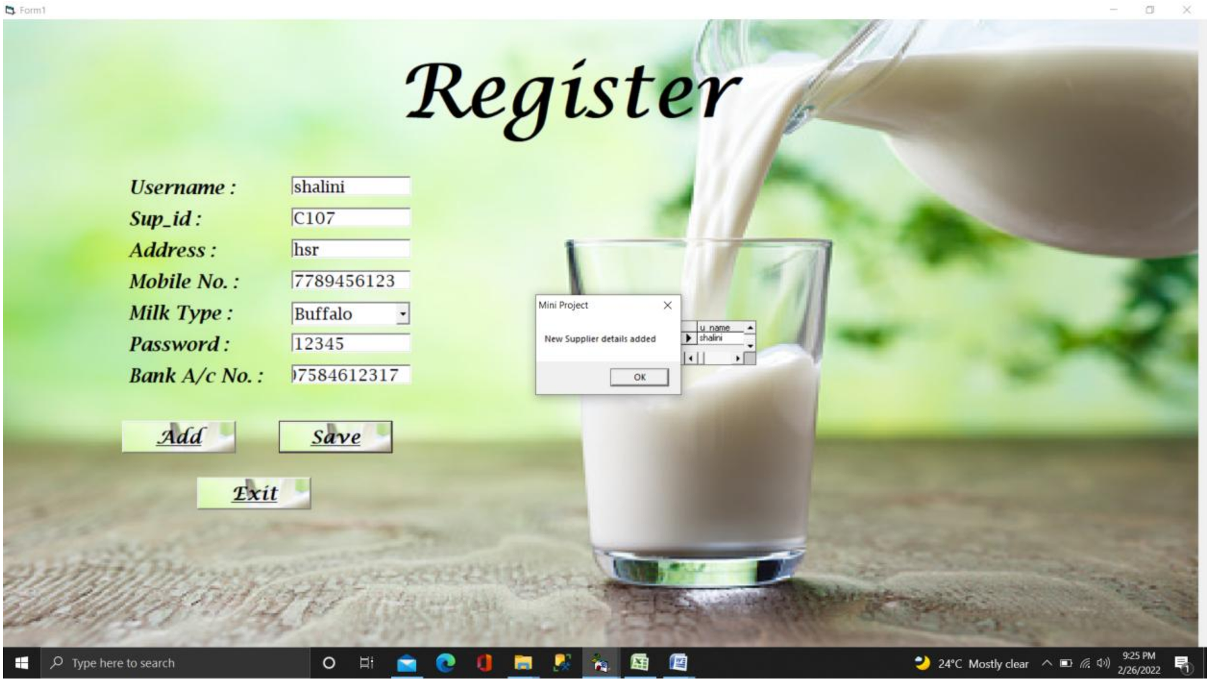
INCORRECT USERNAME OR PASSWORD!!

7.5 Test case 2 :



1	Test case ID	DMS01	Test Case Description		Functionality of Registration	
2	Created by	VINAY S P	Reviewed By		Version	
3						
4	QA Tester's log					
5						
6	Tester's Name	Vinay S P	Date Tested	10/2/2022	Test Case(Pass/Fail/Not	PASS
7						
8	S.no	Prerequisities		S.no	Test Data Requirements	
9	1	Starting the VB6		1	username	
10	2	Running the Project		2	password	
11				3	register	
12				4	login	
13						
14	Test conditions	Verifying the Leave Appeal Functionality				
15						
16 17	Steps	Step Details	Expected Results	Actual Results	Pass/Fail/not executed/Suspended	
18	1	Run the Project	open register page	As Expected	Pass	
19	2	Appeal for Registration	Enter the details	As Expected	Pass	
20						
21						
22						
23						
24						

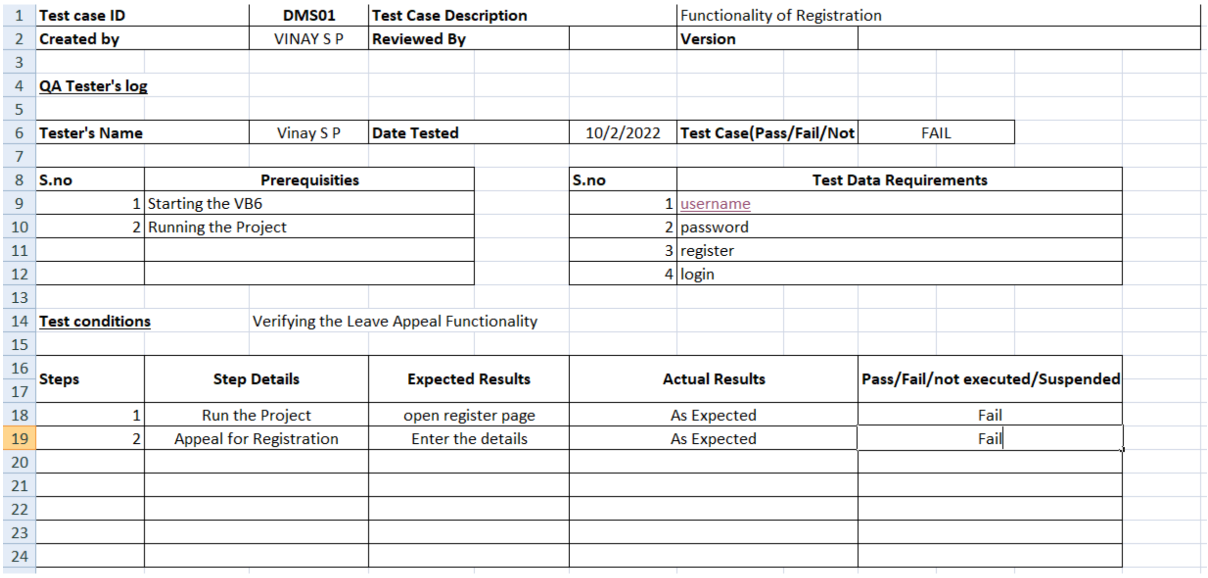
Here we test the register functionality, on running the project registration page appears as expected and on entering valid credentials successful message box is displayed.



Username : shalini
Sup_id : C107
Address : hsr
Mobile No. : 7789456123
Milk Type : Buffalo
Password : 12345
Bank A/c No. :)7584612317

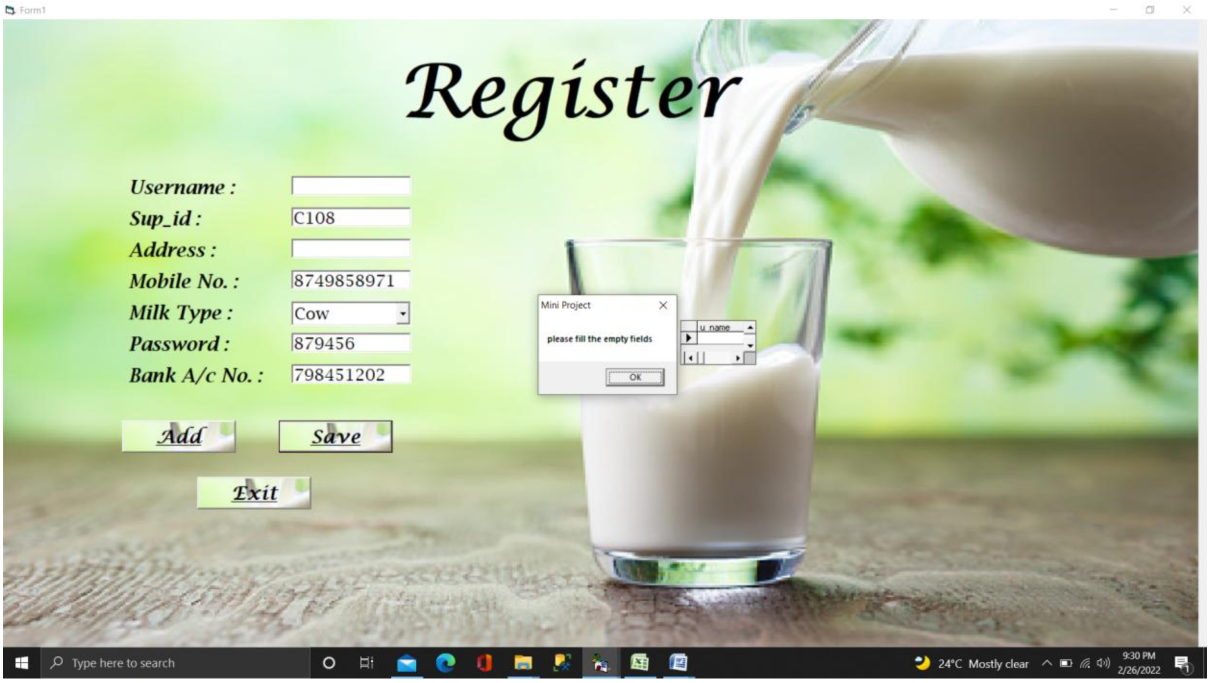
Mini Project
New Supplier details added
OK

shalini



1	Test case ID		DMS01	Test Case Description			Functionality of Registration	
2	Created by		VINAY S P	Reviewed By			Version	
3								
4	QA Tester's log							
5								
6	Tester's Name		Vinay S P	Date Tested		10/2/2022	Test Case(Pass/Fail/Not	FAIL
7								
8	S.no	Prerequisities				S.no	Test Data Requirements	
9	1	Starting the VB6				1	username	
10	2	Running the Project				2	password	
11						3	register	
12						4	login	
13								
14	Test conditions		Verifying the Leave Appeal Functionality					
15								
16	Steps	Step Details		Expected Results		Actual Results		Pass/Fail/not executed/Suspended
17								
18	1	Run the Project		open register page		As Expected		Fail
19	2	Appeal for Registration		Enter the details		As Expected		Fail
20								
21								
22								
23								
24								

Here we test the register functionality, on running the project registration page appears as expected and on entering invalid credentials error message box is displayed.



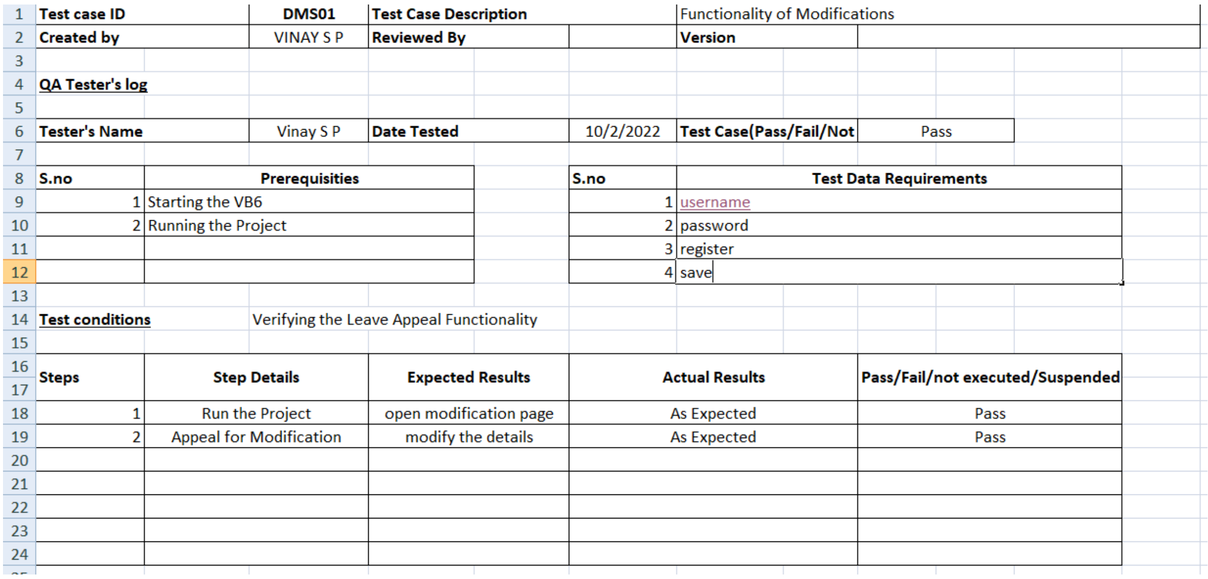
Username :
Sup_id : C108
Address :
Mobile No. : 8749858971
Milk Type : Cow
Password : 879456
Bank A/c No. : 798451202

Mini Project

please fill the empty fields

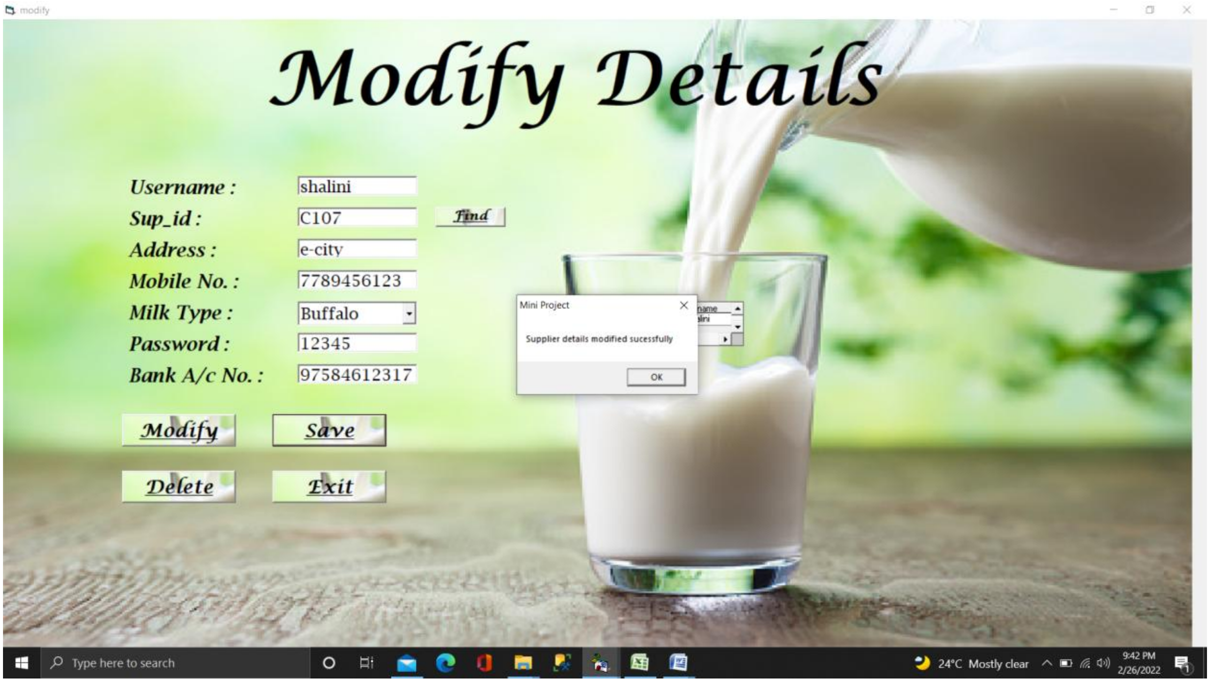
OK

7.6 Test case 3 :



1	Test case ID		DMS01	Test Case Description		Functionality of Modifications	
2	Created by		VINAY S P	Reviewed By		Version	
3							
4	QA Tester's log						
5							
6	Tester's Name		Vinay S P	Date Tested	10/2/2022	Test Case(Pass/Fail/Not	Pass
7							
8	S.no	Prerequisities			S.no	Test Data Requirements	
9	1	Starting the VB6			1	username	
10	2	Running the Project			2	password	
11					3	register	
12					4	save	
13							
14	Test conditions		Verifying the Leave Appeal Functionality				
15							
16	Steps	Step Details		Expected Results	Actual Results		Pass/Fail/not executed/Suspended
17							
18	1	Run the Project		open modification page	As Expected		Pass
19	2	Appeal for Modification		modify the details	As Expected		Pass
20							
21							
22							
23							
24							

Here we test the modification functionality, on running the project modification page appears as expected and on entering valid credentials successful message box is displayed.



Username : shalini

Sup_id : C107

Address : e-city

Mobile No. : 7789456123

Milk Type : Buffalo

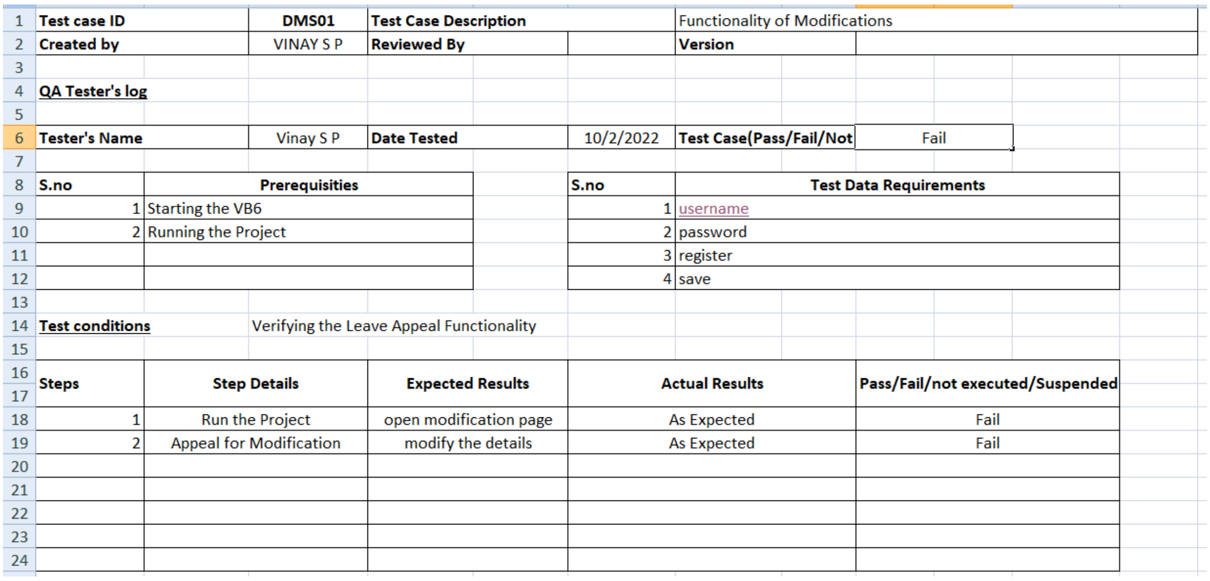
Password : 12345

Bank A/c No. : 97584612317

Mini Project

Supplier details modified sucessfully

OK



Test case ID	DMS01	Test Case Description		Functionality of Modifications	
Created by	VINAY S P	Reviewed By		Version	

QA Tester's log

Tester's Name	Vinay S P	Date Tested	10/2/2022	Test Case(Pass/Fail/Not	Fail

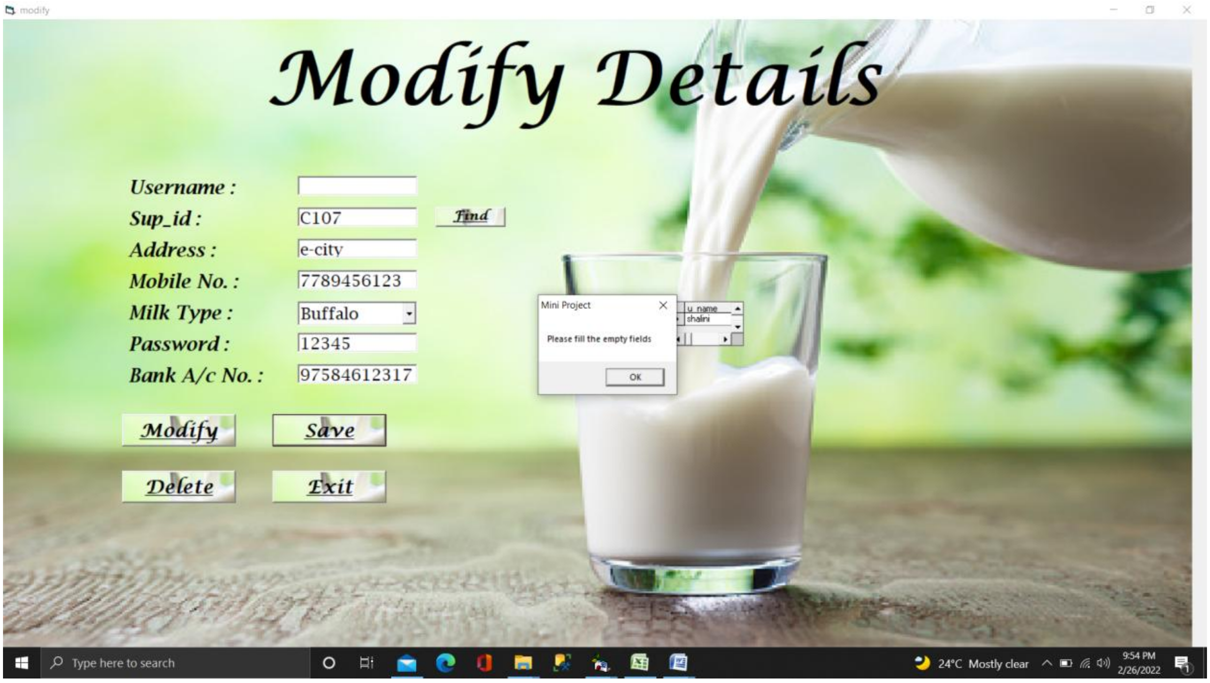
S.no	Prerequisities
1	Starting the VB6
2	Running the Project

S.no	Test Data Requirements
1	username
2	password
3	register
4	save

Test conditions Verifying the Leave Appeal Functionality

Steps	Step Details	Expected Results	Actual Results	Pass/Fail/not executed/Suspended
1	Run the Project	open modification page	As Expected	Fail
2	Appeal for Modification	modify the details	As Expected	Fail

Here we test the modification functionality, on running the project modification page appears as expected and on entering invalid credentials error message box is displayed.



Username :
Sup_id : C107
Address : e-city
Mobile No. : 7789456123
Milk Type : Buffalo
Password : 12345
Bank A/c No. : 97584612317

Mini Project

Please fill the empty fields

OK

u_name
shalini

8 CONCLUSION

While developing the system a conscious effort has been made to develop the application, making the use of available resources, tools and techniques that would generate the proper application.

While making this application, the main intension was to make this application simpler, user friendly UI. The application being flexible and is running successfully. The system adequately meets the user needs and their requirements are satisfied.

As in case of any system development processes where there are a number of shortcomings, there have been some shortcomings in the development of the application and this project will undergo further modifications in the future.

Therefore, our application will definitely prove to be the successful stepping stone in replacing the outdated manual method of maintaining/managing records. This application when implemented it will remove all the manual errors caused and will computerize the management system.

9 FUTURE ENHANCEMENT

The scope of the project includes that what all future enhancement can be done in this application to make it more feasible to use.

In future the application can be converted into a web application with more features were the users will be allowed to register themselves without admin guidance. They would be able to make online payments.

10 BIBLIOGRAPHY

- Visual Basic 6 Black Book by Steven Holzner.
- Visual Basic 6: The Complete Reference by Noel Jerke.
- Learning SQL by Alan Beaulier.
- SQL Cookbook by Anthony Molinaro.
- https://www.geeksforgeeks.org
- https://www.w3schools.com
- https://support.microsoft.com
- https://www.scribd.com