

Documentation Book

Table of Contents

[**1.** **INTRODUCTION** 2](#_Toc30794085)

[**1.1** **BACKGROUND** 2](#_Toc30794086)

[**1.2** **MISSION** 2](#_Toc30794087)

[**1.3** **VISSION** 2](#_Toc30794088)

[**1.4** **OBJECTIVES** 3](#_Toc30794089)

[**1.5** **SWOT ANALYZE OF THE COMPANY** 3](#_Toc30794090)

[1.5.1 Strength 3](#_Toc30794091)

[1.5.2 Weakness 3](#_Toc30794092)

[1.5.3 Opportunities 3](#_Toc30794093)

[1.5.4 Threats 3](#_Toc30794094)

[**2.** **Existing System Problem** 4](#_Toc30794095)

[**3.** **Proposed SYSTEM (New System)** 4](#_Toc30794096)

[**4.** **Feasibility Study** 5](#_Toc30794097)

[4.1 Operational feasibility 5](#_Toc30794098)

[4.2 Technical feasibility 6](#_Toc30794099)

[4.3 Economic feasibility 6](#_Toc30794100)

[4.4 schedule feasibility 7](#_Toc30794101)

[**5.** **SYSTEM DATA DICTIONARY (Table Structures)** 7](#_Toc30794102)

[**6.** **SCREENSHOOTS** 10](#_Toc30794103)

[**7.** **RECOMANDATION** 14](#_Toc30794104)

[8. **CONCLUTION** 14](#_Toc30794105)

1. **INTRODUCTION**
   1. **BACKGROUND**

First, I am liibaan the owner **LIBAN super market** I want to talk the background of my company.

It was founded 2012 on April 23 at TARABUNKA and found it by myself and my hope was to be business man.

The company Is growing Very well and has expended his services to the society living in Mogadishu and nearby places

**LIBAN Super Market** now has many branches in Mogadishu especially Tabeelaha, Tarabunka.

* 1. **MISSION**
* To make a delivery to customers: -
* Delivery takes place that the customers get their product easily

It makes us easily to get more customers to industry and it’s a father service that we will have according our competitors

* To build a call center: -
* The company will make special number that the customers will communicate to check that they

* 1. **VISSION**
* To import the product specially
* To the get more employees
* To get a system to manage the company
* To build more department
  1. **OBJECTIVES**
* Establish the different kinds of inventory kept at LIBAN SUPER MARKET
* Examine different inventory control techniques practiced at the firm.
* Establish the costs incurred by the firm through inventory management.
  1. **SWOT ANALYZE OF THE COMPANY**
     1. **Strength**

- there are no Much competitors in the area

- the company has Many customers

- the company has many branches in Mogadishu

- the company has relationship for foreign suppliers

- employees of the company has experience

* + 1. **Weakness**

- The super market had no land

- It has a problem of payment for employees

- the company has no delivery

- the houses of employee are far from the company so that they might get delay

* + 1. **Opportunities**

- if the company promotes manual system to computerized system it may increase its customer

- if the company adds delivery it may also increase its customer

- if the company get a system it may also increase its customer

* + 1. **Threats**

- The land of the company is governmental land

- if the company is not paid much costs to its employee it may leave the experienced employees

- government policies to the business may impact the company’s activities

1. **EXISTING SYSTEM PROBLEM**

Working hand in their long-term career to work in the commercial interest of the FA and the loss is so long it was so used to it, but later decreased roundup continued use because of the harm such an ID.

Some of his problems is.

1. Not easily get money from your company, who is missing you and how much you are having difficulty getting this information to the business.
2. It is also difficult to get credit for your debt.
3. Not keeping a good record of how much you have and how much you sell it is another challenge.
4. You cannot get your daily and gross income from your business.
5. It is easy to steal money from a business easily without feeling that act.
6. Loss of vital information or omission of data.
7. Employees working in the business are often overpriced.
8. It is very difficult to manage a place of business.
9. **PROPOSED SYSTEM (NEW SYSTEM)**

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work of the financial activities.

**AdVantages of the Proposed System**

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features

1. Security of data : when using manual system, it does not have much security example if you are using books for registering data may be stolen or can affect other risk so that using the system ensures that data will not access second part
2. Ensure data accuracy’s.

Computerized systems are highly accurate than manual system because of a person may calculate wrong calculation or mistake caused by typing errors

1. Minimize manual data entry.

The system reduces time taken manual system and facilitates the company’s activities also system convinces customers and motivates the staff to work hard

1. Minimum time needed for the various processing.

When using manual system, it takes much time for processing data so this System saves time and enhances job facility

1. ­Better service
2. User friendliness and interactive.

**Features of New system**

1. Registration and Management of Customers, Suppliers, Employees
2. Income and Cost records
3. Expenses
4. General and Specific Reports
5. Debts And liablities Records
6. **FEASIBILITY STUDY**
   1. Operational feasibility

We believe that this system will succeed 100% because of mostly the company’s employee is educated and can understand the system and operations that is related it also this is a high quality which manager of the company will satisfaction.

This System Will the manager to get data easily which might take much time that may lead Customer dis satisfaction

Some company staff were already use small system that they were using to store data However, Training the employee to this system is easy and will not take much time.

The system needs a person which if the system uses more people administrator that can notice us if the system needs maintenance

The use of this systems saves more time that the writing and searching data and customers will see how to get their order easily

* 1. Technical feasibility

The Resources is ready which is necessary to publish the system on time and also has the quality needed.

The Tools needed to set up the system is ready for the company, and this system will cover the needs of the company now and until the company will work.

The system is unique to this company will be the competition between other companies and has new features which is differ from existing system

If the system shall use the right and appropriate way, it will take a great role for the growth of company’s net worth and can manage the future business transaction of the company

This system will solve some problem that are facing the company

The system has been developed using VB.net And Sql Server the project is technically feasible for development.

* 1. Economic feasibility

Our developers do not need much expense and also the training of the staff is same as but always there is a cost

In order to set up this system we need hardware devices like (desktop or laptop) and even printer to print invoices and reports to the company

Its independent system and it will not need other system because its desktop application, may be to need utility applications but company prepares every cost which the setting up this system is needed

Training the staff will not take much time also we are ready to solve and make clear everything related this system

|  |  |  |
| --- | --- | --- |
| Cost Name | Cost | Description |
| Internet | $15 | Internet usage |
| Pre-paid | $10 | Calling usage |
| Training | $100 | Training usage |
| Total | $125 |  |

* 1. schedule feasibility

We have a 2-month period to develop a new information system with quality needed and the customer satisfied

If error comes while testing the system, we already planned if something like that happened to make correction.

There are many challenges facing us but we will concentrate how to develop a new system with high quality and customer and company satisfaction

|  |  |  |
| --- | --- | --- |
| Name | Date | Responsible |
| Developing the system | 2 Week |  |
| Testing | 1 Week |  |
| Training the staff | 1 Week |  |

1. **SYSTEM DATA DICTIONARY (TABLE STRUCTURES)**

Table Name: Customers

Primary Key: CID

| FieldName | Data Type | Field Length | Constrain | Description |
| --- | --- | --- | --- | --- |
| CID | int |  | Primary key | Customer id |
| Cname | Varchar | 30 | Not Null | Customer name |
| Ctell | Int |  | Unique | Customer Tell |
| Cadress | Varchar | 50 | Not Null | Home address |
| Cdate | Data | ------ | Not Null | Register Date |
| Creference | varchar | 50 | Not Null | Customer Reference |
| Creference-No | Int |  | Unique | Reference tell |

Table Name: Employees

Primary Key:Eid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Length | Constrain | Description |
| EId | int |  | Primary key | Employee id |
| EmpName | Varchar | 30 | Not Null | Employee name |
| Emptell | Int |  | Unique | Employee Tell |
| EmpReference | Varchar | 50 | Not Null | Emp Reference |
| EmpReference-No | int |  | Unique | Referemce Tell |
| EmpSalary | Money |  | Not Null | Employee Salary |
| Rdate | Date | -- | Not Null | Registred date |

Table Name: Revenue

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Length | Constrain | Description |
| R-name | Varchar | 30 | ----- | Revenue name |
| R-date | date | -- | Not Null | date |
| R-dsecription | varchar | 140 | Unique | Revenue Descr |
| Cr | int |  | Not Null | Income revenue |

Table Name: Expenses

primary key: Expid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Length | Constrain | Description |
| ExpID | int |  | Primary key | Expense id |
| Expname | Varchar | 20 | Not Null | Expense name |
| Expdate | date | --- | Unique | Date of expense |
| Dr | int |  | Not Null | Expense Amount |
| Descr | Varchar | 150 | Not Null | description |

Table Name: Cash

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Length | Constrain | Description |
| Cash Name | Varchar | 30 | -- | Cash name |
| Cash Descr | Varchar | 50 | Not Null | Cash Description |
| cash-date | date | --- | Not Null | Date of Cash |
| Dr | int |  | Not Null | cash out |
| Cr | Int |  | Not Null | Cash in |

Table Name: Account payables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Length | Constrain | Description |
| Supp-Id | Int | 30 | foreign Key | Supplier Name |
| Supp-Name | Varchar | 40 | Not Null | Cash Description |
| Dr | int |  | Not Null | cash out |
| Cr | Int |  | Not Null | Cash in |

Table Name: Suppliers

Primary Key:Supp\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Length | Constrain | Description |
| Supp-Id | Int |  | Primary key | Supplier Id |
| Supp-Name | Varchar | 40 | Not Null | Supplier Name |
| Supp-Tell | int |  | Unique | Supplier Tell |
| Supp-address | Varchar | 20 | Not Null | Supplier address |

Table Name: Account Receivables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Length | Constrain | Description |
| Cus-Id | Int | 30 | Foreign key | Customer ID |
| Cus-Name | Varchar | 40 | Not Null | Customer name |
| Dr | int |  |  | cash out |
| Cr | Int |  |  | Cash in |

**Customers**

**Supplier**

View customer

|  |
| --- |
|  |
| **New**  **supplier** |

|  |
| --- |
| **0** |
| **Customer**  **Registration** |

|  |  |
| --- | --- |
| **S1** | **Customers** |

|  |  |
| --- | --- |
| **S** | **Suppliers** |

|  |
| --- |
|  |
| **Liability**  **New** |

|  |
| --- |
|  |
| **Receiving**  **dept.** |

|  |
| --- |
|  |
| **Taking**  **Dept.** |

|  |  |
| --- | --- |
| **S** | **Liabilities** |

|  |  |
| --- | --- |
| **S** | **Dept.** |

|  |  |
| --- | --- |
| **S** | **Cash** |

|  |
| --- |
|  |
| **Pay**  **Loan** |

|  |
| --- |
|  |
| **Register**  **Employee** |

|  |  |
| --- | --- |
| **S** | **Employee** |

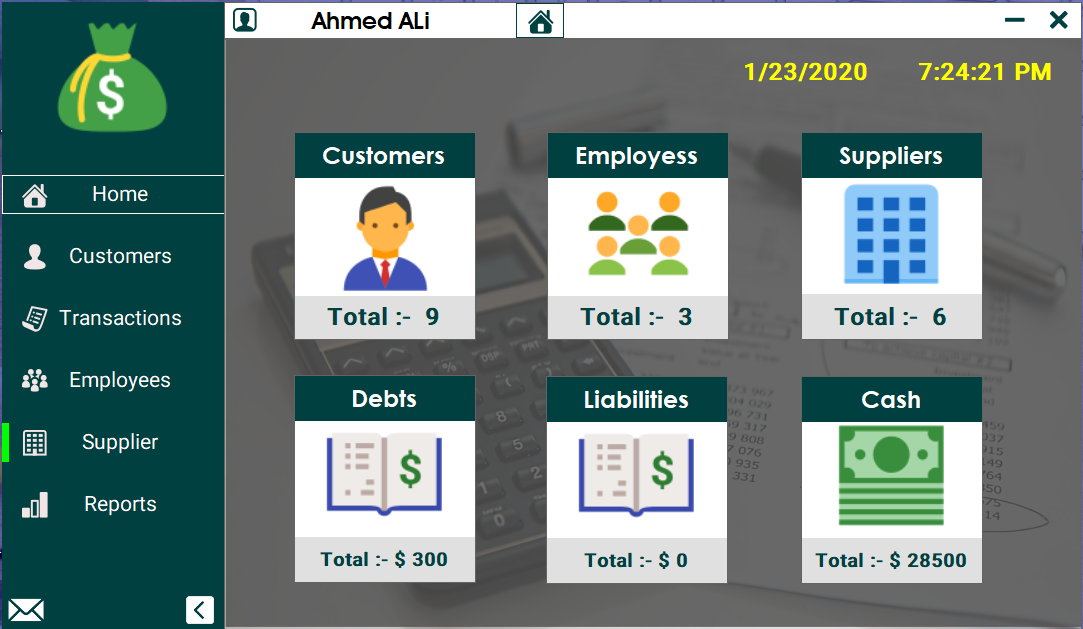
|  |
| --- |
|  |
| **Pay**  **Salary** |

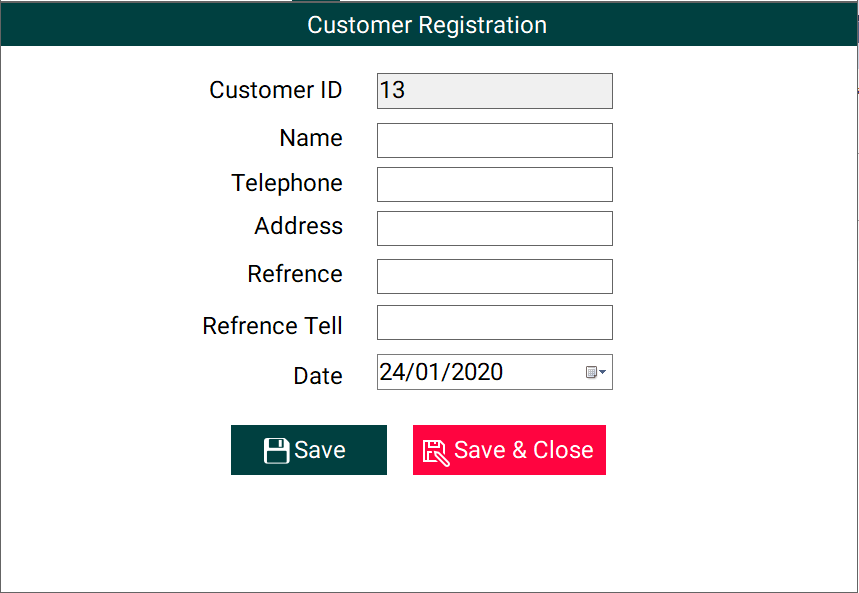
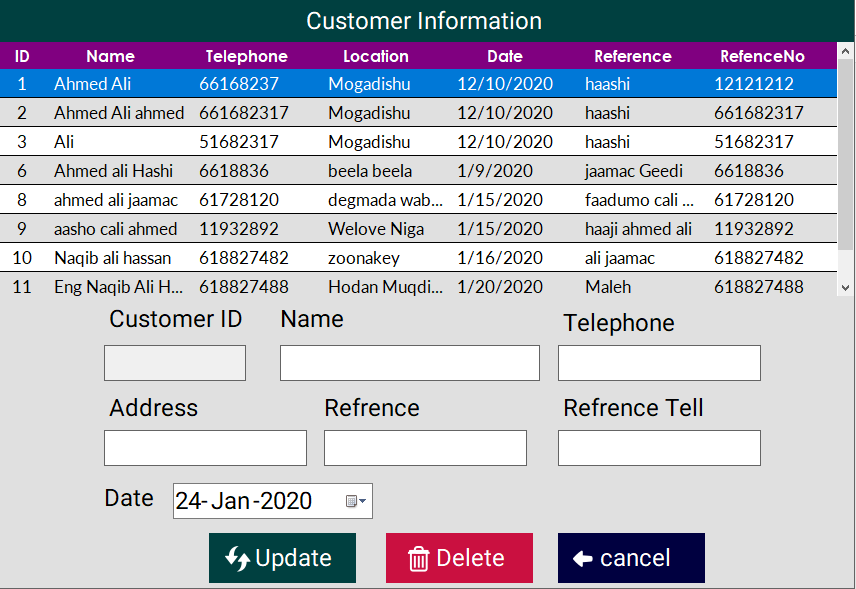
|  |
| --- |
|  |
| **Cash Out** |

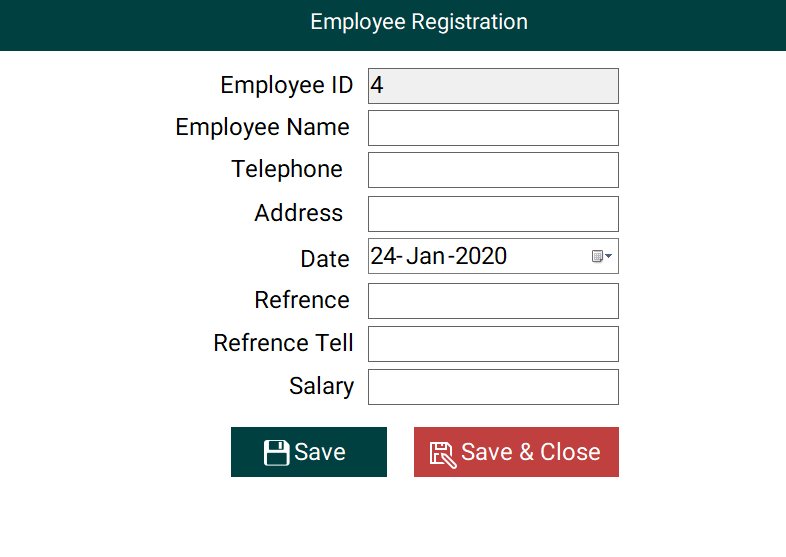
|  |
| --- |
|  |
| **Cash In** |

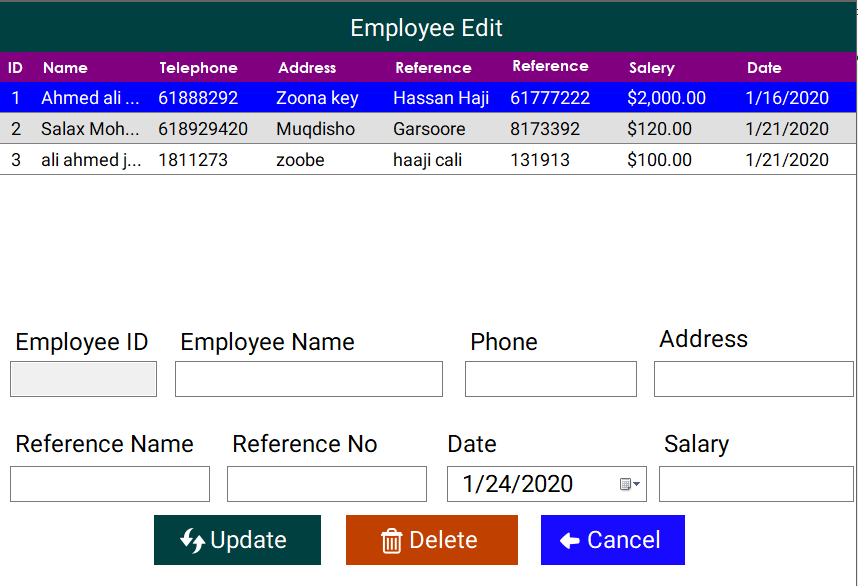
**Employee**

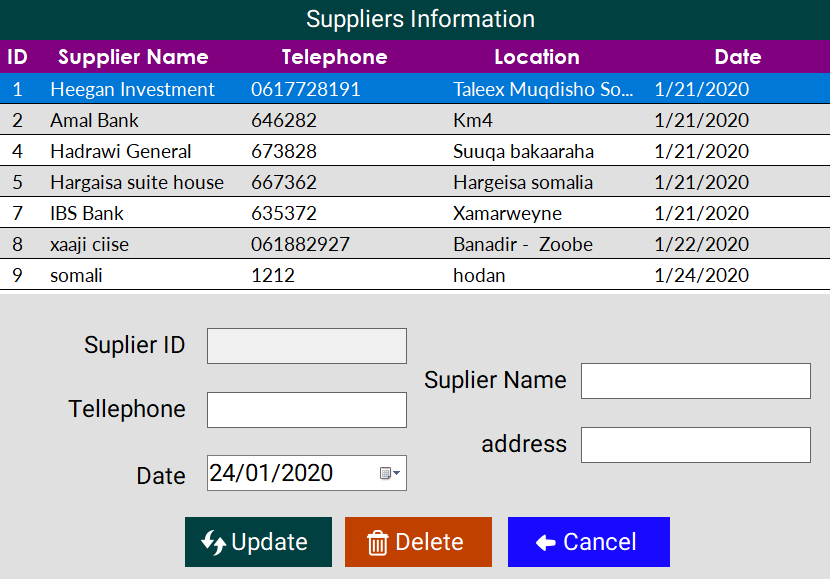
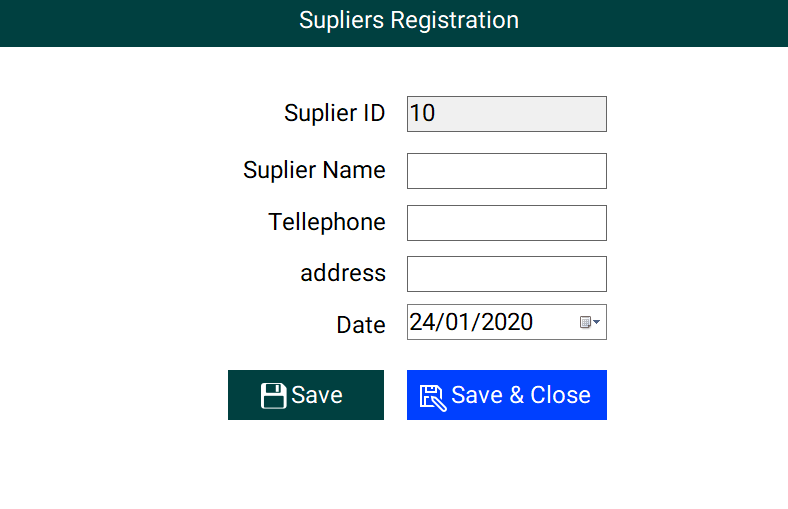
1. **SCREENSHOOTS**

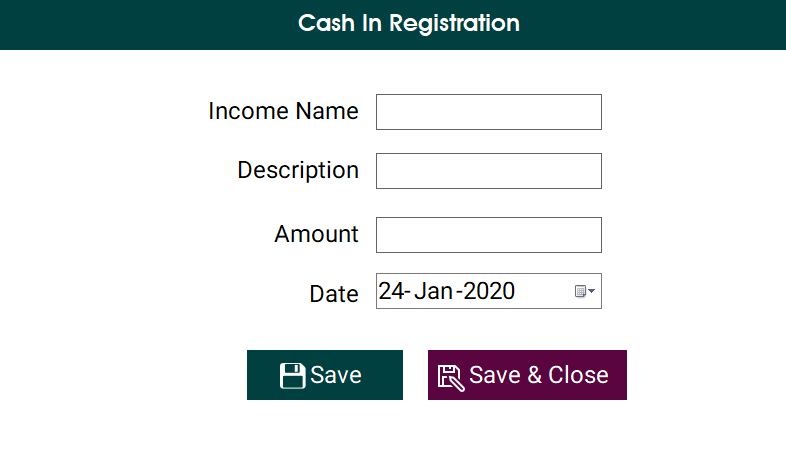
Main Dashboard Admin Form

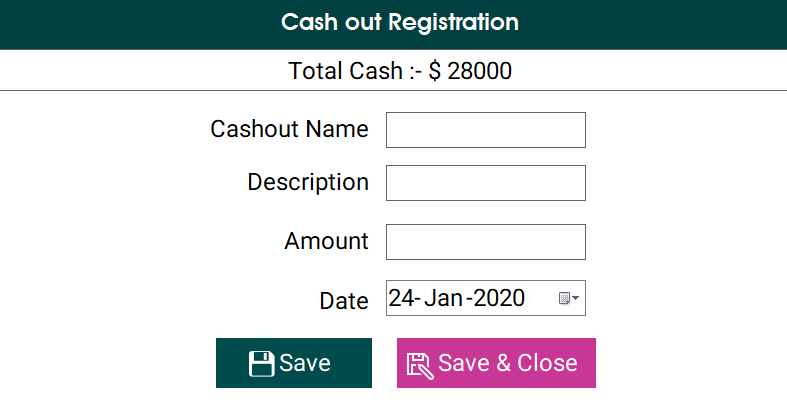
Customer Registration Edit Customer Information

Employee Registration Edit Employee information

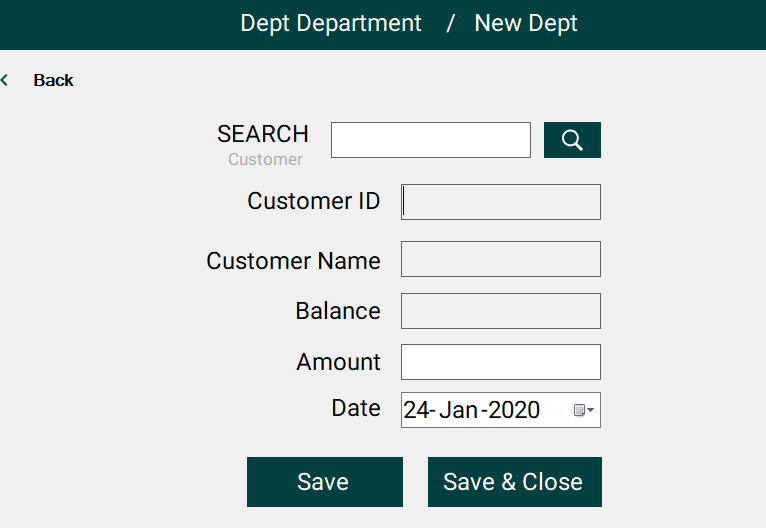
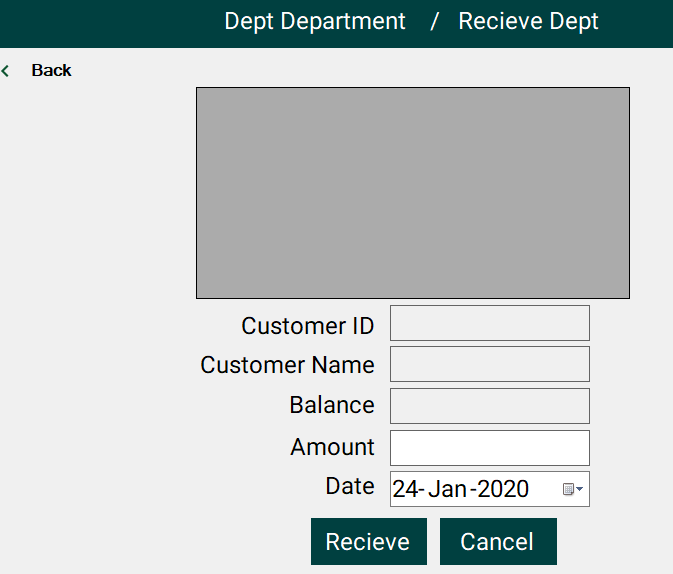


Supplier Registration Edit Supplier Information

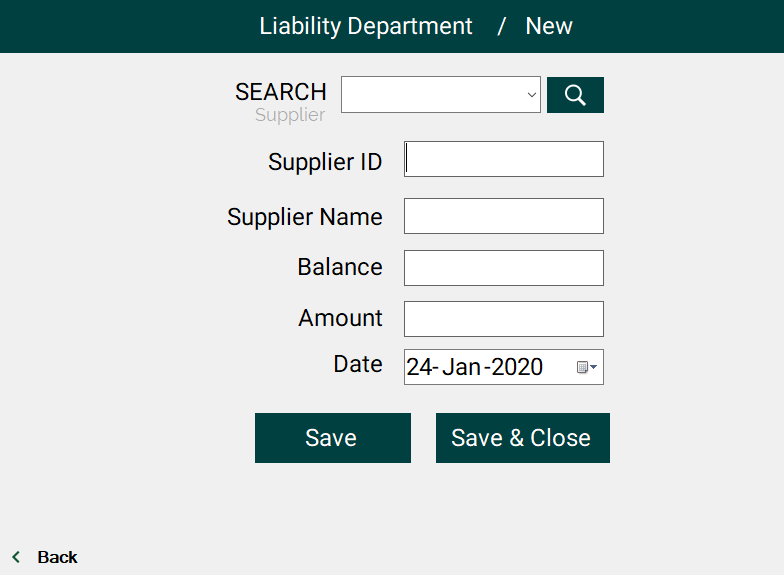
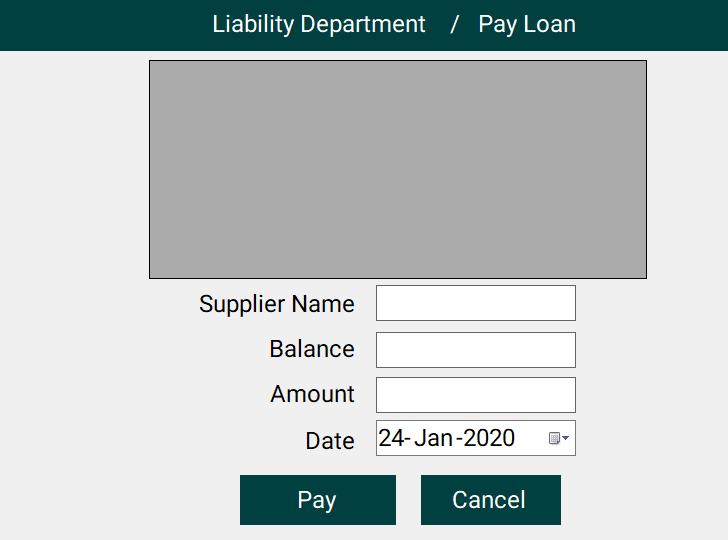
Income Form Cash Out or Costs

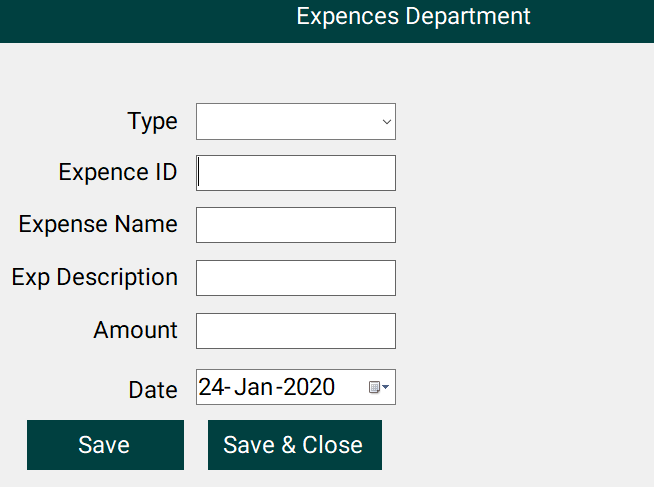


Debts Receive Debts



Liability Pay Liability



 Expense Edit Expense Info

**Coding**

* **Main form admin**

Public Class Form1

Private Sub Button1\_MouseHover(sender As Object, e As EventArgs) Handles Button6.MouseEnter, Button5.MouseEnter, Button4.MouseEnter, Button3.MouseEnter, Button2.MouseEnter, Button1.MouseEnter, btnsetting.MouseEnter

Dim a As Button = sender

Panel5.Top = a.Top

Panel5.BringToFront()

End Sub

Private Sub PictureBox1\_Click(sender As Object, e As EventArgs) Handles PictureBox1.Click

If MessageBox.Show("Do You want To close", "Financial Management System", MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation) = MsgBoxResult.Yes Then

login.Close()

End If

End Sub

Private Sub Form1\_Load() Handles MyBase.Load

Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

End Sub

Private Sub PictureBox2\_Click(sender As Object, e As EventArgs) Handles PictureBox2.Click

Me.WindowState = FormWindowState.Minimized

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

If Panel4.Controls.Contains(Menusvb.instance) Then

Panel4.Controls.Remove(Menusvb.instance)

Main.instance.dock = DockStyle.Fill

Else

Panel4.Controls.Add(Menusvb.instance)

Menusvb.\_instance.BringToFront()

Menusvb.\_instance.Location = New Point(0, 175)

Menusvb.\_instance.Height = 100

Menusvb.\_instance.btn1.Text = "Register"

Menusvb.\_instance.btn1.Tag = "C1"

Menusvb.\_instance.btn2.Text = "Edit"

Menusvb.\_instance.btn2.Tag = "C2"

End If

End Sub

Private Sub PictureBox4\_Click(sender As Object, e As EventArgs) Handles PictureBox4.Click

If Panel2.Width = 222 Then

Panel2.Width = 56

PictureBox6.Visible = False

Else

Panel2.Width = 222

PictureBox6.Visible = True

End If

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

If Panel4.Controls.Contains(Menusvb.instance) Then

Panel4.Controls.Remove(Menusvb.instance)

Main.instance.dock = DockStyle.Fill

Else

Panel4.Controls.Add(Menusvb.instance)

' Main.instance.dock = DockStyle.None

Menusvb.\_instance.BringToFront()

Menusvb.\_instance.Location = New Point(0, 175)

Menusvb.\_instance.btn1.Text = "Cash In"

Menusvb.\_instance.btn1.Tag = "cash1"

Menusvb.\_instance.btn2.Text = "Cash Out"

Menusvb.\_instance.btn2.Tag = "cash2"

Menusvb.\_instance.btn3.Text = "A.Recievable"

Menusvb.\_instance.btn3.Tag = "ar"

Menusvb.\_instance.btn4.Text = "A.Payble"

Menusvb.\_instance.btn4.Tag = "ap"

Menusvb.\_instance.btn5.Text = "Expence"

Menusvb.\_instance.btn5.Tag = "ex"

Menusvb.\_instance.Height = 250

End If

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

If Panel4.Controls.Contains(Menusvb.instance) Then

Panel4.Controls.Remove(Menusvb.instance)

Main.instance.dock = DockStyle.Fill

Else

Panel4.Controls.Add(Menusvb.instance)

'Main.instance.dock = DockStyle.None

Menusvb.\_instance.BringToFront()

Menusvb.\_instance.Location = New Point(0, 300)

Menusvb.\_instance.Height = 100

Menusvb.\_instance.btn1.Text = "Register"

Menusvb.\_instance.btn1.Tag = "em1"

Menusvb.\_instance.btn2.Text = "Edit"

Menusvb.\_instance.btn2.Tag = "em2"

End If

End Sub

Private Sub Button5\_Click(sender As Object, e As EventArgs) Handles Button5.Click

If Panel4.Controls.Contains(Menusvb.instance) Then

Panel4.Controls.Remove(Menusvb.instance)

Main.instance.dock = DockStyle.Fill

Else

Panel4.Controls.Add(Menusvb.instance)

' Main.instance.dock = DockStyle.None

Menusvb.\_instance.BringToFront()

Menusvb.\_instance.Location = New Point(0, 360)

Menusvb.\_instance.Height = 100

Menusvb.\_instance.btn1.Text = "Register"

Menusvb.\_instance.btn1.Tag = "sup1"

Menusvb.\_instance.btn2.Text = "Edit"

Menusvb.\_instance.btn2.Tag = "sup2"

End If

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

Panel4.Controls.Remove(Menusvb.instance)

Main.instance.bringtofront()

Main.instance.Main\_Load()

End Sub

Private Sub PictureBox6\_Click(sender As Object, e As EventArgs) Handles PictureBox6.Click

feedback.Show()

End Sub

Private Sub Button6\_Click(sender As Object, e As EventArgs) Handles Button6.Click

Panel4.Controls.Add(Reports.instance)

Reports.instance.dock = DockStyle.Fill

Reports.instance.bringtoFront()

End Sub

Private Sub Button7\_Click(sender As Object, e As EventArgs)

Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

Panel4.Controls.Remove(Menusvb.instance)

Main.instance.bringtofront()

Main.instance.Main\_Load()

End Sub

Private Sub btnsetting\_Click(sender As Object, e As EventArgs) Handles btnsetting.Click

Panel4.Controls.Add(Setting.instance)

Setting.instance.dock = DockStyle.Fill

Setting.instance.bringtofront()

End Sub

Private Sub PictureBox3\_Click(sender As Object, e As EventArgs) Handles PictureBox3.Click

End Sub

End Class

* **Customer registration**

Public Class customer

Public Shared \_instance As New customer

Public Shared ReadOnly Property instance

Get

If \_instance Is Nothing Or \_instance.IsDisposed Then

\_instance = New customer

End If

Return \_instance

End Get

End Property

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Try

If chck() Then

cmd.Connection = con

cmd.CommandText = "exec RegisterCustomer @cid,@cname,@ctell,@cadd,@dt,@ref,@rtel,'insert'"

cmd.Parameters.AddWithValue("@cid", cid.Text)

cmd.Parameters.AddWithValue("@cname", cname.Text)

cmd.Parameters.AddWithValue("@ctell", ctell.Text)

cmd.Parameters.AddWithValue("@cadd", cadd.Text)

cmd.Parameters.AddWithValue("@dt", Rdate.Value)

cmd.Parameters.AddWithValue("@ref", Rname.Text)

cmd.Parameters.AddWithValue("@rtel", Rtell.Text)

connect()

cmd.ExecuteNonQuery()

MsgBox("Record Saved")

clear()

getId()

Else

MessageBox.Show("Please Fill Blanks", "Financial Management System", MessageBoxButtons.OK, MessageBoxIcon.Information)

End If

Catch ex As Exception

MsgBox(Err.Description)

End Try

cmd.Parameters.Clear()

End Sub

Sub clear()

cid.Clear()

cname.Clear()

ctell.Clear()

Rname.Clear()

Rtell.Clear()

Rdate.Value = Today

cadd.Clear()

End Sub

Private Sub customer\_Load(sender As Object, e As EventArgs) Handles Me.Load

getId()

End Sub

Sub getId()

Try

cmd.Connection = con

cmd.CommandText = "select case when max(cid) is null then 1 else max(cid)+1 end from customers"

connect()

cid.Text = cmd.ExecuteScalar

Catch ex As Exception

MsgBox(Err.Description)

End Try

End Sub

Function chck() As Boolean

Return cid.Text <> "" And cname.Text <> "" And ctell.Text <> "" And Rname.Text <> "" And Rtell.Text <> "" And Rdate.Text <> "" And cadd.Text <> ""

End Function

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

Try

If chck() Then

cmd.Connection = con

cmd.CommandText = "exec RegisterCustomer @cid,@cname,@ctell,@cadd,@dt,@ref,@rtel,'insert'"

cmd.Parameters.AddWithValue("@cid", cid.Text)

cmd.Parameters.AddWithValue("@cname", cname.Text)

cmd.Parameters.AddWithValue("@ctell", ctell.Text)

cmd.Parameters.AddWithValue("@cadd", cadd.Text)

cmd.Parameters.AddWithValue("@dt", Rdate.Value)

cmd.Parameters.AddWithValue("@ref", Rname.Text)

cmd.Parameters.AddWithValue("@rtel", Rtell.Text)

connect()

cmd.ExecuteNonQuery()

MsgBox("Customer Saved")

clear()

Form1.Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

Main.instance.bringtofront()

Else

MessageBox.Show("Please Fill Blanks", "Financial Management System", MessageBoxButtons.OK, MessageBoxIcon.Information)

End If

Catch ex As Exception

MsgBox(Err.Description)

End Try

cmd.Parameters.Clear()

End Sub

End Class

* **Edit Customer Information**

Public Class EditCustomer

Public Shared \_instance As New EditCustomer

Public Shared ReadOnly Property instance

Get

If \_instance Is Nothing Or \_instance.IsDisposed Then

\_instance = New EditCustomer

End If

Return \_instance

End Get

End Property

Public Sub EditCustomer\_Load() Handles MyBase.Load

loadData()

End Sub

Sub loadData()

ds.Clear()

da.SelectCommand = cmd

da.SelectCommand.Connection = con

da.SelectCommand.CommandText = "select \* from SelectCustomer"

da.Fill(ds, "T1")

DataGridView1.DataSource = ds.Tables("T1")

DataGridView1.Columns(0).Width = 50

DataGridView1.Columns(0).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

DataGridView1.Columns(1).Width = 145

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

sql("update")

End Sub

Sub clear()

cid.Clear()

cname.Clear()

ctell.Clear()

cref.Clear()

ctell.Clear()

Rdate.Value = Today

cadd.Clear()

End Sub

Function chck() As Boolean

Return cid.Text <> "" And cname.Text <> "" And ctell.Text <> "" And cref.Text <> "" And ctell.Text <> "" And Rdate.Text <> "" And cadd.Text <> ""

End Function

Sub sql(type As String)

Try

If chck() Then

cmd.Connection = con

cmd.CommandText = "exec RegisterCustomer @cid,@cname,@ctell,@cadd,@dt,@ref,@rtel,'" & type & "'"

cmd.Parameters.AddWithValue("@cid", cid.Text)

cmd.Parameters.AddWithValue("@cname", cname.Text)

cmd.Parameters.AddWithValue("@ctell", ctell.Text)

cmd.Parameters.AddWithValue("@cadd", cadd.Text)

cmd.Parameters.AddWithValue("@dt", Rdate.Value)

cmd.Parameters.AddWithValue("@ref", cref.Text)

cmd.Parameters.AddWithValue("@rtel", refno.Text)

connect()

If MsgBox(type & " Record of Customer :- " & cname.Text) Then

cmd.ExecuteNonQuery()

loadData()

clear()

End If

Else

MessageBox.Show("Please Fill Blanks", "Financial Management System", MessageBoxButtons.OK, MessageBoxIcon.Information)

End If

Catch ex As Exception

MsgBox(Err.Description)

End Try

cmd.Parameters.Clear()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

sql("delete")

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

Form1.Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

Main.instance.bringtofront()

Main.instance.main\_load()

End Sub

Private Sub DataGridView1\_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick

Try

cid.Text = DataGridView1.CurrentRow.Cells(0).Value

cname.Text = DataGridView1.CurrentRow.Cells(1).Value

ctell.Text = DataGridView1.CurrentRow.Cells(2).Value

cadd.Text = DataGridView1.CurrentRow.Cells(3).Value

Rdate.Value = DataGridView1.CurrentRow.Cells(4).Value

cref.Text = DataGridView1.CurrentRow.Cells(5).Value

refno.Text = DataGridView1.CurrentRow.Cells(6).Value

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub DataGridView1\_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick

End Sub

End Class

* Employee Registration

Public Class ex1

Public Shared \_instance As New ex1

Public Shared ReadOnly Property instance

Get

If \_instance Is Nothing Or \_instance.IsDisposed Then

\_instance = New ex1

End If

Return \_instance

End Get

End Property

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Try

If chck() Then

cmd.Connection = con

cmd.CommandText = "exec employeeinfo @eid,@ename,@etell,@eadd,@dt,@ref,@rtel,@sel,'insert'"

cmd.Parameters.AddWithValue("@eid", emid.Text)

cmd.Parameters.AddWithValue("@ename", ename.Text)

cmd.Parameters.AddWithValue("@etell", etell.Text)

cmd.Parameters.AddWithValue("@eadd", eadd.Text)

cmd.Parameters.AddWithValue("@dt", dt.Value)

cmd.Parameters.AddWithValue("@ref", ref.Text)

cmd.Parameters.AddWithValue("@rtel", reftell.Text)

cmd.Parameters.AddWithValue("@sel", esalery.Text)

connect()

cmd.ExecuteNonQuery()

MsgBox("Employee Saved !")

clear()

getId()

Else

MessageBox.Show("Please Fill Blanks", "Financial Management System", MessageBoxButtons.OK, MessageBoxIcon.Information)

End If

Catch ex As Exception

MsgBox(Err.Description)

End Try

cmd.Parameters.Clear()

End Sub

Private Function chck() As Boolean

Return emid.Text <> "" And ename.Text <> "" And etell.Text <> "" And eadd.Text <> "" And esalery.Text <> "" And dt.Text <> "" And ref.Text <> "" And reftell.Text <> ""

End Function

Sub getId()

Try

cmd.Connection = con

cmd.CommandText = "select case when max(eid) is null then 1 else max(eid)+1 end from employee"

connect()

emid.Text = cmd.ExecuteScalar

Catch ex As Exception

MsgBox(Err.Description)

End Try

End Sub

Private Sub clear()

emid.Clear()

ename.Clear()

etell.Clear()

ref.Clear()

etell.Clear()

dt.Value = Today

eadd.Clear()

esalery.Clear()

reftell.Clear()

End Sub

Private Sub ex1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

getId()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

Try

If chck() Then

cmd.Connection = con

cmd.CommandText = "exec employeeinfo @eid,@ename,@etell,@eadd,@dt,@ref,@rtel,@sel,'insert'"

cmd.Parameters.AddWithValue("@eid", emid.Text)

cmd.Parameters.AddWithValue("@ename", ename.Text)

cmd.Parameters.AddWithValue("@etell", etell.Text)

cmd.Parameters.AddWithValue("@eadd", eadd.Text)

cmd.Parameters.AddWithValue("@dt", dt.Value)

cmd.Parameters.AddWithValue("@ref", ref.Text)

cmd.Parameters.AddWithValue("@rtel", reftell.Text)

cmd.Parameters.AddWithValue("@sel", esalery.Text)

connect()

cmd.ExecuteNonQuery()

MsgBox("Employee Saved !")

clear()

Form1.Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

Main.instance.bringtofront()

Else

MessageBox.Show("Please Fill Blanks", "Financial Management System", MessageBoxButtons.OK, MessageBoxIcon.Information)

End If

Catch ex As Exception

MsgBox(Err.Description)

End Try

cmd.Parameters.Clear()

End Sub

End Class

* **Edit Employee information**

Public Class EmployeeEdit

Public Shared \_instance As New EmployeeEdit

Public Shared ReadOnly Property instance

Get

If \_instance Is Nothing Or \_instance.IsDisposed Then

\_instance = New EmployeeEdit

End If

Return \_instance

End Get

End Property

Public Sub EmployeeEdit\_Load() Handles MyBase.Load

loadata()

End Sub

Private Sub loadata()

ds.Clear()

da.SelectCommand = cmd

da.SelectCommand.Connection = con

da.SelectCommand.CommandText = "select \* from selectEmployee"

da.Fill(ds, "Emo")

DataGridView1.DataSource = ds.Tables("Emo")

DataGridView1.Columns("salery").DefaultCellStyle.Format = Format("1", "c")

DataGridView1.Columns(0).Width = 35

DataGridView1.Columns(0).DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

sql("update")

End Sub

Sub sql(type As String)

Try

If chck() Then

cmd.Connection = con

cmd.CommandText = "exec employeeinfo @eid,@ename,@etell,@eadd,@dt,@ref,@rtel,@sel,'" & type & "'"

cmd.Parameters.AddWithValue("@eid", eid.Text)

cmd.Parameters.AddWithValue("@ename", ename.Text)

cmd.Parameters.AddWithValue("@etell", ephone.Text)

cmd.Parameters.AddWithValue("@eadd", eadd.Text)

cmd.Parameters.AddWithValue("@dt", Rdate.Value)

cmd.Parameters.AddWithValue("@ref", ref.Text)

cmd.Parameters.AddWithValue("@rtel", reftell.Text)

cmd.Parameters.AddWithValue("@sel", esalery.Text)

connect()

If MsgBox(type & " Record of Customer :- " & ename.Text) Then

cmd.ExecuteNonQuery()

loadata()

clear()

End If

Else

MessageBox.Show("Please Fill Blanks", "Financial Management System", MessageBoxButtons.OK, MessageBoxIcon.Information)

End If

Catch ex As Exception

MsgBox(Err.Description)

End Try

cmd.Parameters.Clear()

End Sub

Private Sub clear()

eid.Clear()

ename.Clear()

ephone.Clear()

ref.Clear()

Rdate.Value = Today

eadd.Clear()

esalery.Clear()

reftell.Clear()

End Sub

Private Function chck() As Boolean

Return eid.Text <> "" And ename.Text <> "" And ephone.Text <> "" And eadd.Text <> "" And esalery.Text <> "" And Rdate.Text <> "" And ref.Text <> "" And reftell.Text <> ""

End Function

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

sql("delete")

End Sub

Private Sub DataGridView1\_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick

Try

eid.Text = DataGridView1.CurrentRow.Cells(0).Value

ename.Text = DataGridView1.CurrentRow.Cells(1).Value

ephone.Text = DataGridView1.CurrentRow.Cells(2).Value

eadd.Text = DataGridView1.CurrentRow.Cells(3).Value

ref.Text = DataGridView1.CurrentRow.Cells(4).Value

reftell.Text = DataGridView1.CurrentRow.Cells(5).Value

esalery.Text = DataGridView1.CurrentRow.Cells(6).Value

Rdate.Text = DataGridView1.CurrentRow.Cells(7).Value

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

Form1.Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

Main.instance.bringtofront()

Main.instance.main\_load()

End Sub

Private Sub DataGridView1\_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick

End Sub

End Class

* **Supplier Registration**

Imports System.Data.SqlClient

Public Class sup1

Public Shared \_instance As New sup1

Public Shared ReadOnly Property instance

Get

If \_instance Is Nothing Or \_instance.IsDisposed Then

\_instance = New sup1

End If

Return \_instance

End Get

End Property

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Try

If Check() Then

da.InsertCommand = New SqlCommand

da.InsertCommand.Connection = con

da.InsertCommand.CommandText = "insert into supplier values(@id,@name,@tell,@add,@date)"

da.InsertCommand.Parameters.AddWithValue("@id", sid.Text)

da.InsertCommand.Parameters.AddWithValue("@name", sname.Text)

da.InsertCommand.Parameters.AddWithValue("@tell", stell.Text)

da.InsertCommand.Parameters.AddWithValue("@add", sadd.Text)

da.InsertCommand.Parameters.AddWithValue("@date", Rdate.Value)

da.InsertCommand.ExecuteNonQuery()

MsgBox("Supplier Saved")

clear()

getId()

End If

Catch ex As Exception

MsgBox("Error Occured")

End Try

End Sub

Private Sub sup1\_Load(sender As Object, e As EventArgs) Handles Me.Load

getid()

End Sub

Sub getId()

Try

cmd.Connection = con

cmd.CommandText = "select case when max(sid) is null then 1 else max(sid)+1 end from supplier"

connect()

sid.Text = cmd.ExecuteScalar

Catch ex As Exception

MsgBox(Err.Description)

End Try

End Sub

Private Sub clear()

sid.Clear()

sname.Clear()

stell.Clear()

sadd.Clear()

Rdate.Value = Today

End Sub

Private Function Check() As Boolean

Return sname.Text <> "" And stell.Text <> "" And sadd.Text <> "" And Rdate.Text <> ""

End Function

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

Try

If Check() Then

da.InsertCommand = New SqlCommand

da.InsertCommand.Connection = con

da.InsertCommand.CommandText = "insert into supplier values(@id,@name,@tell,@add,@date)"

da.InsertCommand.Parameters.AddWithValue("@id", sid.Text)

da.InsertCommand.Parameters.AddWithValue("@name", sname.Text)

da.InsertCommand.Parameters.AddWithValue("@tell", stell.Text)

da.InsertCommand.Parameters.AddWithValue("@add", sadd.Text)

da.InsertCommand.Parameters.AddWithValue("@date", Rdate.Value)

da.InsertCommand.ExecuteNonQuery()

MsgBox("Supplier Saved")

clear()

getId()

Form1.Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

Main.instance.bringtofront()

Main.instance.Main\_Load()

End If

Catch ex As Exception

MsgBox("Error Occured")

End Try

End Sub

End Class

* **Edit Supplier Information**

Imports System.Data.SqlClient

Public Class sup2

Public Shared \_instance As New sup2

Public Shared ReadOnly Property instance

Get

If \_instance Is Nothing Or \_instance.IsDisposed Then

\_instance = New sup2

End If

Return \_instance

End Get

End Property

Public Sub sup2\_Load() Handles MyBase.Load

filldata()

End Sub

Private Sub filldata()

Try

ds.Clear()

da.SelectCommand = New SqlCommand

da.SelectCommand.Connection = con

da.SelectCommand.CommandText = "select \* from Viewsupplier"

da.Fill(ds, "supp")

DataGridView1.DataSource = ds.Tables("supp")

DataGridView1.Columns(0).Width = 50

DataGridView1.Columns(0).DefaultCellStyle.Alignment = ContentAlignment.MiddleCenter

Catch ex As Exception

MsgBox("Error Occured")

End Try

End Sub

Private Sub DataGridView1\_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick

Try

sid.Text = DataGridView1.CurrentRow.Cells(0).Value

Sname.Text = DataGridView1.CurrentRow.Cells(1).Value

Stell.Text = DataGridView1.CurrentRow.Cells(2).Value

sadd.Text = DataGridView1.CurrentRow.Cells(3).Value

Rdate.Text = DataGridView1.CurrentRow.Cells(4).Value

Catch ex As Exception

MsgBox("Error Occured")

End Try

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

Form1.Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

Main.instance.bringtofront()

Main.instance.main\_load()

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

SQlQUery("update")

End Sub

Sub SQlQUery(type As String)

Try

If Check() Then

cmd.Connection = con

cmd.CommandText = "exec Suppliers @sid,@sname,@stell,@sadd,@dt,'" & type & "'"

cmd.Parameters.AddWithValue("@sid", sid.Text)

cmd.Parameters.AddWithValue("@sname", Sname.Text)

cmd.Parameters.AddWithValue("@stell", Stell.Text)

cmd.Parameters.AddWithValue("@sadd", sadd.Text)

cmd.Parameters.AddWithValue("@dt", Rdate.Value)

connect()

cmd.ExecuteNonQuery()

MsgBox("Supplier " & type & "d")

clear()

filldata()

cmd.Parameters.Clear()

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub clear()

sid.Clear()

sname.Clear()

stell.Clear()

sadd.Clear()

Rdate.Value = Today

End Sub

Private Function Check() As Boolean

Return sname.Text <> "" And stell.Text <> "" And sadd.Text <> "" And Rdate.Text <> ""

End Function

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

If MessageBox.Show("You Are Deleting : " & Sname.Text, "Financial Management System", MessageBoxButtons.YesNo, MessageBoxIcon.Question) = DialogResult.Yes Then

SQlQUery("delete")

End If

End Sub

Private Sub DataGridView1\_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick

End Sub

End Class

* **Income Form**

Public Class cashin

Public Shared \_instance As New cashin

Public Shared ReadOnly Property instance

Get

If \_instance Is Nothing Or \_instance.IsDisposed Then

\_instance = New cashin

End If

Return \_instance

End Get

End Property

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

query("deposit")

End Sub

Sub query(type As String)

Try

If chck() Then

cmd.Connection = con

cmd.CommandText = "exec Cashinfo @name,@desc,@date,@amount,'" & type & "'"

cmd.Parameters.AddWithValue("@name", Icname.Text)

cmd.Parameters.AddWithValue("@desc", icdesc.Text)

cmd.Parameters.AddWithValue("@amount", icamount.Text)

cmd.Parameters.AddWithValue("@date", DDate.Value)

connect()

cmd.ExecuteNonQuery()

MsgBox("Record Saved")

cmd.Parameters.Clear()

clear()

Else

MsgBox("Plz Fill Blanks")

End If

Catch ex As Exception

MsgBox("Error")

End Try

End Sub

Private Function chck() As Boolean

Return Icname.Text <> "" And icdesc.Text <> "" And icamount.Text <> "" And IsDate(DDate.Text)

End Function

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

query("deposit")

Form1.Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

Main.instance.bringtofront()

Main.instance.Main\_Load()

End Sub

Sub clear()

Icname.Clear()

icdesc.Clear()

icamount.Clear()

DDate.Value = Today

End Sub

Private Sub cashin\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub

End Class

* **Cash Out or Costs**

Public Class cashout

Public Shared \_instance As New cashout

Public Shared ch As Double

Dim bl As Boolean = True

Public Shared ReadOnly Property instance

Get

If \_instance Is Nothing Or \_instance.IsDisposed Then

\_instance = New cashout

End If

Return \_instance

End Get

End Property

Sub query(type As String)

Try

If chck() Then

cmd.Connection = con

cmd.CommandText = "EXEC cashInfo @name,@desc,@date,@amount,'" & type & "'"

cmd.Parameters.AddWithValue("@name", coname.Text)

cmd.Parameters.AddWithValue("@desc", codesc.Text)

cmd.Parameters.AddWithValue("@amount", coamount.Text)

cmd.Parameters.AddWithValue("@date", ddate.Value)

connect()

If coamount.Text <= ch And ch <> 0 Then

cmd.ExecuteNonQuery()

MsgBox("Record Saved")

clear()

cashout\_Load()

bl = True

Else

MsgBox("Insufficient Balance")

coamount.Focus()

bl = False

End If

Else

MsgBox("Plz Fill Blanks")

End If

Catch ex As Exception

MsgBox("Error")

End Try

cmd.Parameters.Clear()

End Sub

Private Function chck() As Boolean

Return coname.Text <> "" And codesc.Text <> "" And coamount.Text <> "" And IsDate(ddate.Text)

End Function

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

query("withDraw")

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

query("withDraw")

If bl Then

Form1.Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

Main.instance.bringtofront()

Main.instance.Main\_Load()

End If

End Sub

Sub clear()

coname.Clear()

codesc.Clear()

coamount.Clear()

DDate.Value = Today

End Sub

Public Sub cashout\_Load() Handles MyBase.Load

Main.instance.Main\_Load()

lblCash.Text = "Total Cash :- $ " & ch

clear()

End Sub

End Class

* **Debts**

Imports System.Data.SqlClient

Public Class ar

Public Shared \_instance As New ar

Dim av As New AutoCompleteStringCollection

Dim drr As SqlDataReader

Dim dap As New SqlDataAdapter

Dim dss As DataSet

Dim cmd As New SqlCommand

Public Shared ReadOnly Property instance

Get

If \_instance Is Nothing Or \_instance.IsDisposed Then

\_instance = New ar

End If

Return \_instance

End Get

End Property

Private Sub ar\_Load(sender As Object, e As EventArgs) Handles Me.Load

TabControl1.Appearance = TabAppearance.FlatButtons

TabControl1.ItemSize = New Point(0, 1)

TabControl1.SizeMode = TabSizeMode.Fixed

dataload()

End Sub

Sub dataload()

Try

Dim dss As New DataSet

da.SelectCommand = New SqlCommand

da.SelectCommand.Connection = con

da.SelectCommand.CommandText = "select \* from debts"

da.Fill(dss, "debts")

DataGridView1.DataSource = dss.Tables("debts")

DataGridView1.Columns("amount").DefaultCellStyle.Format = FormatCurrency(0, 0)

Catch ex As Exception

MsgBox("Error")

End Try

End Sub

Private Sub Button7\_Click(sender As Object, e As EventArgs) Handles Button7.Click

TabControl1.SelectedIndex = 1

dataload()

lbltop.Text += " / New Dept"

End Sub

Private Sub Button6\_Click(sender As Object, e As EventArgs) Handles Button6.Click

TabControl1.SelectedIndex = 2

dataload()

lbltop.Text += " / Recieve Dept"

End Sub

Private Sub Label14\_Click(sender As Object, e As EventArgs) Handles Label14.Click

TabControl1.SelectedIndex = 0

lbltop.Text = "Dept Department"

End Sub

Private Sub Label15\_Click(sender As Object, e As EventArgs) Handles Label15.Click

TabControl1.SelectedIndex = 0

lbltop.Text = "Dept Department"

End Sub

Private Sub txtsearch\_GotFocus(sender As Object, e As EventArgs) Handles txtsearch.GotFocus

txtsearch.Clear()

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

Try

If txtsearch.Text <> "" Then

cmd.Connection = con

cmd.CommandText = "select cid,cname from customers where ctell ='" & txtsearch.Text & "'"

connect()

drr = cmd.ExecuteReader

If drr.Read Then

cid.Text = drr(0)

cname.Text = drr(1)

drr.Close()

cmd.CommandText = "select case when sum(dr)-sum(cr) is null then 0 else sum(dr)-sum(cr) end from acountRec where cid=" & cid.Text & ""

txtbalance.Text = cmd.ExecuteScalar()

Else

MsgBox("We cannot Fined Customer Related To Specified Number", MsgBoxStyle.Information)

End If

drr.Close()

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

drr.Close()

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

query("insert")

End Sub

Sub query(type As String)

Try

If Check() Then

cmd.Connection = con

cmd.CommandText = type & " into acountRec(cid,cname,date,dr) values(@id,@name,@date,@dr)"

cmd.Parameters.AddWithValue("@id", cid.Text)

cmd.Parameters.AddWithValue("@name", cname.Text)

cmd.Parameters.AddWithValue("@date", Rdate.Value)

cmd.Parameters.AddWithValue("@dr", txtamount.Text)

connect()

If txtamount.Text <> 0 Then

cmd.ExecuteNonQuery()

MsgBox("Debt Recorded")

clear()

Else

MsgBox("You cannot Debt 0")

End If

Else

MsgBox("Plz Fill Blanks")

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

cmd.Parameters.Clear()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

query("insert")

Form1.Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

Main.instance.bringtofront()

Main.instance.Main\_Load()

End Sub

Private Sub DataGridView1\_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick

Try

CID1.Text = DataGridView1.CurrentRow.Cells(0).Value

cname1.Text = DataGridView1.CurrentRow.Cells(1).Value

blnce.Text = DataGridView1.CurrentRow.Cells(2).Value

Catch ex As Exception

MsgBox("Error")

End Try

End Sub

Private Sub Button5\_Click(sender As Object, e As EventArgs) Handles Button5.Click

Try

If chck2() Then

cmd.Connection = con

cmd.CommandText = "insert into acountRec(cid,cname,date,cr) values(@id,@name,@date,@cr); insert into cash(cashname,descr,cdate,dr) values('Debit Recieved','From: '+@name,@date,@cr)"

cmd.Parameters.AddWithValue("@id", CID1.Text)

cmd.Parameters.AddWithValue("@name", cname1.Text)

cmd.Parameters.AddWithValue("@date", Rdate1.Value)

cmd.Parameters.AddWithValue("@cr", txtamount1.Text)

connect()

If CInt(txtamount1.Text) <= blnce.Text And CInt(txtamount1.Text) <> 0 Then

cmd.ExecuteNonQuery()

MsgBox("Debt Received")

dataload()

clear()

Else

MsgBox("Ma Bixi Kartid Lacagta Lagugu Leyahay lacag Ka badan")

End If

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

cmd.Parameters.Clear()

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

Form1.Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

Main.instance.bringtofront()

Main.instance.Main\_Load()

End Sub

Sub clear()

cid.Clear()

cname.Clear()

txtbalance.Clear()

txtamount.Clear()

blnce.Clear()

CID1.Clear()

cname1.Clear()

txtamount1.Clear()

End Sub

Private Function Check() As Boolean

Dim a As Boolean = False

If IsNumeric(txtamount.Text) Then

a = True

ErrorProvider1.Clear()

Else

ErrorProvider1.SetError(txtamount, "Amount Must be Number")

End If

Return a

End Function

Private Function chck2() As Boolean

Dim a As Boolean = False

If IsNumeric(txtamount1.Text) Then

a = True

ErrorProvider1.Clear()

Else

ErrorProvider1.SetError(txtamount1, "The Amount Must BE Number")

End If

Return a

End Function

Private Sub TabPage1\_Click(sender As Object, e As EventArgs) Handles TabPage1.Click

End Sub

End Class

* **Liability**

Imports System.Data.SqlClient

Public Class ap

Public Shared \_instance As New ap

Public Shared ReadOnly Property instance

Get

If \_instance Is Nothing Or \_instance.IsDisposed Then

\_instance = New ap

End If

Return \_instance

End Get

End Property

Private Sub Button7\_Click(sender As Object, e As EventArgs) Handles Button7.Click

TabControl1.SelectedIndex = 1

lbltop.Text += " / New "

End Sub

Private Sub Button6\_Click(sender As Object, e As EventArgs) Handles Button6.Click

TabControl1.SelectedIndex = 2

dataload()

lbltop.Text += " / Pay Loan "

End Sub

Private Sub Label13\_Click(sender As Object, e As EventArgs) Handles Label13.Click

TabControl1.SelectedIndex = 0

lbltop.Text = "Liability Department"

End Sub

Private Sub Label14\_Click(sender As Object, e As EventArgs) Handles Label14.Click

TabControl1.SelectedIndex = 0

lbltop.Text = "Liability Department"

End Sub

Sub LoadTocombo()

Try

cmd.Connection = con

cmd.CommandText = "select sname from supplier"

connect()

dr = cmd.ExecuteReader

While dr.Read

ComboBox1.Items.Add(dr(0))

End While

Catch ex As Exception

MsgBox("Error")

End Try

dr.Close()

End Sub

Private Sub ap\_Load(sender As Object, e As EventArgs) Handles Me.Load

LoadTocombo()

dataload()

End Sub

Private Sub ComboBox1\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ComboBox1.SelectedIndexChanged

Try

cmd.Connection = con

cmd.CommandText = "select sid,sname from supplier where sname ='" & ComboBox1.SelectedItem & "'"

connect()

dr = cmd.ExecuteReader

If dr.Read Then

Sid.Text = dr(0)

Sname.Text = dr(1)

dr.Close()

cmd.CommandText = "select case when sum(CR) - sum(Dr) is null then 0 else sum(Cr)- sum(Dr) end from accountPay where sid=" & Sid.Text & ""

balance.Text = cmd.ExecuteScalar()

Else

MsgBox("We cannot Fined Customer Related To Specified Supplier", MsgBoxStyle.Information)

End If

Catch ex As Exception

MsgBox("Error")

End Try

dr.Close()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

Form1.Panel4.Controls.Add(Main.instance)

Main.instance.dock = DockStyle.Fill

Main.instance.bringtofront()

Main.instance.Main\_Load()

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

'Try

cmd.Connection = con

cmd.CommandText = "insert into accountPay(sid,sname,date,cr) values(@id,@name,@dt,@amount)"

cmd.Parameters.AddWithValue("@id", Sid.Text)

cmd.Parameters.AddWithValue("@name", Sname.Text)

cmd.Parameters.AddWithValue("@dt", Rdate.Value)

cmd.Parameters.AddWithValue("@amount", txtamount.Text)

If txtamount.Text <> 0 Then

cmd.ExecuteNonQuery()

MsgBox("Liability Saved")

dataload()

clear()

End If

dr.Close()

'Catch ex As Exception

' MsgBox("Error")

'End Try

dr.Close()

cmd.Parameters.Clear()

End Sub

Sub clear()

Sid.Clear()

Sname.Clear()

balance.Clear()

txtamount.Clear()

Rdate.Value = Today

Rdate1.Value = Today

sname1.Clear()

balance1.Clear()

txtamount1.Clear()

End Sub

Sub dataload()

Try

ds.Clear()

da.SelectCommand = New SqlCommand

da.SelectCommand.Connection = con

da.SelectCommand.CommandText = "select \* from liability"

da.Fill(ds, "lb")

DataGridView1.DataSource = ds.Tables("lb")

DataGridView1.Columns("amount").DefaultCellStyle.Format = FormatCurrency(0, 0)

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Dim id As Integer

Private Sub DataGridView1\_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick

Try

id = DataGridView1.CurrentRow.Cells(0).Value

sname1.Text = DataGridView1.CurrentRow.Cells(1).Value

balance1.Text = DataGridView1.CurrentRow.Cells(2).Value

Catch ex As Exception

End Try

End Sub

Private Sub Button5\_Click(sender As Object, e As EventArgs) Handles Button5.Click

'Try

If True Then

cmd.Connection = con

cmd.CommandText = "insert into accountPay(sid,sname,date,dr) values(@id,@name,@date,@cr); insert into cash(cashname,descr,cdate,cr) values('Loan Payed','To: '+@name,@date,@cr)"

cmd.Parameters.AddWithValue("@id", id)

cmd.Parameters.AddWithValue("@name", sname1.Text)

cmd.Parameters.AddWithValue("@date", Rdate1.Value)

cmd.Parameters.AddWithValue("@cr", txtamount1.Text)

connect()

If CInt(txtamount1.Text) <= balance1.Text And CInt(txtamount1.Text) > 0 Then

If CInt(txtamount1.Text) <= caSH1 Then

cmd.ExecuteNonQuery()

MsgBox("Loan Payed")

dataload()

clear()

Else

MsgBox("Lacagta Aad Haysato Ma dhamo")

End If

Else

MsgBox("Ma Bixi Kartid Lacagta Lagugu Leyahay lacag Ka badan")

End If

End If

'Catch ex As Exception

' MsgBox(ex.Message)

'End Try

cmd.Parameters.Clear()

End Sub

Private Sub TabPage3\_Click(sender As Object, e As EventArgs) Handles TabPage3.Click

End Sub

End Class

* **Expense**

Public Class ex

Public Shared \_instance As New ex

Public Shared ReadOnly Property instance

Get

If \_instance Is Nothing Or \_instance.IsDisposed Then

\_instance = New ex

End If

Return \_instance

End Get

End Property

Dim x1 As Integer

Dim y1 As Integer

Private Sub ComboBox1\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles CMBTYPE.SelectedIndexChanged

If CMBTYPE.SelectedItem = "Salary" Then

'ComboBox1.Location = LBL

'Label6.Location = New Point(100, 50)

empId.Visible = True

empIDlabel.Visible = True

Else

'ComboBox1.Location = New Point(205, 55)

'Label6.Location = New Point(131, 58)

empId.Visible = False

empIDlabel.Visible = False

End If

End Sub

Private Sub Button7\_Click(sender As Object, e As EventArgs) Handles Button7.Click

TabControl1.SelectedIndex = 1

End Sub

Private Sub Button6\_Click(sender As Object, e As EventArgs) Handles Button6.Click

TabControl1.SelectedIndex = 2

End Sub

Private Sub Label16\_Click(sender As Object, e As EventArgs) Handles Label16.Click

TabControl1.SelectedIndex = 0

End Sub

Private Sub Label15\_Click(sender As Object, e As EventArgs) Handles Label15.Click

TabControl1.SelectedIndex = 0

End Sub

Private Sub ex\_Load(sender As Object, e As EventArgs) Handles Me.Load

x1 = CMBTYPE.Location.X

y1 = CMBTYPE.Location.Y

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Try

If True Then

End If

Catch ex As Exception

End Try

End Sub

Private Sub TabPage3\_Click(sender As Object, e As EventArgs) Handles TabPage3.Click

End Sub

End Class

**RECOMANDATION**

The following were the recommendations on how Liibaan company could

improve its productivity by managing its costs properly:

1. The company should establish a central data base for archiving the financial records to enable easy monitoring, decision making and proper finance management.
2. financial management procedures should be put in place to allow Proper management of finance in a cost-effective manner. The company should have a financial retention schedule; this would Help in retaining active and useful information and the disposal of an unwanted costs and that of low or no value to the company.
3. **CONCLUTION**

It was revealed that production inventory is the most financial management system used at **LIBAN SUPER MARKET** company. Production Inventory plays a very big role in the performance of any organization in this age characterized by total cost management.

financial management effectiveness adds value to the organization by allowing decision to be made faster and allowing production without hindrances as much as possible.

It was also noted that the company enjoys economic Oder quantity as the best control technique and this is attributed to how the method determine the optimal order that minimizes quality, costs and product availability

The efficiency of any control technique ensures that resources of the organization are managed with the least cost. The performance of any organization therefore depends on the achieved production targets, how much capital is held in stocks which can be achieved by a well-managed inventory technique used.