

Appendix A

Program codes of Application Graphical User Interface for SDSS_IUWM

Main Page Developed for IUWM

main_form.vb

```
PublicClass main_form
```

```
PrivateSub main_form_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles MyBase.Load  
EndSub
```

```
PrivateSub BasicDataToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles BasicDataToolStripMenuItem.Click  
BasicData.Show()  
EndSub
```

```
PrivateSub PopulationForecastingToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles PopulationForecastingToolStripMenuItem1.Click  
Population_Forecasting.Show()  
EndSub
```

```
PrivateSub ExitToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles ExitToolStripMenuItem1.Click  
Me.Close()  
EndSub
```

```
PrivateSub DomsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs)  
DomesticDemand.Show()  
EndSub
```

```
PrivateSub InstitutionalDemandToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs)  
Institutional_Demand.Show()  
EndSub
```

```
PrivateSub IndustrialDemandToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs)  
Industrialdemand.Show()  
EndSub
```

```
PrivateSub AgriculturalDemandToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs)  
Agricultural_Demand.Show()  
EndSub
```

```
PrivateSub InstitutionalDemandToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs)  
Livestock_Demand_cal.Show()  
EndSub
```

```
PrivateSub TotalDemandToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TotalDemandToolStripMenuItem.Click
    Water_Demand.Show()
EndSub
```

```
PrivateSub WaterScarcityToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Water_Scarcity.Show()
EndSub
```

```
PrivateSub WaterSupplyToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles WaterSupplyToolStripMenuItem1.Click
    WaterSupply.Show()
EndSub
```

```
PrivateSub WaterQualityToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles WaterQualityToolStripMenuItem.Click
    dwqi.Show()
EndSub
```

```
PrivateSub HelpToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles HelpToolStripMenuItem.Click

EndSub
```

```
PrivateSub WaterSupplyToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles WaterSupplyToolStripMenuItem.Click

EndSub
```

```
PrivateSub WaterDemandToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles WaterDemandToolStripMenuItem.Click

EndSub
```

```
PrivateSub WastewaterGeToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

EndSub
```

```
PrivateSub ToolStripComboBox2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

EndSub
```

```
PrivateSub ZoneWiseScarcityToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

EndSub
```

```
PrivateSub ToolStripComboBox1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

EndSub
```

```
PrivateSub ZonewiseGenerationToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

EndSub
```

```
PrivateSub TreatedWWReuseToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TreatedWWReuseToolStripMenuItem.Click
```

```
EndSub
```

```
PrivateSub SelectTWWReuseToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SelectTWWReuseToolStripMenuItem.Click  
    RWH_Potential.Show()
```

```
EndSub
```

```
PrivateSub BestTreatmentTechnToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BestTreatmentTechnToolStripMenuItem.Click  
    'WW_TT_Selection.Show()  
    Suggested_Tech.Show()
```

```
EndSub
```

```
PrivateSub WastewaterTreatmentToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles WastewaterTreatmentToolStripMenuItem.Click
```

```
EndSub
```

```
PrivateSub WWToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles WWToolStripMenuItem.Click  
    WW_Char.Show()
```

```
EndSub
```

```
PrivateSub RainWaterStoragePotentialToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RainWaterStoragePotentialToolStripMenuItem.Click  
    RWH_Potential.Show()
```

```
EndSub
```

```
PrivateSub WDPIToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles WDPIToolStripMenuItem.Click  
    WDPI.Show()
```

```
EndSub
```

```
PrivateSub HelpToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles HelpToolStripMenuItem1.Click  
    Help.Show()
```

```
EndSub
```

```
PrivateSub CityWaterDemandToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CityWaterDemandToolStripMenuItem.Click  
    Demand_Cal.Show()
```

```
EndSub
```

```
PrivateSub UWBEstimationToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles UWBEstimationToolStripMenuItem.Click  
    UWB_Est.Show()
```

```
EndSub
```

```
PrivateSub ScenarioAnalysisToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ScenarioAnalysisToolStripMenuItem.Click  
    S_Gen.Show()
```

```
EndSub
```

```
EndClass
```

Interface Developed for Basic Data Input

basic_data.vb

```
Imports MySql.Data.MySqlClient
Imports System.Data.SqlClient

PublicClass BasicData
Dim mysqlconn As MySqlConnection
Dim COMMAND As MySqlCommand

PrivateSub BasicData_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    mysqlconn = New MySqlConnection
    mysqlconn.ConnectionString = My.Settings.testConnectionString
Dim Reader As MySqlDataReader
Dim state_name AsString
    State.Items.Clear()

Try
    mysqlconn.Open()
Dim Query AsString
    Query = "select State_Name from mp.state"
'MessageBox.Show(Query)
    COMMAND = New MySqlCommand(Query, mysqlconn)
' MessageBox.Show("connection successfull")
    Reader = COMMAND.ExecuteReader

While Reader.Read
    state_name = Reader("State_Name")
    State.Items.Add(state_name)
EndWhile

    mysqlconn.Close()
' MessageBox.Show("connection unsuccessful")
Catch ex As MySqlException
    MessageBox.Show(ex.Message)
Finally
    mysqlconn.Dispose()
EndTry

EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    dwqi.Show()
EndSub

PrivateSub Button1_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    mysqlconn = New MySqlConnection
    mysqlconn.ConnectionString = "server = localhost;userid = mp;
password= mp;database= mp"
Dim pop AsString
Dim Reader As MySqlDataReader
'opening and closing connection
Try

    mysqlconn.Open()
Dim Query AsString
```

```

        Query = "select * from mp.basicdata where State ='"&
State.Text &"'and city='"& City.Text &"' and year ='"&
TextBox_year.Text &"'"
'MessageBox.Show(Query)
        COMMAND = New MySqlCommand(Query, mysqlconn)
' MessageBox.Show("connection successfull")

        Reader = COMMAND.ExecuteReader
Dim count As Integer
        count = 0

While Reader.Read

        pop = Reader("Population")
        population_data.Text = pop

EndWhile

        mysqlconn.Close()
' MessageBox.Show("connection unsuccessful")
Catch ex As MySqlException
        MessageBox.Show(ex.Message)
Finally
        mysqlconn.Dispose()

EndTry
EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
        mysqlconn = New MySqlConnection
        mysqlconn.ConnectionString = "server = localhost;userid = mp;
password= mp;database= mp"

Dim mQuery As String

        mQuery = "insert into mp.basicdata
(State,city,year,Population) values ('"& State.Text &"', '"&
City.Text &"', '"& TextBox_year.Text &"', '"& population_data.Text &"')"
        COMMAND = New MySqlCommand(mQuery, mysqlconn)

Try

        mysqlconn.Open()
        COMMAND.ExecuteNonQuery()
        mysqlconn.Close()
' MessageBox.Show("connection unsuccessful")
        MessageBox.Show("Data Saved")
' Button_Login.Visible = True
Catch ex As MySqlException
        MessageBox.Show("Data already exist")
Finally
        mysqlconn.Dispose()

EndTry
EndSub

PrivateSub State_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles State.SelectedIndexChanged

        mysqlconn = New MySqlConnection

```

```

        mysqlconn.ConnectionString = My.Settings.testConnectionString
Dim Reader1 As MySqlDataReader
Dim city_name AsString
'MessageBox.Show(mysqlconn.ConnectionString)
Try
    mysqlconn.Open()
Dim Query AsString
    Query = "select City_Name from mp.city_india where
State_id=" & State.SelectedIndex + 1

'MessageBox.Show(Query)
    COMMAND = New MySqlCommand(Query, mysqlconn)
    Reader1 = COMMAND.ExecuteReader
'MessageBox.Show("Reading successfull")
    City.Text = ""
    TextBox_year.Text = ""
    population_data.Text = ""
    City.Items.Clear()

While Reader1.Read
    city_name = Reader1("City_Name")
'MessageBox.Show(city_name)
    City.Items.Add(city_name)
EndWhile

    mysqlconn.Close()
' MessageBox.Show("connection unsuccessful")
Catch ex As MySqlException
    MessageBox.Show(ex.Message)
Finally
    mysqlconn.Dispose()
EndTry

EndSub

PrivateSub City_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles City.SelectedIndexChanged

    mysqlconn = New MySqlConnection
    mysqlconn.ConnectionString = My.Settings.testConnectionString
Dim Reader As MySqlDataReader
Dim population AsString
'MessageBox.Show(mysqlconn.ConnectionString)
Try
    mysqlconn.Open()
Dim Query AsString
    Query = "select Population from mp.city_india where
State_id=" & State.SelectedIndex + 1 & " and City_id=" &
City.SelectedIndex + 1

'MessageBox.Show(Query)
    COMMAND = New MySqlCommand(Query, mysqlconn)
    Reader = COMMAND.ExecuteReader
'MessageBox.Show("Reading successfull")
    Reader.Read()
    population = Reader("Population")
    TextBox_year.Text = 2011
    population_data.Text = population

    mysqlconn.Close()
' MessageBox.Show("connection unsuccessful")

```

```

Catch ex As MySqlConnection
    MessageBox.Show(ex.Message)
Finally
    mysqlconn.Dispose()
EndTry
EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Demand_Cal.Show()

EndSub

PrivateSub population_data_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles population_data.TextChanged

EndSub

PrivateSub ComboBox3_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox3.SelectedIndexChanged

EndSub

PrivateSub TextBox_supply_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs)

EndSub

PrivateSub ComboBox2_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox2.SelectedIndexChanged

EndSub

PrivateSub TextBox_scarcity_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs)

EndSub

PrivateSub TextBox_demand_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs)

EndSub

PrivateSub PopulationForecastingToolStripMenuItem_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs)
    Population_Forecasting.Show()
EndSub

PrivateSub ExitToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs)
Me.Close()
EndSub

PrivateSub GroupBox2_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GroupBox2.Enter

EndSub

```

```

PrivateSub Button1_Click_2(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Population_Forecasting.Show()
EndSub
EndClass

```

Page Developed for Population Forecasting

Population_Forecasting.vb

```

PublicClass Population_Forecasting
Inherits System.Windows.Forms.Form
Dim pn AsDouble
' Dim pop As Double
Dim Pop_dec(10) As TextBox
Dim rate_dec(10) AsDouble
Dim rate_decii(10) AsDouble
Dim year(10) As TextBox
PrivateSub Population_Forecasting_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
    setcontrolarray()
    Textp_1.Text = "207650"
    Textp_2.Text = "266002"
    Textp_3.Text = "355771"
    Textp_4.Text = "489864"
    Textp_5.Text = "671934"
    Textp_6.Text = "773865"
    Textp_7.Text = "1030863"
    Textp_8.Text = "1202443"
    Textp_9.Text = "1435113"

    year(1).Text = "1931"
    year(2).Text = "1941"
    year(3).Text = "1951"
    year(4).Text = "1961"
    year(5).Text = "1971"
    year(6).Text = "1981"
    year(7).Text = "1991"
    year(8).Text = "2001"
    year(9).Text = "2011"
Dim co AsInteger
For co = 2001 To 2050
    ComboBox1.Items.Add(co)
    ComboBox2.Items.Add(co)
Next

'Text1.Text = pop
'Text2.Text = year_p
'Text4.Text = pgr

EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
Dim n AsInteger = 1
Try

For n = 1 To 10
    Pop_dec(n).Clear()
    year(n).Clear()

```



```

Next
    ComboBox1.Text = ""
    TextBox1.Clear()
    TextBox6.Clear()
    TextBox5.Clear()
    TextBox2.Clear()
    TextBox3.Clear()
Catch ex As Exception
    MsgBox("No entries are filled", , "Message")
EndTry
EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
Dim rate, rate1, rate2, rate_ap, rate_iix, rate_iiy, decade AsDouble
Dim n, n1, count, count1, count2 AsInteger
Dim pop_AP, pop_GP, pop_II, pop_avg AsDouble

    count = 0 : count1 = 0 : count2 = 0 : rate1 = 1
'for checking whether all the input year data is differ by 10
Try
For n1 = 1 To 9
Dim year_diff AsInteger
If Microsoft.VisualBasic.StrComp(year(n1 + 1).Text, "") = 0 Then

Else

                year_diff = CInt(year(n1 + 1).Text) -
CInt(year(n1).Text)
If year_diff <> 10 Then
                    MsgBox("please check the years")
GoTo line1
EndIf
EndIf
Next

For n = 2 To 9
'If Microsoft.VisualBasic.StrComp(year(n).Text, "") = 0 Then
'Else
'Try

                rate_dec(n - 1) = (Cdbl(Pop_dec(n).Text) -
Cdbl(Pop_dec(n - 1).Text)) / Cdbl(Pop_dec(n - 1).Text)

'Catch ex As InvalidCastException
'MsgBox(" Please enter all the values")
'End Try
                count = count + 1
'End If
Next

For n = 2 To count + 1
If Microsoft.VisualBasic.StrComp(year(n).Text, "") = 0 Then
Else
                rate_decii(n - 1) = Cdbl(Pop_dec(n).Text) -
Cdbl(Pop_dec(n - 1).Text)
EndIf
Next

For n = 2 To count

```

```

        rate_iiy = rate_iiy + (rate_decii(n) - rate_decii(n -
1))
Next
        rate_iiy = rate_iiy / (count - 1)

'Calculate decadal growth rate
For count = 1 To count
If rate_dec(count) > 0 Then
    ratel = ratel * rate_dec(count)
    count1 = count1 + 1
EndIf
Next

rate = Math.Pow(ratel, 1 / count1)

'Try

rate_ap = (Pop_dec(count).Text - Pop_dec(1).Text) / (count
- 1)

rate2 = rate_ap / Pop_dec(1).Text

rate_iix = rate_ap

If Microsoft.VisualBasic.StrComp(ComboBox1.Text, "") = 0 Then
    MsgBox("Enter Forcasting Year")
GoTo line1
EndIf
    decade = (CDBl(Me.ComboBox1.Text) -
CDBl(year(count).Text)) / 10
    pop_GP = CDBl(Pop_dec(count).Text) * Math.Pow((1 + rate),
decade)
    pop_AP = Pop_dec(count).Text + (decade * rate_ap)
    pop_II = Pop_dec(count).Text + (decade * rate_iix) +
(decade * (decade + 1) * rate_iiy / 2)
Me.TextBox1.Text = CStr(Math.Round(pop_AP))
Me.TextBox2.Text = CStr(Math.Round(pop_GP))
Me.TextBox3.Text = CStr(Math.Round(pop_II))
    pop_avg = (pop_AP + pop_GP + pop_II) / 3
Me.avg_Pop.Text = CStr(Math.Round(pop_avg))
Me.TextBox5.Text = CStr(Math.Round(rate, 4) * 100)
Me.TextBox6.Text = CStr(Math.Round(rate2, 4) * 100)
Catch ex As Exception
    MsgBox("please enter all the values")
EndTry
line1:
EndSub

PrivateSub ToolTip1_Popup(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PopupEventArgs) Handles ToolTip1.Popup

    ToolTip1.AutoPopDelay = 5000
    ToolTip1.InitialDelay = 1000
    ToolTip1.ReshowDelay = 500
' Force the ToolTip text to be displayed whether or not the form is
active.
    ToolTip1.ShowAlways = True

    ToolTip1.SetToolTip(TextBox27, "Enter Population")

```

```

        ToolTip1.SetToolTip(TextBox28, "Enter Year")

EndSub
Sub setcontrolarray()
    Pop_dec(1) = Textp_1
    Pop_dec(2) = Textp_2
    Pop_dec(3) = Textp_3
    Pop_dec(4) = Textp_4
    Pop_dec(5) = Textp_5
    Pop_dec(6) = Textp_6
    Pop_dec(7) = Textp_7
    Pop_dec(8) = Textp_8
    Pop_dec(9) = Textp_9
    Pop_dec(10) = Textp_10
    year(1) = Text_y1
    year(2) = Text_y2
    year(3) = Text_y3
    year(4) = Text_y4
    year(5) = Text_y5
    year(6) = Text_y6
    year(7) = Text_y7
    year(8) = Text_y8
    year(9) = Text_y9
    year(10) = Text_y10
EndSub
PublicSub TextBox4_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles avg_Pop.TextChanged
    Dim avg_population AsDouble
        avg_population = Val(avg_Pop.Text)
EndSub
PrivateSub ComboBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox1.SelectedIndexChanged
    Dim MyVariablenew1 AsString

        MyVariablenew1 = ComboBox1.Text

EndSub

PrivateSub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
    Me.Close()
EndSub

PrivateSub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    Dim pn AsDouble
    Dim r AsDouble
    Dim po AsInteger
    Dim n1 AsShort
    'Estimate future population
    If TextBox27.Text = ""Or TextBox28.Text = ""Or ComboBox2.Text = ""Or
    TextBox29.Text = ""Then
        MsgBox("Fill all the entries", , "Message")
    Else
        n1 = Val(ComboBox2.Text) - Val(TextBox28.Text)
        'n = Val(Text2.Text) - Val(Text1.Text)
        po = Val(TextBox27.Text)

        r = CDb1(TextBox29.Text)
    EndIf
EndSub

```

```

'Text5.Text = Format(Text5.Text, "###0.##")
If ComboBox3.Text = "Arithmetical Progression Method"Then
    pn = po + (po * r * n1 / 100)
    TextBox30.Text = (Math.Round(pn, 0))
EndIf
If ComboBox3.Text = "Geometrical Progression Method"Then
    pn = po * ((1 + (r / 100)) ^ n1)
    TextBox30.Text = (Math.Round(pn, 0))
EndIf
EndIf
EndSub
EndClass

```

Page Developed for Water Deamand (WD) Estimation

Demand_Cal.vb

```

PublicClass Demand_Cal
Dim i AsInteger
Dim total_demandlhd As Int64
Dim fire_d As Int64
Dim Textile, Agro, Sugar, Paper, Eng, Cement, Steel, thermal, leather,
Drug, Other, Total_Industry As Int64
Dim off_dem, sch_dem, host_dem, hot_dem, rest_dem, hosp_dem, rail_dem,
mall_dem, int_dem As Int64
Dim hosp1_dem, hosp2_dem AsDouble
Dim population_forecast AsDouble
Dim water_demand1 = New Water_Demand()
PrivateSub tabPage1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tabPage1.Click

EndSub

PrivateSub Demand_Cal_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
Dim demand_drink, demand_bath, demand_wcloth, demand_wutensil,
demand_wcleanhouse, demand_lawn, demand_flush As Int64

'converting textbox data to integer

    demand_drink = Val(Text_drinking.Text)
    demand_bath = Val(Text_bath.Text)

    demand_wcloth = Val(Text_clothes.Text)
    demand_wutensil = Val(Text_utensils.Text)
    demand_wcleanhouse = Val(Text_cleaning.Text)
    demand_lawn = Val(Text_garden.Text)
    demand_flush = Val(Text_fl.Text)
    total_demandlhd = demand_drink + demand_bath + demand_wcloth +
demand_wutensil + demand_wcleanhouse + demand_lawn + demand_flush
'Text_Total.Text = total_demandlhd

    TextBox_popfor.Text = BasicData.population_data.Text
    TextBox_FF_POP.Text = BasicData.population_data.Text

EndSub

PrivateSub btnPrev_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrev.Click

```

```

        i -= 1

If i >= 0 Then

        TabControl1.SelectedIndex = i
Else
        TabControl1.SelectedIndex = 5
        i = 5
EndIf
EndSub

PrivateSub btnNext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNext.Click
        i += 1

If i <= 5 Then

        TabControl1.SelectedIndex = i
Else
        TabControl1.SelectedIndex = 0
        i = 0
EndIf

EndSub

PrivateSub GroupBox1_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GroupBox1.Enter

EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cal_domesticdemand.Click
Dim cal_domesticdemand As Int64
Dim var_pop As Int64
'If Population_Forecasting.TextBox1.Text.Length = 0 Then
'population_forecast = Val(TextBox_popfor.Text)
'Else
'population_forecast = Val(Population_Forecasting.TextBox1)
'End If

'TextBox_popfor.Text = Population_Forecasting.TextBox1.Text
'var_pop = Val(TextBox_popfor.Text)
If TextBox_popfor.Text = NothingThen
        MsgBox("Population Data Required")
EndIf

        var_pop = BasicData.population_data.Text

Try

        cal_domesticdemand = ((total_demandlhd * var_pop) /
1000000)
Catch ex As Exception
        MessageBox.Show("Population Forecasting is required")
        Population_Forecasting.Show()
EndTry

        Text_Total.Text = cal_domesticdemand
EndSub

```

```

PrivateSub TextBox_popfor_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox_popfor.TextChanged
'population_forecast = Val(TextBox_popfor.Text)
    TextBox_popfor.Text = BasicData.population_data.Text

EndSub

PrivateSub Cal_Fire_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Cal_Fire.Click
Dim var_pop As Int64
Dim pop_thousand As Int64

    var_pop = BasicData.population_data.Text
    pop_thousand = var_pop / 1000
    fire_d = 100 * Math.Sqrt(pop_thousand)
    Fire_demand.Text = fire_d / 1000

EndSub

PrivateSub Fire_Demand_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Fire_demand.TextChanged
'Fire_demand.Text = fire_d

EndSub

PrivateSub CheckBox1_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox_Fire.CheckedChanged

'Dim var_pop As Int64
If CheckBox_Fire.Checked = TrueThen
'var_pop = BasicData.population_data.Text
    Fire_demand.Text = 0.01 * Val(Text_Total.Text)

EndIf

EndSub

PrivateSub GroupBox4_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GroupBox4.Enter

EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
If CheckBox_Textile.Checked = TrueThen
    Agro = Val(TextBox_Agro.Text)

EndIf

If CheckBox_cement.Checked = TrueThen
    Cement = Val(TextBox_Cement.Text)

EndIf

If CheckBox_drug.Checked = TrueThen
    Drug = Val(TextBox_drug.Text)

EndIf

If CheckBox_eng.Checked = TrueThen
    Eng = Val(TextBox_Eng.Text)

EndIf

If CheckBox_leather.Checked = TrueThen
    leather = Val(TextBox_leather.Text)

EndIf

```

```

If CheckBox_other.Checked = TrueThen
    Other = Val(TextBox_other.Text)
EndIf

If CheckBox_paper.Checked = TrueThen
    Paper = Val(TextBox_Paper.Text)
EndIf

If CheckBox_steel.Checked = TrueThen
    Steel = Val(TextBox_Steel.Text)
EndIf

If CheckBox_sugar.Checked = TrueThen
    Sugar = Val(TextBox_sugar.Text)
EndIf

If CheckBox_Textile.Checked = TrueThen
    Textile = Val(TextBox_Textile.Text)
EndIf

If CheckBox_thermal.Checked = TrueThen
    thermal = Val(TextBox_Thermal.Text)
EndIf

    Total_Industry = Agro + Cement + Drug + Eng + leather + Other
+ Paper + Steel + Sugar + Textile + thermal
    TextBox_Total.Text = Total_Industry / 1000
EndSub

PrivateSub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    Total_WW_Gen.Show()
EndSub

PrivateSub Label40_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label40.Click

EndSub

PrivateSub Button_OD_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button_OD.Click
Dim FP As Int64
    FP = (Val(TextBox_Float.Text) * 135) / 1000000
    TextBox_OD.Text = Val(FP) + Val(TextBox_AOD.Text)
EndSub

PrivateSub Button_Tdem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button_Tdem.Click

    Label_dom.Text = Text_Total.Text
    Label_dom.Visible = True
    Label_Ind.Text = TextBox_Total.Text
    Label_Ind.Visible = True
    Label_Inst.Text = TextBox_Inst.Text
    Label_Inst.Visible = True
    Label_FF.Text = Fire_demand.Text
    Label_FF.Visible = True
    Label_OD.Text = TextBox_OD.Text
    Label_OD.Visible = True
    TextBox_TD_POP.Text = BasicData.population_data.Text

```

```

        TextBox_TotDem.Text = Val(Text_Total.Text) +
Val(TextBox_Inst.Text) + Val(TextBox_Total.Text) +
Val(Fire_demand.Text) + Val(TextBox_OD.Text)
EndSub

PrivateSub CheckBox_float_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox_float.CheckedChanged
If CheckBox_Fire.Checked = TrueThen
    TextBox_Float.Text = 0.05 * Val(Text_Total.Text)
'MsgBox(TextBox_Float.Text)
EndIf
EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
'off_dem = Val(TextBox_offfn.Text) * Val(TextBox_off.Text) * 45
    off_dem = Val(TextBox_offD.Text)
    TextBox_off.Text = Math.Round(off_dem / 150)
'sch_dem = Val(TextBox_schn.Text) * Val(TextBox_sch.Text) * 45
    sch_dem = Val(TextBox_schD.Text)
    TextBox_sch.Text = Math.Round(sch_dem / 150)
'host_dem = Val(TextBox_hosn.Text) * Val(TextBox_hos.Text) * 135
    host_dem = Val(TextBox_hosD.Text)
    TextBox_hos.Text = Math.Round(host_dem / 150)
'hot_dem = Val(TextBox_hotn.Text) * Val(TextBox_hot.Text) * 180
    hot_dem = Val(TextBox_hotD.Text)
    TextBox_hot.Text = Math.Round(hot_dem / 150)
'rest_dem = Val(TextBox_offfn.Text) * Val(TextBox_off.Text) * 45
    rest_dem = Val(TextBox_resD.Text)
    TextBox_res.Text = Math.Round(rest_dem / 150)
'hosp_dem = hosp1_dem + hosp2_dem
    hosp_dem = Val(TextBox_hosp1D.Text) + Val(TextBox_hosp2D.Text)
    MsgBox(hosp_dem)
    TextBox_hosp.Text = Math.Round(hosp_dem / 150)
'rail_dem = Val(TextBox_jn.Text) * 70 * 100000
    rail_dem = Val(TextBox_rjD.Text)
    TextBox_rj.Text = Math.Round(rail_dem / 150)
'mall_dem = Val(TextBox_mall.Text) * 15 * 500
    mall_dem = Val(TextBox_mD.Text)
    TextBox_m.Text = Math.Round(mall_dem / 150)
    int_dem = off_dem + sch_dem + host_dem + hot_dem + rest_dem +
hosp_dem + rail_dem + rail_dem + mall_dem
    TextBox_Inst.Text = Math.Round(int_dem / 1000000)

EndSub

PrivateSub CheckBox_m100_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox_m100.CheckedChanged
If CheckBox_m100.Checked = TrueThen
    hosp2_dem = Val(TextBox_hosp2D.Text)
EndIf
EndSub

PrivateSub CheckBox_l100_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox_l100.CheckedChanged
If CheckBox_l100.Checked = TrueThen
    hosp1_dem = Val(TextBox_hosp1D.Text)
EndIf
EndSub

```



```

PrivateSub Button3_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    WaterSupply.Show()
EndSub

PrivateSub Label56_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label56.Click

EndSub

PrivateSub TextBox_res_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox_res.TextChanged

EndSub

PrivateSub GroupBox3_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GroupBox3.Enter

EndSub

PrivateSub Label87_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label87.Click

EndSub
EndClass

```

Interface Developed for Water Supply (WS)

WaterSupply.vb

```

Imports System.Net.Mime.MediaTypeNames
Imports Excel = Microsoft.Office.Interop.Excel

PublicClass WaterSupply
    Dim surface_water, ground_water, alter_water AsDouble
    PrivateSub WaterSupply_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Label_supply.Visible = False
        Button3.Visible = False
    EndSub
    PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        MessageBox.Show("Data to be calculated")
    EndSub

    PublicSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        ground_water = TextBox_groundwater.Text
        surface_water = TextBox_surfacewater.Text
        alter_water = TextBox_SAS.Text
        Label_supply.Text = ground_water + surface_water
        Label_supply.Visible = True
    EndSub

    PrivateSub Label17_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label_supply.Click

    EndSub
EndSub

```

```

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Waterqualityindex.Show()
EndSub

PrivateSub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    WaterSupplyZone.Show()
EndSub

PrivateSub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    'Water_supply_zones.Show()
EndSub

PrivateSub Button5_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    map.Show()
EndSub

PrivateSub GroupBox1_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GroupBox1.Enter

EndSub

PrivateSub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
    finalmap.Show()
EndSub

PrivateSub Label17_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label17.Click

EndSub
EndClass

```

Interface Developed for Wastewater Generation Sub-module

Total_WW_Gen.vb

```

PublicClass Total_WW_Gen
Dim TotalWW As Int64
Dim Percent_Convert AsString
Dim Treat_Cap AsString
PrivateSub Total_WW_Gen_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
For co AsInteger = 70 To 90
    ComboBox1.Items.Add(co)
Next
For col AsInteger = 0 To 300 Step 10
    ComboBox2.Items.Add(col)
Next
    Name_City.Visible = True
    Name_City.Text = BasicData.City.Text
    Name_.Visible = True
    Name_.Text = BasicData.City.Text
EndSub

PrivateSub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TextBox1.TextChanged

```

```

EndSub

PrivateSub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    WW_Char.Show()
EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    BasicData.Show()
EndSub

PrivateSub TWW_Box_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs)

EndSub

PrivateSub UTWWBox_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles UTWW_Box.TextChanged

EndSub

PrivateSub TextBox4_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TextBox4.TextChanged

EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

EndSub

PrivateSub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
If (Treat_Cap > TextBox1.Text) Then
    MessageBox.Show(" Please Select Correct Treatment Capacity
of City ")

Else
    UTWW_Box.Text = TextBox1.Text - Treat_Cap
    TextBox4.Text = Math.Round(UTWW_Box.Text / 50)
EndIf
EndSub

PrivateSub ComboBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox1.SelectedIndexChanged

    Percent_Convert = ComboBox1.Text
EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    TextBox1.Text = (Demand_Cal.Text_Total.Text * Percent_Convert
/ 100)
EndSub

```

```

PrivateSub GroupBox1_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GroupBox1.Enter

EndSub

PrivateSub ComboBox2_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox2.SelectedIndexChanged
    Treat_Cap = ComboBox2.Text
EndSub

PrivateSub Name__Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Name_.Click

EndSub

PrivateSub Button_EWW_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button_EWW.Click

    TextBox1.Text = (Demand_Cal.Text_Total.Text * Percent_Convert
/ 100)
    Treat_Cap = ComboBox2.Text
If (Treat_Cap > TextBox1.Text) Then
    MessageBox.Show(" Please Select Correct Treatment Capacity
of City ")

Else
    UTWW_Box.Text = TextBox1.Text - Treat_Cap
    TextBox4.Text = UTWW_Box.Text / 50

EndIf

EndSub
EndClass

```

Interface Development for Wastewater Characteristics Input

WW_Char.vb

```

Imports MySql.Data.MySqlClient
Imports System.Data.SqlClient

PublicClass WW_Char
Dim MP_Conn, MP_Conn1 As MySqlConnection
Dim MP_Command As MySqlCommand
Dim MP_Reader As MySqlDataReader
Dim MP_Query AsString

PrivateSub WW_Char_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

EndSub

PrivateSub Submit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Submit.Click

    MP_Conn = New MySqlConnection
    MP_Conn.ConnectionString = My.Settings.testConnectionString
    MP_Conn1 = New MySqlConnection
    MP_Conn1.ConnectionString = My.Settings.testConnectionString

```

```

Dim _BOD_, _COD_, _TDS_, _TSS_, _DO_, _pH_, _EC_, _Sod_, _Nitrate_,
_Phosphorous_, _Boron_, _TC_ AsDouble
Dim _BOD1_, _COD1_, _TDS1_, _TSS1_, _DO1_, _EC1_, _Sod1_, _Nitrate1_,
_Phosphorous1_, _Boron1_, _TC1_ AsDouble
    _BOD_ = -999
    _COD_ = -999
    _TDS_ = -999
    _TSS_ = -999
    _DO_ = -999
    _pH_ = -999
    _EC_ = -999
    _Sod_ = -999
    _Nitrate_ = -999
    _Phosphorous_ = -999
    _Boron_ = -999
    _TC_ = -999

IfNot (BOD_.Text = String.Empty) Then
Double.TryParse(BOD_.Text, _BOD_)
'MessageBox.Show(_BOD_)
EndIf

IfNot (COD_.Text = String.Empty) Then
Double.TryParse(COD_.Text, _COD_)
EndIf

IfNot (TDS_.Text = String.Empty) Then
Double.TryParse(TDS_.Text, _TDS_)
EndIf

IfNot (TSS_.Text = String.Empty) Then
Double.TryParse(TSS_.Text, _TSS_)
EndIf

IfNot (DO_.Text = String.Empty) Then
Double.TryParse(DO_.Text, _DO_)
EndIf

IfNot (pH_.Text = String.Empty) Then
Double.TryParse(pH_.Text, _pH_)
EndIf

IfNot (EC_.Text = String.Empty) Then
Double.TryParse(EC_.Text, _EC_)
EndIf

IfNot (Sod_.Text = String.Empty) Then
Double.TryParse(Sod_.Text, _Sod_)
EndIf

IfNot (Nitrate_.Text = String.Empty) Then
Double.TryParse(Nitrate_.Text, _Nitrate_)
EndIf
IfNot (Phosphorous_.Text = String.Empty) Then
Double.TryParse(Phosphorous_.Text, _Phosphorous_)
EndIf

IfNot (Boron_.Text = String.Empty) Then
Double.TryParse(Boron_.Text, _Boron_)
EndIf

```

```

IfNot (TC_.Text = String.Empty) Then
Double.TryParse(TC_.Text, _TC_)
EndIf

Try
    MP_Conn.Open()

    MP_Query = "truncate table mp.char"
    MP_Command = New MySqlCommand(MP_Query, MP_Conn)
    MP_Command.ExecuteNonQuery()

    MP_Query = "insert into mp.Char
(BOD_, COD_, TDS_, TSS_, DO_, pH_, EC_, Sod_, Nitrate_, Phosphorous_, Boron_, TC_
) values ('"& _BOD_ &"', '"& _COD_ &"', '"& _TDS_ &"', '"& _TSS_ &"',
'"& _DO_ &"', '"& _pH_ &"', '"& _EC_ &"', '"& _Sod_ &"', '"& _Nitrate_
&"', '"& _Phosphorous_ &"', '"& _Boron_ &"', '"& _TC_ &"')"
    MP_Command = New MySqlCommand(MP_Query, MP_Conn)
    MP_Command.ExecuteNonQuery()

    MP_Query = "truncate table mp.tech_criteria_temp"
    MP_Command = New MySqlCommand(MP_Query, MP_Conn)
    MP_Command.ExecuteNonQuery()

    MP_Query = "select * from mp.tech_criteria"
    MP_Command = New MySqlCommand(MP_Query, MP_Conn)
    MP_Reader = MP_Command.ExecuteReader
'
    MessageBox.Show("1")

    MP_Conn1.Open()
While MP_Reader.Read
    _BOD1_ = (1 - MP_Reader(2)) * _BOD_
    _COD1_ = (1 - MP_Reader(3)) * _COD_
    _TDS1_ = (1 - MP_Reader(4)) * _TDS_
    _TSS1_ = (1 - MP_Reader(5)) * _TSS_
    _DO1_ = (1 - MP_Reader(6)) * _DO_
    _EC1_ = (1 - MP_Reader(7)) * _EC_
    _Sod1_ = (1 - MP_Reader(8)) * _Sod_
    _Nitratel_ = (1 - MP_Reader(9)) * _Nitrate_
    _Phosphorous1_ = (1 - MP_Reader(10)) * _Phosphorous_
    _Boron1_ = (1 - MP_Reader(11)) * _Boron_
    _TC1_ = (1 - MP_Reader(12)) * _TC_

    MP_Query = "insert into mp.tech_criteria_temp (ID,
Tech_Name,
BOD_, COD_, TDS_, TSS_, DO_, pH_, EC_, Sod_, Nitrate_, Phosphorous_, Boron_, TC_,
Cost_) values ('"& MP_Reader(0) &"', '"& MP_Reader(1) &"', '"& _BOD1_
&"', '"& _COD1_ &"', '"& _TDS1_ &"', '"& _TSS1_ &"', '"& _DO1_ &"', '"&
MP_Reader(13) &"', '"& _EC1_ &"', '"& _Sod1_ &"', '"& _Nitratel_
&"', '"& _Phosphorous1_ &"', '"& _Boron1_ &"', '"& _TC1_ &"', '"&
MP_Reader(14) &"')"
    MP_Command = New MySqlCommand(MP_Query, MP_Conn1)
    MP_Command.ExecuteNonQuery()
EndWhile
'
    MessageBox.Show("2")

    MP_Conn.Close()
    MP_Conn1.Close()

    MessageBox.Show("Data Stored")
Catch ex As MySqlException

```

```

        MessageBox.Show(ex.Message)
    Finally
        MP_Conn.Dispose()
        MP_Conn1.Dispose()
    EndTry

EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Suggested_Tech.Show()
EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Process.Start("C:\Users\Hp\Documents\Visual Studio
2008\Projects\Reuse.pdf")
    Dim FilePath AsString = "C:\Test.pdf"
    IfMy.Computer.FileSystem.FileExists(FilePath) Then
        Try
            Process.Start("acrobat", FilePath)
        Catch ex As Exception
            Try
                Process.Start("AcroRd32", FilePath)
            Catch ex2 As Exception
                Try
                    Process.Start(FilePath)
                Catch ex3 As Exception
                    MsgBox("Instal Acrobat Reader")
                EndTry
            EndTry
        EndTry
    Else
        'MsgBox("File not found.")
    EndIf

EndSub
EndClass

```

Interface Developed for Reuse Based Wastewater Treatment Technology Selection Sub-module

Suggested_Tech.vb

```

Imports MySql.Data.MySqlClient
Imports System.Data.SqlClient

PublicClass Suggested_Tech
    Dim MP_Conn As MySqlConnection
    Dim MP_Command As MySqlCommand
    Dim MP_Reader As MySqlDataReader
    Dim MP_Query AsString

    PrivateSub Suggested_Tech_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        MP_Conn = New MySqlConnection
        MP_Conn.ConnectionString = My.Settings.testConnectionString

    Dim reuse_name AsString
        CB_Reuse.Items.Clear()

```

```

Try
    MP_Conn.Open()
    MP_Query = "select ID, Reuse_Name from mp.reuse_cat"
    MP_Command = New MySqlCommand(MP_Query, MP_Conn)
    MP_Reader = MP_Command.ExecuteReader

While MP_Reader.Read
    reuse_name = MP_Reader("Reuse_Name")
    CB_Reuse.Items.Add(reuse_name)
EndWhile

    MP_Conn.Close()
' MessageBox.Show("connection unsuccessful")
Catch ex As MySqlException
    MessageBox.Show(ex.Message)
Finally
    MP_Conn.Dispose()
EndTry

EndSub

PrivateSub CB_Reuse_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CB_Reuse.SelectedIndexChanged
Dim _char_(12) AsDouble
Dim _reuse_(14) AsDouble
Dim _indx_(11) AsByte
Dim _isug_(8) AsByte
Dim _sug_(8) AsString
Dim _TC_(12) AsDouble
Dim _temp_(12) AsDouble
Dim i, j, k, l AsByte
'Dim _msg_ As String
Dim Satya AsBoolean
    MP_Conn = New MySqlConnection
    MP_Conn.ConnectionString = My.Settings.testConnectionString

    _sug_(1) = ""
    _sug_(2) = ""
    _sug_(3) = ""
    _sug_(4) = ""
    _sug_(5) = ""
    _sug_(6) = ""
    _sug_(7) = ""
    _sug_(8) = ""

    TB_1.Text = ""
    TB_2.Text = ""
    TB_3.Text = ""

Try
    MP_Conn.Open()
    MP_Query = "select * from mp.reuse_cat where ID = "&
CB_Reuse.SelectedIndex + 1
    MP_Command = New MySqlCommand(MP_Query, MP_Conn)
    MP_Reader = MP_Command.ExecuteReader
    MP_Reader.Read()
Double.TryParse(MP_Reader(0), _reuse_(0))
For i = 2 To 14
Double.TryParse(MP_Reader(i), _reuse_(i))

```



```

Next
        j = 0
For i = 2 To 12
If _reuse_(i) >= 0 Then
        j = j + 1
        _indx_(j) = i
EndIf
Next

'
'       _msg_ = " "
'       For i = 1 To j
'       _msg_ = _msg_ & " " & _reuse_( _indx_(i))
'       Next
'       MessageBox.Show(_msg_)
MP_Conn.Close()
MP_Conn.Open()

MP_Query = "select * from mp.tech_criteria_temp"
MP_Command = New MySqlCommand(MP_Query, MP_Conn)
MP_Reader = MP_Command.ExecuteReader
k = 0
l = 0

While MP_Reader.Read
        Satya = True
        k = k + 1
If j <> 0 Then
For i = 1 To j
If MP_Reader( _indx_(i) ) > _reuse_( _indx_(i) ) Then
        Satya = False
EndIf
Next
If Satya Then
If MP_Reader(13) < _reuse_(13) And MP_Reader(13) > _reuse_(14) Then
        Satya = False
EndIf
EndIf
If Satya Then
        l = l + 1
        _sug_(l) = MP_Reader(1)
'MessageBox.Show(MP_Reader(1))
EndIf
Else
If Satya Then
If MP_Reader(13) < _reuse_(13) And MP_Reader(13) > _reuse_(14) Then
        Satya = False
EndIf
EndIf
If Satya Then
        l = l + 1
        _sug_(l) = MP_Reader(1)
'MessageBox.Show(MP_Reader(1))
EndIf
EndIf
EndWhile

TB_1.Text = _sug_(1)
TB_2.Text = _sug_(2)
TB_3.Text = _sug_(3)

```

```

        MP_Conn.Close()

Catch ex As MySqlException
    MessageBox.Show(ex.Message)
Finally
    MP_Conn.Dispose()
EndTry

EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Process.Start("C:\Users\Hp\Documents\Visual Studio
2008\Projects\TreatmentCost.pdf")
Dim FilePath AsString = "C:\Test.pdf"
IfMy.Computer.FileSystem.FileExists(FilePath) Then
Try
    Process.Start("acrobat", FilePath)
Catch ex As Exception
Try
    Process.Start("AcroRd32", FilePath)
Catch ex2 As Exception
Try
    Process.Start(FilePath)
Catch ex3 As Exception
    MsgBox("Instal Acrobat Reader")
EndTry
EndTry
EndTry
Else
'MsgBox("File not found.")
EndIf

EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    S_Gen.Show()
EndSub
EndClass

```

Interface DDeveloped for Rainwater Runoff Estimation

RainWater_Runoff.vb

```

PublicClass RWH_Potential
Dim Built_Area, SubUrban_Area, Open_Area, Park_Area, Agri_Area,
Forest_Area AsDouble
Dim R_Built1, R_Sub1, R_Open1, R_Park1, R_Agr1, R_Forest1, Rainfall
AsDouble
Dim Tot_Runoff AsDouble
PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Rainfall = Int64.Parse(R_fall.Text) * 0.001
    Built_Area = Int64.Parse(Builtup.Text)
    SubUrban_Area = Int64.Parse(Suburban.Text)
    Open_Area = Int64.Parse(Open.Text)
    Park_Area = Int64.Parse(Park.Text)
    Agri_Area = Int64.Parse(Agri.Text)
    Forest_Area = Int64.Parse(Forest.Text)
    R_Built.Text = 0.95 * Built_Area * Rainfall

```

```

R_Sub.Text = 0.45 * SubUrban_Area * Rainfall
R_Open.Text = 0.5 * Open_Area * Rainfall
R_Park.Text = 0.3 * Park_Area * Rainfall
R_Agri.Text = 0.6 * Agri_Area * Rainfall
R_Forest.Text = 0.15 * Forest_Area * Rainfall
R_Built1 = Double.Parse(R_Built.Text)
R_Sub1 = Double.Parse(R_Sub.Text)
R_Open1 = Double.Parse(R_Open.Text)
R_Park1 = Double.Parse(R_Park.Text)
R_Agri1 = Double.Parse(R_Agri.Text)
R_Forest1 = Double.Parse(R_Forest.Text)
Tot_Runoff = R_Built1 + R_Sub1 + R_Open1 + R_Agri1 + R_Park1 +
R_Forest1
MsgBox(Tot_Runoff)
EndSub

PrivateSub RWH_Potential_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load

EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
R_Total.Text = Math.Round((R_Built1 + R_Sub1 + R_Open1 +
R_Park1 + R_Agri1 + R_Forest1) / 1000000)

EndSub

PrivateSub Suburban_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Suburban.TextChanged

EndSub

PrivateSub Label8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label8.Click

EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
RWstorage.Show()

EndSub

PrivateSub CheckBox1_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox_RW.CheckedChanged
If CheckBox_RW.Checked = TrueThen
R_fall.Text = 828
Builtup.Text = 58220000
Suburban.Text = 1394000
Open.Text = 3280000
Park.Text = 410000
Agri.Text = 5904000
Forest.Text = 246000

EndIf

EndSub
EndClass

```

Interface Developed for Water Supply Sustainability Analysis

UWB_Est.vb

```
PublicClass UWB_Est

PrivateSub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox_AS.TextChanged

EndSub

PrivateSub UWB_Est_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    S_Gen.Show()
EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    WDPI.Show()
EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    TextBox_WDP.Text = (TextBox_UWB.Text) / TextBox_FOS.Text
EndSub

PrivateSub Button_UWB_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button_UWB.Click
    TextBox_UWB.Text = TextBox_dem.Text - TextBox_OS.Text
    MsgBox("Urban Water Balance may show higher than govt. estimation")
EndSub

PrivateSub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

EndSub
EndClass
```

Interface Developed for Water for Development Planning Index (WDPI)

WDPI.vb

```
PublicClass WDPI
Dim UR, WW, FW, PRV, EP, SWQ, GWQ, AD, REL, CON, WSC, WWC, WWSW, TWW, RSC, RP, EE, RR, GRP, MAP, PP, PA AsDouble
Dim N_UR, N_WW, N_FW, N_PRV, N_EP, N_SWQ, N_GWQ, N_AD, N_REL, N_CON, N_WSC, N_WWC, N_WWSW, N_TWW, N_RSC, N_RP, N_EE, N_RR, N_GRP, N_MAP, N_PP, N_PA AsDouble
PrivateSub Label135_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label135.Click

EndSub
```

```

PrivateSub CheckBox_VNS_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox_VNS.CheckedChanged
If CheckBox_VNS.Checked = TrueThen
    Urb_rate.Text = 0.95
    Water_with.Text = 70
    Fresh_scar.Text = 90
    Poll_risk.Text = 70
    Eco_pres.Text = 10
    Surface_wq.Text = 65
    Ground_wq.Text = 90
    Adeq.Text = 100
    Reli.Text = 100
    Consum.Text = 27
    WS_coverage.Text = 65
    WW_coverage.Text = 30
    Separation_wsw.Text = 30
    Treated_ww.Text = 37.5
    Surface_rsc.Text = 20
    Reuse_pot.Text = 0
    Eco_eff.Text = 30
    Resource_rec.Text = 40
    Ground_rp.Text = 10
    Mgt_ap.Text = 30
    Pub_accept.Text = 20
    Pub_part.Text = 30

```

```
EndIf
```

```
EndSub
```

```

PrivateSub CheckBox2_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox_wt.CheckedChanged
If CheckBox_wt.Checked = TrueThen
    TextBox_w1.Text = 0.2
    TextBox_w2.Text = 0.4
    TextBox_w3.Text = 0.3
    TextBox_w4.Text = 0.1
    TextBox_w5.Text = 1
    TextBox_w6.Text = 0.5
    TextBox_w7.Text = 0.5
    TextBox_w8.Text = 0.4
    TextBox_w9.Text = 0.4
    TextBox_w10.Text = 0.2
    TextBox_w11.Text = 0.35
    TextBox_w12.Text = 0.35
    TextBox_w13.Text = 0.3
    TextBox_w14.Text = 0.1
    TextBox_w15.Text = 0.2
    TextBox_w16.Text = 0.2
    TextBox_w17.Text = 0.05
    TextBox_w18.Text = 0.15
    TextBox_w19.Text = 0.2
    TextBox_w20.Text = 0.4
    TextBox_w21.Text = 0.4
    TextBox_w22.Text = 0.2

```

```
EndIf
```

```
EndSub
```

```

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
Dim WDI AsDouble

```

```

        WDI = (TextBox_pres.Text) * (TextBox_wp.Text) +
        (TextBox_state.Text) * (TextBox_ws.Text) + (TextBox_res.Text) *
        (TextBox_wr.Text)
        TextBox_WDPI.Text = WDI

```

EndSub

```

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click

```

```

    UR = Urb_rate.Text
    WW = Water_with.Text
    FW = Fresh_scar.Text
    PRV = Poll_risk.Text
    EP = Eco_pres.Text
    SWQ = Surface_wq.Text
    GWQ = Ground_wq.Text
    AD = Adeq.Text
    REL = Reli.Text
    CON = Consum.Text
    WSC = WS_coverage.Text
    WWC = WW_coverage.Text
    WWSW = Separation_wsw.Text
    TWW = Treated_ww.Text
    RSC = Surface_rsc.Text
    RP = Reuse_pot.Text
    EE = Eco_eff.Text
    RR = Resource_rec.Text
    GRP = Ground_rp.Text
    MAP = Mgt_ap.Text
    PA = Pub_accept.Text
    PP = Pub_part.Text
    N_UR = ((3 - UR) / 3) * 10
    N_WW = (100 - WW) / 10
    N_FW = (100 - FW) / 10
    N_PRV = (100 - PRV) / 10
    N_EP = (100 - EP) / 10
    N_SWQ = SWQ / 10
    N_GWQ = GWQ / 10
    N_AD = AD / 10
    N_REL = REL / 10
    N_CON = ((100 - CON) / 100) * 10
    N_WSC = WSC / 10
    N_WWC = WWC / 10
    N_WWSW = WWSW / 10
    N_TWW = TWW / 10
    N_RSC = RSC / 10
    N_EE = EE / 10
    N_RP = (RP / 40) * 10
    N_RR = RR / 10
    N_GRP = GRP / 10
    N_MAP = MAP / 10
    N_PA = PA / 10
    N_PP = PP / 10
    TextBox_SI1.Text = N_UR
    TextBox_SI2.Text = N_WW
    TextBox_SI3.Text = N_FW
    TextBox_SI4.Text = N_PRV
    TextBox_SI5.Text = N_EP
    TextBox_SI6.Text = N_SWQ
    TextBox_SI7.Text = N_GWQ
    TextBox_SI8.Text = N_AD
    TextBox_SI9.Text = N_REL

```

```

    TextBox_SI10.Text = N_CON
    TextBox_SI11.Text = N_WSC
    TextBox_SI12.Text = N_WW
    TextBox_SI13.Text = N_WWSW
    TextBox_SI14.Text = N_TWW
    TextBox_SI15.Text = N_RSC
    TextBox_SI16.Text = N_EE
    TextBox_SI17.Text = N_RR
    TextBox_SI18.Text = N_RP
    TextBox_SI19.Text = N_GRP
    TextBox_SI20.Text = N_MAP
    TextBox_SI21.Text = N_PA
    TextBox_SI22.Text = N_PP

```

EndSub

```

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    TextBox_pres.Text = ((TextBox_SI1.Text) * (TextBox_w1.Text) +
    (TextBox_SI2.Text) * (TextBox_w2.Text) + (TextBox_SI3.Text) *
    (TextBox_w3.Text) + (TextBox_SI4.Text) * (TextBox_w4.Text)) * (0.8) +
    ((TextBox_SI5.Text) * (TextBox_w5.Text)) * (0.2)
    TextBox_state.Text = ((TextBox_SI6.Text) * (TextBox_w6.Text) +
    (TextBox_SI7.Text) * (TextBox_w7.Text)) * (0.25) + ((TextBox_SI8.Text)
    * (TextBox_w8.Text) + (TextBox_SI9.Text) * (TextBox_w9.Text) +
    (TextBox_SI10.Text) * (TextBox_w10.Text)) * (0.375) +
    ((TextBox_SI11.Text) * (TextBox_w11.Text) + (TextBox_SI12.Text) *
    (TextBox_w12.Text) + (TextBox_SI13.Text) * (TextBox_w13.Text)) *
    (0.375)
    TextBox_res.Text = ((TextBox_SI14.Text) * (TextBox_w14.Text) +
    (TextBox_SI15.Text) * (TextBox_w15.Text) + (TextBox_SI16.Text) *
    (TextBox_w16.Text) + (TextBox_SI17.Text) * (TextBox_w17.Text) +
    (TextBox_SI18.Text) * (TextBox_w18.Text) + (TextBox_SI19.Text) *
    (TextBox_w19.Text)) * (0.67) + ((TextBox_SI20.Text) *
    (TextBox_w20.Text) + (TextBox_SI21.Text) * (TextBox_w21.Text) +
    (TextBox_SI22.Text) * (TextBox_w22.Text)) * (0.33)

```

EndSub

```

PrivateSub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    Err_WDPI.Show()

```

EndSub

```

PrivateSub Button_Des_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button_Des.Click
    Process.Start("C:\Users\Hp\Documents\Visual Studio
    2008\Projects\WDPI_Table.pdf")
    Dim FilePath AsString = "C:\Test.pdf"
    IfMy.Computer.FileSystem.FileExists(FilePath) Then
    Try
        Process.Start("acrobat", FilePath)
    Catch ex As Exception
    Try
        Process.Start("AcroRd32", FilePath)
    Catch ex2 As Exception
    Try
        Process.Start(FilePath)
    Catch ex3 As Exception
        MsgBox("Instal Acrobat Reader")
    EndTry
    EndTry

```

```

EndTry
Else
'MsgBox("File not found.")
EndIf

EndSub

PrivateSub Button_WDPIResult_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button_WDPIResult.Click
Dim pre, st, re AsString
    pre = ""
    st = ""
    re = ""
'//////////////////////////////////////PRESSURE//////////////////////////////////////
If N_UR < 7 Then
    pre = String.Concat(pre, "1")
EndIf
If N_WW < 7 Then
    pre = pre + " " + "2"
EndIf
If N_FW < 8 Then
    pre = pre + " " + "3"
EndIf
If N_PRV < 8 Then
    pre = pre + " " + "4"
EndIf
If N_EP < 7 Then
    pre = pre + " " + "5"
EndIf
    Label_pressure.Text = pre
    Label_pressure.Visible = True

'//////////////////////////////////////STATE//////////////////////////////////////

If N_SWQ < 7 Then
    st = String.Concat(st, "6")
'MsgBox(st)
EndIf
If N_GWQ < 7 Then
    st = st + " " + "7"
EndIf
If N_AD < 8 Then
    st = st + " " + "8"
EndIf
If N_REL < 8 Then
    st = st + " " + "9"
EndIf
If N_CON < 9 Then
    st = String.Concat(st, " ", "10")
EndIf
If N_WSC < 7 Then
    st = st + " " + "11"
EndIf
If N_WWC < 7 Then
    st = st + " " + "12"
EndIf
If N_WSW < 6 Then
    st = st + " " + "13"
EndIf
    Label_state.Text = st
    Label_state.Visible = True

```



```

'////////////////////////////////////RESPONSE////////////////////////////////////

If N_TWW < 9 Then
    re = String.Concat(re, "14")
EndIf
If N_RSC < 2 Then
    re = re + " " + "15"
EndIf
If N_RP < 5 Then
    re = re + " " + "16"
EndIf
If N_EE < 6 Then
    re = re + " " + "17"
EndIf
If N_RR < 6 Then
    re = re + " " + "18"
EndIf
If N_GRP < 3 Then
    re = re + " " + "19"
EndIf
If N_MAP < 5 Then
    re = re + " " + "20"
EndIf
If N_PP < 5 Then
    re = re + " " + "21"
EndIf
If N_PA < 5 Then
    re = re + " " + "22"
EndIf
    Label_response.Text = re
    Label_response.Visible = True
EndSub

PrivateSub GroupBox4_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GroupBox4.Enter

EndSub

PrivateSub WDPI_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

EndSub

PrivateSub Label75_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label75.Click

EndSub

PrivateSub Label5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label5.Click

EndSub
EndClass

```

Scenario Analyses for Water Supply Sustainability (WSS)

S_Gen.vb

```
PublicClass S_Gen
Dim TWW, RS, RTH, IWE, I_UWB As Int64
Dim P_TWW, P_RS, P_RTH, P_IWE, UWB As Int64
PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

EndSub

PrivateSub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs)

EndSub

PrivateSub S_Gen_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
For co As Integer = 0 To 100 Step 5
    S1.Items.Add(co)
Next
For co1 As Integer = 0 To 100 Step 5
    S2.Items.Add(co1)
Next
For co2 As Integer = 0 To 100 Step 5
    S3.Items.Add(co2)
Next
For co3 As Integer = 0 To 100 Step 5
    S4.Items.Add(co3)
Next

    N_City.Visible = True
    N_City.Text = BasicData.City.Text
    N_.Visible = True
    N_.Text = BasicData.City.Text
EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
Dim P_UWB As Int64
    I_UWB = (S1.Text * P_TWW + S2.Text * P_RS + S3.Text * P_RTH +
S4.Text * P_IWE) * 0.01
    P_UWB = Math.Abs(UWB)
    TextBox_Imp.Text = Math.Round(((I_UWB) / (P_UWB)) * 100)
    TextBox_S.Text = Math.Round(I_UWB)

EndSub

PrivateSub ComboBox2_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
S3.SelectedIndexChanged

EndSub

PrivateSub Button2_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    TWW = S1.Text * 0.01 * P_TWW
```

```

        RS = S2.Text * 0.01 * P_RS
        RTH = S3.Text * 0.01 * P_RTH
        IWE = S4.Text * 0.01 * P_IWE
Me.Chart1.Series("Reclaimed Water").Points.AddXY("Reclaimed Water",
TWW)
Me.Chart1.Series("Surface Runoff Storage").Points.AddXY("Surface
Runoff Storage", RS)
Me.Chart1.Series("Roof-Top Harvesting").Points.AddXY("Roof-Top
Harvesting", RTH)
Me.Chart1.Series("Illegal GW Extraction").Points.AddXY("Illegal GW
Extraction", IWE)

EndSub

PrivateSub CheckBox_VNS_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox_VNS.CheckedChanged
If CheckBox_VNS.Checked = TrueThen
    P_TWW = 109500
    P_RS = 5195
    P_RTH = 2080
    P_IWE = 35770
    UWB = -36500
    TextBox_AWC.Text = 136875
    TextBox_AUWB.Text = 36500
    TextBox_IWP.Text = 35770
    TextBox_WW.Text = 109500
    TextBox_RTH.Text = 2080
    TextBox_RF.Text = 2597
EndIf
EndSub

PrivateSub Label14_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label14.Click

EndSub

PrivateSub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TextBox_RTH.TextChanged

EndSub

PrivateSub GroupBox3_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs)

EndSub

PrivateSub Label32_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label32.Click

EndSub

PrivateSub Button_ES_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button_ES.Click
    TextBox_C.Text = 50
    TextBox_G.Text = 10
    TextBox_R.Text = 50
    TextBox_RS.Text = 20
    TextBox_RW.Text = 0
EndSub

```

```

PrivateSub Label12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label12.Click
EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Future_WDPI.Show()
EndSub
EndClass

```

Interface Developed to Display the Final Map of the City Area

Finalmap.vb

```

Imports ESRI.ArcGIS.Geodatabase
Imports ESRI.ArcGIS.SystemUI
Imports ESRI.ArcGIS.Carto
Imports ESRI.ArcGIS.Controls
Imports ESRI.ArcGIS.Display
Imports ESRI.ArcGIS.Geometry
Imports ESRI.ArcGIS.esriSystem
Imports Microsoft.VisualBasic
Imports System
Imports System.Drawing
Imports System.Collections
Imports System.ComponentModel
Imports System.Windows.Forms
Imports ESRI.ArcGIS.Geoprocessor
Imports ESRI.ArcGIS.DataManagementTools
'Imports ESRI.ArcGIS.Framework
'Imports ESRI.ArcGIS.ArcMapUI
'Imports ESRI.ArcGIS.Geometry
Imports System.IO

PublicClass finalmap
Private m_mapControl As IMapControl3 = Nothing
Private m_mapDocumentName AsString = String.Empty

#Region"Get BasicMap from Map"

'''<summary>Get BasicMap from Map.</summary>
'''
'''<param name="map">An IMap interface.</param>
'''
'''<returns>An IBasicMap interface.</returns>
'''
'''<remarks>IBasicMap is a subset of IMap that provides support for
ArcScene and ArcGlobe. The Map (2D), Scene (3D), and Globe (3D)
coclasses implement this interface. Components used by ArcMap,
ArcScene, and ArcGlobe, (such as the IdentifyDialog) utilize IBasicMap
rather than IMap.</remarks>
PublicFunction GetBasicMapFromMap(ByVal map As IMap) As IBasicMap

Dim basicMap As IBasicMap = TryCast(map, IBasicMap) ' Dynamic Cast

Return basicMap

EndFunction
#EndRegion

```

```

#Region"Perform Spatial Query"

'''<summary>Creates a spatial query which performs a spatial search
for features in the supplied feature class and has the option to also
apply an attribute query via a where clause.</summary>
'''
'''<param name="featureClass">An
ESRI.ArcGIS.Geodatabase.IFeatureClass</param>
'''<param name="searchGeometry">An ESRI.ArcGIS.Geometry.IGeometry
(Only high-level geometries can be used)</param>
'''<param name="spatialRelation">An
ESRI.ArcGIS.Geodatabase.esriSpatialRelEnum (e.g.,
esriSpatialRelIntersects)</param>
'''<param name="whereClause">A System.String, (e.g., "city_name =
'Redlands'").</param>
'''
'''<returns>An IFeatureCursor holding the results of the query will be
returned.</returns>
'''
'''<remarks>Call the SpatialQuery method by passing in a reference to
the Feature Class, a Geometry used for the search and the spatial
operation to be preformed. An exmample of a spatial opertaion would be
intersects (e.g., esriSpatialRelEnum.esriSpatialRelContains). If you
would like to return everything found by the spatial operation use ""
for the where clause. Optionally a whereclause (e.g. "income > 1000")
maybe applied if desired. The SQL syntax used to specify the where
clause is the same as that of the underlying database holding the
data.</remarks>
PublicFunction PerformSpatialQuery(ByVal featureClass As
IFeatureClass, ByVal searchGeometry As IGeometry, ByVal
spatialRelation As esriSpatialRelEnum, ByVal whereClause As
System.String) As IFeatureCursor

' create a spatial query filter
Dim spatialFilter As ISpatialFilter = New SpatialFilterClass()

' specify the geometry to query with
    spatialFilter.Geometry = searchGeometry

' specify what the geometry field is called one Feature Class that we
will be querying against
Dim nameOfShapeField As System.String = featureClass.name
    spatialFilter.GeometryField = Name

' specify the type of spatial operation to use
    spatialFilter.SpatialRel =
esriSpatialRelEnum.esriSpatialRelContains

' create the where statement
    spatialFilter.WhereClause = whereClause

' perform the query and use a cursor to hold the results
Dim queryFilter As IQueryFilter = New QueryFilterClass()
    queryFilter = CType(spatialFilter, IQueryFilter)
Dim featureCursor As IFeatureCursor = featureClass.Search(queryFilter,
False)

Return featureCursor

EndFunction

```

```
#EndRegion
```

```
PrivateSub ContextMenuStrip1_Opening(ByVal sender As System.Object,  
ByVal e As System.ComponentModel.CancelEventArgs)
```

```
EndSub
```

```
PrivateSub AddMapCommandsToToolbarControl(ByVal toolbarControl As  
IToolbarControl)
```

```
toolbarControl.AddToolBarDef("esriControls.ControlsMapNavigationToolba  
r", -1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)  
    toolbarControl.AddItem("esriControls.ControlsMapIdentifyTool",  
-1, -1, True, 0, esriCommandStyles.esriCommandStyleIconOnly)  
    toolbarControl.AddItem("esriControls.ControlsMapFindCommand",  
-1, -1, True, 0, esriCommandStyles.esriCommandStyleIconOnly)
```

```
toolbarControl.AddItem("esriControls.ControlsMapHyperlinkTool", -1, -  
1, True, 0, esriCommandStyles.esriCommandStyleIconOnly)  
    toolbarControl.AddItem("esriControls.ControlsMapMeasureTool",  
-1, -1, True, 0, esriCommandStyles.esriCommandStyleIconOnly)
```

```
toolbarControl.AddItem("esriControls.ControlsLayerListToolControl", -  
1, -1, True, 0, esriCommandStyles.esriCommandStyleIconOnly)
```

```
toolbarControl.AddItem("esriControls.ControlsLayerTransparencyCommand"  
, -1, -1, True, 0, esriCommandStyles.esriCommandStyleIconOnly)  
    toolbarControl.AddItem("esriControls.ControlsMapSwipeTool", -  
1, -1, True, 0, esriCommandStyles.esriCommandStyleIconOnly)
```

```
EndSub
```

```
PrivateSub finalmap_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
'Load a pre-authored map document into the PageLayoutControl using  
relative paths.
```

```
'AxTOCControl1.LabelEdit = esriTOCControlEdit.esriTOCControlManual
```

```
'Dim sFileName As String =
```

```
"..\..\..\..\data\GulfOfStLawrence\Gulf_of_St._Lawrence.mxd"
```

```
'If AxPageLayoutControl1.CheckMxFile(sFileName) Then
```

```
'AxPageLayoutControl1.LoadMxFile(sFileName)
```

```
'End If
```

```
'AxTOCControl1.SetBuddyControl(IMapControl3)
```

```
' AxToolbarControl1.SetBuddyControl(AxPageLayoutControl1)
```

```
' AxToolbarControl2.SetBuddyControl(AxPageLayoutControl1)
```

```
'AxToolbarControl3.SetBuddyControl(AxPageLayoutControl1)
```

```
'get the MapControl
```

```
Dim m_mapControl As IMapControl3 = Nothing
```

```
Dim m_mapDocumentName AsString = String.Empty
```

```
    m_mapControl = CType(AxMapControl1.Object, IMapControl3)
```

```
'Load a pre-authored map document into the PageLayoutControl using  
relative paths.
```

```
'Add generic commands.
```

```
AxToolbarControl1.AddItem("esriControls.ControlsOpenDocCommand", -1, -  
1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)
```

```
AxToolbarControl1.AddItem("esriControls.ControlsAddDataCommand", -1, -  
1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)
```

```
'Add page layout navigation commands.
```

```

AxToolBarControl1.AddItem("esriControls.ControlsPageZoomInTool", -1, -
1, True, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolBarControl1.AddItem("esriControls.ControlsPageZoomOutTool", -1,
-1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)
    AxToolBarControl1.AddItem("esriControls.ControlsPagePanTool",
-1, -1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolBarControl1.AddItem("esriControls.ControlsPageZoomWholePageComma
nd", -1, -1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)
'Add map navigation commands.

AxToolBarControl1.AddItem("esriControls.ControlsMapZoomInTool", -1, -
1, True, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolBarControl1.AddItem("esriControls.ControlsMapZoomOutTool", -1, -
1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)
    AxToolBarControl1.AddItem("esriControls.ControlsMapPanTool", -
1, -1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolBarControl1.AddItem("esriControls.ControlsMapFullExtentCommand",
-1, -1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolBarControl1.AddItem("esriControls.ControlsMapZoomToLastExtentBac
kCommand", -1, -1, False, 0,
esriCommandStyles.esriCommandStyleIconOnly)

AxToolBarControl1.AddItem("esriControls.ControlsMapZoomToLastExtentFor
wardCommand", -1, -1, False, 0,
esriCommandStyles.esriCommandStyleIconOnly)
'Add map inquiry commands.

AxToolBarControl1.AddItem("esriControls.ControlsMapIdentifyTool", -1,
-1, True, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolBarControl1.AddItem("esriControls.ControlsMapFindCommand", -1, -
1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolBarControl1.AddItem("esriControls.ControlsMapMeasureTool", -1, -
1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)

'AxTOCControl1.SetBuddyControl(IMapControl2)
'AxToolBarControl1.SetBuddyControl(AxPageLayoutControl1)

'Set label editing to manual.
    AxTOCControl1.LabelEdit =
esriTOCControlEdit.esriTOCControlManual

'Add custom AddDateTool.
'AxToolBarControl1.AddItem("Commands.AddDateTool", -1, -1, False, 0,
esriCommandStyles.esriCommandStyleIconAndText)

'get the MapControl
    m_mapControl = CType(AxMapControl1.Object, IMapControl3)

'disable the Save menu (since there is no document yet)

' menuSaveDoc.Enabled = False

```

```

EndSub
PrivateSub TabPage4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TabPage4.Click

EndSub

PrivateSub AxToolBarControl2_OnMouseDown(ByVal sender As
System.Object, ByVal e As
ESRI.ArcGIS.Controls.IToolbarControlEvents_OnMouseDownEvent)

EndSub

PrivateSub MenuStrip1_ItemClicked(ByVal sender As System.Object, ByVal
e As System.Windows.Forms.ToolStripItemClickedEventArgs) Handles
MenuStrip1.ItemClicked

EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
'GIS_Query.Show()
' Intialize the Geoprocessor
Dim GP As Geoprocessor = New Geoprocessor

' Set the OverwriteOutput setting to True
    GP.OverwriteOutput = True

' Intialize the MakeFeatureLayer tool
Dim makefeaturelayer As MakeFeatureLayer = New MakeFeatureLayer()
    makefeaturelayer.in_features =
"E:\Prachi_project\Sewage_polygons"|"C:\gp\nfld.gdb\wells"
    makefeaturelayer.out_layer = "Wells_Lyr"
    RunTool(GP, makefeaturelayer, Nothing)

    makefeaturelayer.in_features =
"E:\Prachi_project\Pumping_station"|"C:\gp\nfld.gdb\bedrock"
    makefeaturelayer.out_layer = "bedrock_Lyr"
    RunTool(GP, makefeaturelayer, Nothing)

'////////////////////////////////////
////
' STEP 2: Execute SelectLayerByLocation using the feature
layers to select all wells that intersect theb bedrock geololgy.

'////////////////////////////////////
////

' Intialize the SelectLayerByLocation tool
Dim SelectByLocation As SelectLayerByLocation = New
SelectLayerByLocation()
'SelectByLocation.
    SelectByLocation.in_layer = "Wells_Lyr"
    SelectByLocation.select_features = "bedrock_Lyr"
    SelectByLocation.overlap_type = "INTERSECT"
    RunTool(GP, SelectByLocation, Nothing)
' Intialize the SelectLayerByAttribute tool
Dim SelectByAttribute As SelectLayerByAttribute = New
SelectLayerByAttribute()

```



```

        SelectByAttribute.in_layer_or_view = "Wells_Lyr"
        SelectByAttribute.selection_type = "NEW_SELECTION"
        SelectByAttribute.where_clause = "id =1"
        RunTool(GP, SelectByAttribute, Nothing)

'////////////////////////////////////
////
' STEP 4: Execute CopyFeatures tool to create a new feature
class of wells with well yield > 150 L/min.

'////////////////////////////////////
////

' Intialize the CopyFeatures tool
Dim copy_features As ESRI.ArcGIS.DataManagementTools.CopyFeatures =
New ESRI.ArcGIS.DataManagementTools.CopyFeatures ()

        copy_features.in_features = "Wells_Lyr"
        copy_features.out_feature_class =
"E:\Prachi_project\new_layer" & "C:\gp\nfld.gdb\wells_150"

' Set the output Coordinate System environment
'GP.SetEnvironmentValue("outputCoordinateSystem", "C:\Program
Files\ArcGIS\Coordinate Systems\Projected Coordinate Systems\UTM\Nad
1983\NAD 1983 UTM Zone 21N.prj")

        RunTool(GP, copy_features, Nothing)

EndSub
PrivateSharedSub RunTool(ByVal geoprocessor As Geoprocessor, ByVal
process As IGPPProcess, ByVal TC As ITrackCancel)

' Set the overwrite output option to true
        geoprocessor.OverwriteOutput = True

Try

        geoprocessor.Execute(process, Nothing)
        ReturnMessages(geoprocessor)

Catch err As Exception
        Console.WriteLine(err.Message)
        ReturnMessages(geoprocessor)

EndTry
EndSub
' Function for returning the tool messages.
PrivateSharedSub ReturnMessages(ByVal gp As Geoprocessor)
' Print out the messages from tool executions
Dim Count AsInteger
If gp.MessageCount > 0 Then
For Count = 0 To gp.MessageCount - 1
        Console.WriteLine(gp.GetMessage(Count))

Next
EndIf
EndSub

PrivateSub AxToolbarControl1_OnMouseDown(ByVal sender As
System.Object, ByVal e As
ESRI.ArcGIS.Controls.IToolbarControlEvents_OnMouseDownEvent) Handles
AxToolbarControl1.OnMouseDown

```

```

EndSub
PrivateSub AxTOCControl1_OnEndLabelEdit(ByVal sender As System.Object,
ByVal e As ESRI.ArcGIS.Controls.ITOCControlEvents_OnEndLabelEditEvent)
Handles AxTOCControl1.OnEndLabelEdit
'If the new label is an empty string, prevent the edit.
If e.newLabel.Trim = ""Then e.canEdit = False
EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click

' 'Find the feature for Osaka and get its geometry.
Dim QueryFilter As IQueryFilter = New QueryFilterClass()
'Dim IQueryFilter As QueryFilter = New QueryFilterClass()
Dim ID AsInteger
    QueryFilter.WhereClause = [ID] > 2

'Try to select the feature(s)
Dim FeatureLayer As IFeatureLayer = Me.AxMapControl1.get_Layer(0)
'Dim IFeatureLayer As FeatureLayer = Me.AxMapControl1.get_Layer(0)
Dim featureSelection As IFeatureSelection = FeatureLayer
    featureSelection.SelectFeatures(QueryFilter,
esriSelectionResultEnum.esriSelectionResultNew, False)

'Execute the built in zoom to selected command
Dim zoomToSelected As ControlsZoomToSelectedCommand = New
ControlsZoomToSelectedCommandClass()
    zoomToSelected.OnCreate(Me.AxMapControl1.Object)
    zoomToSelected.OnClick()

EndSub

PrivateSub AxMapControl1_OnMouseDown(ByVal sender As System.Object,
ByVal e As ESRI.ArcGIS.Controls.IMapControlEvents2_OnMouseDownEvent)
Handles AxMapControl1.OnMouseDown

EndSub

PrivateSub statusBarXY_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles statusBarXY.Click

EndSub

PrivateSub axMapControl1_OnMouseMove(ByVal sender AsObject, ByVal e As
IMapControlEvents2_OnMouseMoveEvent) Handles AxMapControl1.OnMouseMove
    statusBarXY.Text = String.Format("{0}, {1} {2}",
e.mapX.ToString("#####.##"), e.mapY.ToString("#####.##"),
axMapControl1.MapUnits.ToString().Substring(4))
EndSub
EndClass

```