using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace AutomatedAssetMIS

{

public partial class FrmGeneralAssetAnalysis : Form

{

private OleDbConnection cn;

//private OleDbConnection cn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Project 2016\AutomatedAssetMIS\AutomatedAssetMIS\dbAsset.accdb");

private OleDbCommand cmd;

private OleDbDataReader dr;

public FrmGeneralAssetAnalysis()

{

InitializeComponent();

cn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + Application.StartupPath + "\\dbAsset.accdb;Persist Security Info=True");

cmd = new OleDbCommand();

}

private void btnCancel\_Click(object sender, EventArgs e)

{

this.Hide();

}

private void FrmGeneralAssetAnalysis\_Load(object sender, EventArgs e)

{

cmd.Connection = cn;

listView1.View = View.Details;

listView1.GridLines = true;

listView1.FullRowSelect = true;

}

private void btnGenerate\_Click(object sender, EventArgs e)

{

try

{

listView1.Items.Clear();

if ((txtAssetLocation.Text == ""))

{

MessageBox.Show("Please specify the asset location before you proceed", "Asset MIS App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

else

{

{

genAssetSalesAnalysisReport(txtAssetLocation.Text);

}

}

}

catch (Exception Er)

{

MessageBox.Show(Er.Message, "Asset MIS App. ", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void genAssetSalesAnalysisReport(string AssetLocation)

{

string[] arrRpt = new string[5];

ListViewItem item;

//Add Column Header

listView1.Columns.Add("Asset Type", 150);

listView1.Columns.Add("Number of Sales", 150);

listView1.Columns.Add("Total Asset Cost", 150);

listView1.Columns.Add("Total Amount Paid", 150);

listView1.Columns.Add("Total Balance", 150);

try

{

cn.Open();

//cmd.CommandText = "select DISTINCT Building\_ID, Building\_Type, Building\_Name, Count(Building\_ID), Sum(Building\_Rentage\_Cost), Sum(Building\_Amount\_Paid) from tblBuildingSales WHERE (Building\_Location ='" + AssetLocation + "') ORDER BY Building\_ID, Building\_Type, Building\_Name";

//cmd.CommandText = "select Count(Building\_ID), Sum(Building\_Rentage\_Cost), Sum(Building\_Amount\_Paid), sum(Building\_Balance) from tblBuildingSales WHERE (Building\_Location ='" + AssetLocation + "') GROUP BY Building\_ID, Building\_Type, Building\_Name";

cmd.CommandText = "select Count(Building\_ID), Sum(Building\_Rentage\_Cost), Sum(Building\_Amount\_Paid), sum(Building\_Balance) from tblBuildingSales WHERE (Building\_Location ='" + AssetLocation + "') ";

dr = cmd.ExecuteReader();

while (dr.Read())

{

arrRpt[0] = "Building ";

arrRpt[1] = ((dr[0]).ToString()); //populate the building type

arrRpt[2] = ((dr[1]).ToString()); //populate the building name

arrRpt[3] = ((dr[2]).ToString()); //populate the number of distinct building id

arrRpt[4] = ((dr[3]).ToString()); //populate the Sum of distinct building Cost

//add item into list

item = new ListViewItem(arrRpt);

listView1.Items.Add(item);

}

dr.Close();

cn.Close();

//Generate report base on Land Sales

cn.Open();

cmd.CommandText = "select Count(Land\_Reg\_ID), Sum(Land\_Sale\_Cost), Sum(Land\_Amount\_Paid), sum(Land\_Balance) from tblLandSales WHERE (Land\_Location ='" + AssetLocation + "') ";

dr = cmd.ExecuteReader();

while (dr.Read())

{

arrRpt[0] = "Land ";

arrRpt[1] = ((dr[0]).ToString()); //populate the Land Numbers

arrRpt[2] = ((dr[1]).ToString()); //populate the Land Cost

arrRpt[3] = ((dr[2]).ToString()); //populate the Sum of Land Amount Paid

arrRpt[4] = ((dr[3]).ToString()); //populate the Sum of distinct Land Balance

//add item into list

item = new ListViewItem(arrRpt);

listView1.Items.Add(item);

}

dr.Close();

cn.Close();

//Generate report base on Furniture Sales

cn.Open();

cmd.CommandText = "select Count(Furniture\_Name), Sum(Furniture\_Cost), Sum(Furniture\_Amount\_Paid), Sum(Furniture\_Balance) from tblFurnitureSales WHERE (Furniture\_Location = '" + AssetLocation + "') ";

dr = cmd.ExecuteReader();

while (dr.Read())

{

arrRpt[0] = "Furniture ";

arrRpt[1] = ((dr[0]).ToString()); //populate the Furniture Name numbers

arrRpt[2] = ((dr[1]).ToString()); //populate the Furniture Cost

arrRpt[3] = ((dr[2]).ToString()); //populate the Furniture Amount Paid

arrRpt[4] = ((dr[3]).ToString()); //populate the Sum of Furniture Balance

//add item into list

item = new ListViewItem(arrRpt);

listView1.Items.Add(item);

}

dr.Close();

cn.Close();

//Generate report base on Vehicle Sales

cn.Open();

cmd.CommandText = "select Count(Vehicle\_Type), Sum(Vehicle\_Cost), Sum(Vehicle\_Amount\_Paid), Sum(Vehicle\_Balance) from tblVehicleSales WHERE (Vehicle\_Location = '" + AssetLocation + "')";

dr = cmd.ExecuteReader();

while (dr.Read())

{

arrRpt[0] = "Vehicle ";

arrRpt[1] = ((dr[0]).ToString()); //populate the Vehicle Type numbers

arrRpt[2] = ((dr[1]).ToString()); //populate the Vehicle Cost

arrRpt[3] = ((dr[2]).ToString()); //populate the Vehicle Amount Paid

arrRpt[4] = ((dr[3]).ToString()); //populate the Sum of Vehicle Balance

//add item into list

item = new ListViewItem(arrRpt);

listView1.Items.Add(item);

}

dr.Close();

cn.Close();

}

catch (Exception Er)

{

dr.Close();

cn.Close();

MessageBox.Show(Er.Message, "Asset MIS App. ", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace AutomatedAssetMIS

{

public partial class FrmAsset : Form

{

private OleDbConnection cn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + Application.StartupPath + "\\dbAsset.accdb;Persist Security Info=True");

//private OleDbConnection cn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Project 2016\AutomatedAssetMIS\AutomatedAssetMIS\dbAsset.accdb");

private OleDbCommand cmd = new OleDbCommand();

private OleDbDataReader dr;

private bool nonNumberEntered = false;

public FrmAsset()

{

InitializeComponent();

}

private void btnBuildingCancel\_Click(object sender, EventArgs e)

{

this.Hide();

}

private void btnLandCancel\_Click(object sender, EventArgs e)

{

this.Hide();

}

private void btnFurnitureCancel\_Click(object sender, EventArgs e)

{

this.Hide();

}

private void btnVehicleCancel\_Click(object sender, EventArgs e)

{

this.Hide();

}

private void btnBuildingClear\_Click(object sender, EventArgs e)

{

txtBuildingID.Text = genBuildingId();

txtBuildingName.Clear();

cboBuildingType.ResetText();

txtRoomNumber.Clear();

txtBuildingOwnerName.Clear();

txtBuildingLocation.Clear();

txtBuildingOwnerPhone.Clear();

optSale.Checked = false;

optRentage.Checked = false;

txtRoomRentageCost.Clear();

txtRoomSaleCost.Clear();

chkWater.Checked = false;

chkToilet.Checked = false;

chkSittingRoom.Checked = false;

chkShop.Checked = false;

chkOthers.Checked = false;

chkKitchen.Checked = false;

chkGarden.Checked = false;

chkFurniture.Checked = false;

chkFence.Checked = false;

chkDinningRoom.Checked = false;

chkBedroom.Checked = false;

txtBuildingAgentName.Clear();

txtBuildingAgentAddress.Clear();

txtBuildingAgentPhone.Clear();

}

private void btnLandClear\_Click(object sender, EventArgs e)

{

txtLandRegID.Text = genLandRegId();

txtLandOwnerName.Clear();

txtLandOwnerAddress.Clear();

txtLandOwnerPhone.Clear();

txtLandLocation.Clear();

txtLandWidth.Clear();

txtLandLength.Clear();

optLandSale.Checked = false;

optLandRentage.Checked = false;

txtLandRentageCost.Clear();

txtLandSaleCost.Clear();

txtLandAgentName.Clear();

txtLandAgentAddress.Clear();

txtLandAgentPhone.Clear();

dtpLandChargeDate.ResetText();

}

private void btnFurnitureClear\_Click(object sender, EventArgs e)

{

cboFurnitureType.ResetText();

txtManufacturerName.Clear();

txtManufacturerAddress.Clear();

txtManufacturerPhone.Clear();

cboFurnitureName.ResetText();

txtFurnitureCost.Clear();

dtpManufactureDate.ResetText();

txtFurnitureAgentName.Clear();

txtFurnitureAgentAddress.Clear();

txtFurnitureAgentPhone.Clear();

dtpFurnitureChargeDate.ResetText();

}

private void btnVehicleClear\_Click(object sender, EventArgs e)

{

cboVehicleType.ResetText();

cboVehicleModel.ResetText();

cboVehicleBrand.ResetText();

optNewModel.Checked = false;

optOldModel.Checked = false;

txtVehicleColor.Clear();

cboVehicleEngineType.ResetText();

txtVehicleEngineNumber.Clear();

txtVehiclePlateNumber.Clear();

cboVehicleMode.ResetText();

cboFuelType.ResetText();

txtVehicleCost.Clear();

dtpDateManufactured.ResetText();

txtVehicleAgentName.Clear();

txtVehicleAgentAddress.Clear();

txtVehicleAgentPhone.Clear();

}

private void btnBuildingSave\_Click(object sender, EventArgs e)

{

try

{

if ((txtBuildingName.Text == "") || (cboBuildingType.Text == "") || (txtRoomNumber.Text == "") || (txtBuildingLocation.Text == "") || (txtBuildingOwnerName.Text == "") || (txtBuildingAgentName.Text == ""))

{

MessageBox.Show("Please fill all Building details before you proceed", "Asset MIS App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

else

{

string BuildingSaleRentage = "Sale", BuildingFacilities = "";

string BuildingID = txtBuildingID.Text;

string BuildingName = txtBuildingName.Text;

string BuildingType = cboBuildingType.Text;

int RoomNumber = Convert.ToInt32(txtRoomNumber.Text);

string BuildingOwnerName = txtBuildingOwnerName.Text;

string BuildingLocation = txtBuildingLocation.Text;

string BuildingOwnerPhone = txtBuildingOwnerPhone.Text;

string BuildingAgentName = txtBuildingAgentName.Text;

string BuildingAgentAddress = txtBuildingAgentAddress.Text;

string BuildingAgentPhone = txtBuildingAgentPhone.Text;

if (optSale.Checked) BuildingSaleRentage = "Sale";

if (optRentage.Checked) BuildingSaleRentage = "Rentage";

double RoomRentageCost = Convert.ToDouble(txtRoomRentageCost.Text);

double RoomSaleCost = Convert.ToDouble(txtRoomSaleCost.Text);

if (chkWater.Checked) BuildingFacilities = "Water, ";

if (chkToilet.Checked) BuildingFacilities = BuildingFacilities + "Toilet, ";

if (chkSittingRoom.Checked) BuildingFacilities = BuildingFacilities + "SittingRoom, ";

if (chkShop.Checked) BuildingFacilities = BuildingFacilities + "Shop, ";

if (chkOthers.Checked) BuildingFacilities = BuildingFacilities + "and Others. ";

if (chkKitchen.Checked) BuildingFacilities = BuildingFacilities + "Kitchen, ";

if (chkGarden.Checked) BuildingFacilities = BuildingFacilities + "Garden, ";

if (chkFurniture.Checked) BuildingFacilities = BuildingFacilities + "Furniture, ";

if (chkFence.Checked) BuildingFacilities = BuildingFacilities + "Fence, ";

if (chkDinningRoom.Checked) BuildingFacilities = BuildingFacilities + "DiningRoom, ";

if (chkBedroom.Checked) BuildingFacilities = BuildingFacilities + "Bedroom, ";

cn.Open();

cmd.CommandText = "insert into tblBuilding values('" + BuildingID + "','" + BuildingName + "', '" + BuildingType + "'," + RoomNumber + ",'" + BuildingOwnerName + "','" + BuildingLocation + "','" + BuildingOwnerPhone + "','" + BuildingAgentName + "','" + BuildingAgentAddress + "','" + BuildingAgentPhone + "','" + BuildingFacilities + "','" + BuildingSaleRentage + "'," + RoomRentageCost + "," + RoomSaleCost + ")";

cmd.ExecuteNonQuery();

MessageBox.Show("Building Asset Information successfully saved!!!", "Asset MIS App.", MessageBoxButtons.OK, MessageBoxIcon.Information);

cn.Close();

btnBuildingClear\_Click(sender, e);

}

}

catch (Exception er)

{

cn.Close();

MessageBox.Show(er.Message, "Asset MIS App.", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnLandSave\_Click(object sender, EventArgs e)

{

try

{

if ((txtLandOwnerName.Text == "") || (txtLandLocation.Text == "") || (txtLandWidth.Text == "") || (txtLandLength.Text == "") || (txtLandAgentName.Text == ""))

{

MessageBox.Show("Please fill all Land details before you proceed", "Asset MIS App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

else

{

string LandSaleRentage = "Sale";

string LandRegID = txtLandRegID.Text;

string LandOwnerName = txtLandOwnerName.Text;

string LandOwnerAddress = txtLandOwnerAddress.Text;

string LandOwnerPhone = txtLandOwnerPhone.Text;

string LandLocation = txtLandLocation.Text;

double LandWidth = Convert.ToDouble(txtLandWidth.Text);

double LandLength = Convert.ToDouble(txtLandLength.Text);

if(optLandSale.Checked) LandSaleRentage = "Sale";

if (optLandRentage.Checked) LandSaleRentage = "Lease(Rentage)";

double LandRentageCost = Convert.ToDouble(txtLandRentageCost.Text);

double LandSaleCost = Convert.ToDouble(txtLandSaleCost.Text);

string LandAgentName = txtLandAgentName.Text;

string LandAgentAddress = txtLandAgentAddress.Text;

string LandAgentPhone = txtLandAgentPhone.Text;

string LandChargeDate = dtpLandChargeDate.Text;

cn.Open();

cmd.CommandText = "insert into tblLand values('" + LandRegID + "','" + LandOwnerName + "','" + LandOwnerAddress + "','" + LandOwnerPhone + "','" + LandLocation + "'," + LandWidth + "," + LandLength + ",'" + LandSaleRentage + "'," + LandRentageCost + "," + LandSaleCost + ",'" + LandAgentName + "','" + LandAgentAddress + "','" + LandAgentPhone + "','" + LandChargeDate + "')";

cmd.ExecuteNonQuery();

MessageBox.Show("Land Asset Information successfully saved!!!", "Asset MIS App.", MessageBoxButtons.OK, MessageBoxIcon.Information);

cn.Close();

btnLandClear\_Click(sender, e);

}

}

catch (Exception er)

{

cn.Close();

MessageBox.Show(er.Message, "Asset MIS App.", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnFurnitureSave\_Click(object sender, EventArgs e)

{

try

{

if ((cboFurnitureType.Text == "") || (txtManufacturerName.Text == "") || (cboFurnitureName.Text == "") || (txtFurnitureCost.Text == "") || (txtFurnitureAgentName.Text == ""))

{

MessageBox.Show("Please fill all Furniture details before you proceed", "Asset MIS App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

else

{

string FurnitureType = cboFurnitureType.Text;

string ManufacturerName = txtManufacturerName.Text;

string ManufacturerAddress = txtManufacturerAddress.Text;

string ManufacturerPhone = txtManufacturerPhone.Text;

string FurnitureName = cboFurnitureName.Text;

double FurnitureCost = Convert.ToDouble(txtFurnitureCost.Text);

string ManufactureDate = dtpManufactureDate.Text;

string FurnitureAgentName = txtFurnitureAgentName.Text;

string FurnitureAgentAddress = txtFurnitureAgentAddress.Text;

string FurnitureAgentPhone = txtFurnitureAgentPhone.Text;

string FurnitureChargeDate = dtpFurnitureChargeDate.Text;

cn.Open();

cmd.CommandText = "insert into tblFurniture values('" + ManufacturerName + "','" + ManufacturerAddress + "','" + ManufacturerPhone + "','" + FurnitureType + "','" + FurnitureName + "'," + FurnitureCost + ",'" + ManufactureDate + "','" + FurnitureAgentName + "','" + FurnitureAgentAddress + "','" + FurnitureAgentPhone + "','" + FurnitureChargeDate + "')";

cmd.ExecuteNonQuery();

MessageBox.Show("Furniture Asset Information successfully saved!!!", "Asset MIS App.", MessageBoxButtons.OK, MessageBoxIcon.Information);

cn.Close();

}

}

catch (Exception er)

{

cn.Close();

MessageBox.Show(er.Message, "Asset MIS App.", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnVehicleSave\_Click(object sender, EventArgs e)

{

try

{

if ((cboVehicleType.Text == "") || (cboVehicleModel.Text == "") || (txtVehicleColor.Text == "") || (cboFuelType.Text == "") || (txtVehicleCost.Text == "")|| (txtVehicleAgentName.Text == ""))

{

MessageBox.Show("Please fill all Vehicle details before you proceed", "Asset MIS App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

else

{

string VehicleModelType = "New Model";

string VehicleType = cboVehicleType.Text;

string VehicleModel = cboVehicleModel.Text;

string VehicleBrand = cboVehicleBrand.Text;

if(optNewModel.Checked) VehicleModelType = "New Model";

if (optOldModel.Checked) VehicleModelType = "Old Model";

string VehicleColor = txtVehicleColor.Text;

string VehicleEngineType = cboVehicleEngineType.Text;

string VehicleEngineNumber = txtVehicleEngineNumber.Text;

string VehiclePlateNumber = txtVehiclePlateNumber.Text;

string VehicleMode = cboVehicleMode.Text;

string FuelType = cboFuelType.Text;

double VehicleCost = Convert.ToDouble(txtVehicleCost.Text);

string DateManufactured = dtpDateManufactured.Text;

string VehicleAgentName = txtVehicleAgentName.Text;

string VehicleAgentAddress = txtVehicleAgentAddress.Text;

string VehicleAgentPhone = txtVehicleAgentPhone.Text;

cn.Open();

cmd.CommandText = "insert into tblVehicle values('" + VehicleType + "','" + VehicleModel + "','" + VehicleBrand + "','" + VehicleModelType + "','" + VehicleColor + "','" + VehicleEngineType + "','" + VehicleEngineNumber + "','" + VehiclePlateNumber + "','" + VehicleMode + "','" + FuelType + "'," + VehicleCost + ",'" + DateManufactured + "','" + VehicleAgentName + "','" + VehicleAgentAddress + "','" + VehicleAgentPhone + "')";

cmd.ExecuteNonQuery();

MessageBox.Show("Vehicle Asset Information successfully saved!!!", "Asset MIS App.", MessageBoxButtons.OK, MessageBoxIcon.Information);

cn.Close();

}

}

catch (Exception er)

{

cn.Close();

MessageBox.Show(er.Message, "Asset MIS App.", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FrmAsset\_Load(object sender, EventArgs e)

{

cmd.Connection = cn;

txtBuildingID.Text = genBuildingId();

txtBuildingID.Enabled = false;

txtLandRegID.Text = genLandRegId();

txtLandRegID.Enabled = false;

}

private string genBuildingId()

{

string BuildingId = "Building";

try

{

int rowCount = 0;

cn.Open();

cmd.CommandText = "select count(\*) as RowCount from tblBuilding";

dr = cmd.ExecuteReader();

while (dr.Read())

rowCount = (int)(dr["RowCount"]);

rowCount++;

if (rowCount < 10)

{

BuildingId = BuildingId + "000" + rowCount;

}

else if (rowCount < 100)

{

BuildingId = BuildingId + "00" + rowCount;

}

else if (rowCount < 1000)

{

BuildingId = BuildingId + "0" + rowCount;

}

else if (rowCount < 10000)

{

BuildingId = BuildingId + rowCount;

}

dr.Close();

cn.Close();

}

catch (Exception Er)

{

dr.Close();

cn.Close();

MessageBox.Show(Er.Message, "Asset MIS App", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return BuildingId;

}

private string genLandRegId()

{

string LandRegId = "Land";

try

{

int rowCount = 0;

cn.Open();

cmd.CommandText = "select count(\*) as RowCount from tblLand";

dr = cmd.ExecuteReader();

while (dr.Read())

rowCount = (int)(dr["RowCount"]);

rowCount++;

if (rowCount < 10)

{

LandRegId = LandRegId + "000" + rowCount;

}

else if (rowCount < 100)

{

LandRegId = LandRegId + "00" + rowCount;

}

else if (rowCount < 1000)

{

LandRegId = LandRegId + "0" + rowCount;

}

else if (rowCount < 10000)

{

LandRegId = LandRegId + rowCount;

}

dr.Close();

cn.Close();

}

catch (Exception Er)

{

dr.Close();

cn.Close();

MessageBox.Show(Er.Message, "Asset MIS App", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return LandRegId;

}

private void txtRoomNumber\_KeyDown(object sender, KeyEventArgs e)

{

nonNumberEntered = false;

if (e.KeyCode < Keys.D0 || e.KeyCode > Keys.D9)

{

if (e.KeyCode < Keys.NumPad0 || e.KeyCode > Keys.NumPad9)

{

if ((e.KeyCode != Keys.Back))

nonNumberEntered = true;

}

}

if (Control.ModifierKeys == Keys.Shift)

{

nonNumberEntered = true;

}

}

private void txtRoomNumber\_KeyPress(object sender, KeyPressEventArgs e)

{

if (nonNumberEntered == true) e.Handled = true; //capture keytrapping i.e. keypress = 0 (null)

}

private void txtBuildingOwnerPhone\_KeyDown(object sender, KeyEventArgs e)

{

nonNumberEntered = false;

if (e.KeyCode < Keys.D0 || e.KeyCode > Keys.D9)

{

if (e.KeyCode < Keys.NumPad0 || e.KeyCode > Keys.NumPad9)

{

if ((e.KeyCode != Keys.Back))

nonNumberEntered = true;

}

}

if (Control.ModifierKeys == Keys.Shift)

{

nonNumberEntered = true;

}

}

private void txtBuildingOwnerPhone\_KeyPress(object sender, KeyPressEventArgs e)

{

if (nonNumberEntered == true) e.Handled = true; //capture keytrapping i.e. keypress = 0 (null)

}

private void txtBuildingAgentPhone\_KeyDown(object sender, KeyEventArgs e)

{

nonNumberEntered = false;

if (e.KeyCode < Keys.D0 || e.KeyCode > Keys.D9)

{

if (e.KeyCode < Keys.NumPad0 || e.KeyCode > Keys.NumPad9)

{

if ((e.KeyCode != Keys.Back))

nonNumberEntered = true;

}

}

if (Control.ModifierKeys == Keys.Shift)

{

nonNumberEntered = true;

}

}

private void txtBuildingAgentPhone\_KeyPress(object sender, KeyPressEventArgs e)

{

if (nonNumberEntered == true) e.Handled = true; //capture keytrapping i.e. keypress = 0 (null)

}

private void txtRoomRentageCost\_KeyPress(object sender, KeyPressEventArgs e)

{

if (nonNumberEntered == true) e.Handled = true; //capture keytrapping i.e. keypress = 0 (null)

}

private void txtRoomRentageCost\_KeyDown(object sender, KeyEventArgs e)

{

nonNumberEntered = false;

if (e.KeyCode < Keys.D0 || e.KeyCode > Keys.D9)

{

if (e.KeyCode < Keys.NumPad0 || e.KeyCode > Keys.NumPad9)

{

if ((e.KeyCode != Keys.Back))

nonNumberEntered = true;

}

}

if (Control.ModifierKeys == Keys.Shift)

{

nonNumberEntered = true;

}

}

private void txtRoomSaleCost\_KeyDown(object sender, KeyEventArgs e)

{

nonNumberEntered = false;

if (e.KeyCode < Keys.D0 || e.KeyCode > Keys.D9)

{

if (e.KeyCode < Keys.NumPad0 || e.KeyCode > Keys.NumPad9)

{

if ((e.KeyCode != Keys.Back))

nonNumberEntered = true;

}

}

if (Control.ModifierKeys == Keys.Shift)

{

nonNumberEntered = true;

}

}

private void txtRoomSaleCost\_KeyPress(object sender, KeyPressEventArgs e)

{

if (nonNumberEntered == true) e.Handled = true; //capture keytrapping i.e. keypress = 0 (null)

}

private void optSale\_CheckedChanged(object sender, EventArgs e)

{

if (optSale.Checked)

{

txtRoomSaleCost.Enabled = true;

txtRoomRentageCost.Enabled = false;

txtRoomRentageCost.Text = "0.00";

}

}

private void optRentage\_CheckedChanged(object sender, EventArgs e)

{

if (optRentage.Checked)

{

txtRoomSaleCost.Enabled = false;

txtRoomRentageCost.Enabled = true;

txtRoomSaleCost.Text = "0.00";

}

}

private void optLandSale\_CheckedChanged(object sender, EventArgs e)

{

if (optLandSale.Checked)

{

txtLandSaleCost.Enabled = true;

txtLandRentageCost.Enabled = false;

txtLandRentageCost.Text = "0.00";

}

}

private void optLandRentage\_CheckedChanged(object sender, EventArgs e)

{

if (optLandRentage.Checked)

{

txtLandSaleCost.Enabled = false;

txtLandRentageCost.Enabled = true;

txtLandSaleCost.Text = "0.00";

}

}

private void txtLandOwnerPhone\_KeyDown(object sender, KeyEventArgs e)

{

TextBox txt = new TextBox();

txt = (TextBox)sender;

nonNumberEntered = false;

if (e.KeyCode < Keys.D0 || e.KeyCode > Keys.D9)

{

if (e.KeyCode < Keys.NumPad0 || e.KeyCode > Keys.NumPad9)

{

if ((e.KeyCode != Keys.Back) && (e.KeyCode != Keys.OemPeriod))

nonNumberEntered = true;

}

}

if (Control.ModifierKeys == Keys.Shift)

{

nonNumberEntered = true;

}

}

private void txtLandOwnerPhone\_KeyPress(object sender, KeyPressEventArgs e)

{

TextBox txt = new TextBox();

txt = (TextBox)sender;

if (nonNumberEntered == true) e.Handled = true; //capture keytrapping i.e. keypress = 0 (null)

}

private void txtManufacturerPhone\_KeyDown(object sender, KeyEventArgs e)

{

TextBox txt = new TextBox();

txt = (TextBox)sender;

nonNumberEntered = false;

if (e.KeyCode < Keys.D0 || e.KeyCode > Keys.D9)

{

if (e.KeyCode < Keys.NumPad0 || e.KeyCode > Keys.NumPad9)

{

if ((e.KeyCode != Keys.Back) && (e.KeyCode != Keys.OemPeriod))

nonNumberEntered = true;

}

}

if (Control.ModifierKeys == Keys.Shift)

{

nonNumberEntered = true;

}

}

private void txtManufacturerPhone\_KeyPress(object sender, KeyPressEventArgs e)

{

TextBox txt = new TextBox();

txt = (TextBox)sender;

if (nonNumberEntered == true) e.Handled = true; //capture keytrapping i.e. keypress = 0 (null)

}

private void txtVehicleAgentPhone\_KeyDown(object sender, KeyEventArgs e)

{

TextBox txt = new TextBox();

txt = (TextBox)sender;

nonNumberEntered = false;

if (e.KeyCode < Keys.D0 || e.KeyCode > Keys.D9)

{

if (e.KeyCode < Keys.NumPad0 || e.KeyCode > Keys.NumPad9)

{

if ((e.KeyCode != Keys.Back) && (e.KeyCode != Keys.OemPeriod))

nonNumberEntered = true;

}

}

if (Control.ModifierKeys == Keys.Shift)

{

nonNumberEntered = true;

}

}

private void txtVehicleAgentPhone\_KeyPress(object sender, KeyPressEventArgs e)

{

TextBox txt = new TextBox();

txt = (TextBox)sender;

if (nonNumberEntered == true) e.Handled = true; //capture keytrapping i.e. keypress = 0 (null)

}

private void txtVehicleAgentPhone\_TextChanged(object sender, EventArgs e)

{

}

}

}