```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.Odbc;
using System.Net.Mail;
namespace Avatar_Solutions
{
    public partial class Processor : Form
    {
        public Processor()
        {
            InitializeComponent();
        }

        private AvatarConnection avConn;
        private OdbcConnection conn;
        private OdbcCommand cmd;
        private OdbcDataAdapter DA;
        DataTable dtbl;

        private List<Report> rptList;
        private List<ServiceRecord> srvRecord;

        int TotalCreated = 0;
        int TotalError = 0;
        int TotalAlreadyCreated = 0;

        public Processor(AvatarConnection con)
        {
            InitializeComponent();
            avConn = con;
            dtbl = new DataTable();
            conn = avConn.conn;

            srvRecord = new List<ServiceRecord>();
            rptList = new List<Report>();


            DA = new OdbcDataAdapter();

            dataGridView1.DataSource = dtbl;


        }

        private void GetData()
        {
            DateTime FromDate = DateTime.Parse(((int)comboBox1.SelectedItem).ToString()
            + "/01/" + comboBox2.Text);
            DateTime ThruDate = FromDate.AddMonths(1).AddDays(-1);

            string sendFrom = FromDate.ToString("yyyy-MM-dd");
            string sendThru = ThruDate.ToString("yyyy-MM-dd");
            //string sendFrom = From.Value.ToString("yyyy-MM-dd");
            //string sendThru = ThruDate.Value.ToString("yyyy-MM-dd");
            //MessageBox.Show("From: " + sendFrom + Environment.NewLine + "Thru: " +
            sendThru);
            DA.SelectCommand = avConn.SQLCaseMgmt(sendFrom, sendThru,
            avConn.Componets,avConn.ProgramExclude);
            DA.SelectCommand.Connection = conn;
            DA.Fill(dtbl);
        }

        private void ProcessData()
        {
            int ReqFTF = avConn.ReqFTF;
            int ReqTot = avConn.ReqTot;
            string billableCode = avConn.BillableCode;

            //This Code Generates Billable Charges
            int NonFTF = 0;
            int FTF = 0;
            int Total = 0;
            bool FoundBillable = false;
            bool started = true;
            string PreviousClient = "";


            int Episode = 0;
            int PrvEpisode = 0;
            string currentClient = "";
            string Service = "";
            int Tot = 0;
            string Program = "";
            string StaffID = "";
            DateTime DOS = DateTime.Now;

            string PrvService = "";
            //int Tot = 0;
            string PrvProgram = "";
            string PrvStaffID = "";
            DateTime PrvDOS = DateTime.Now;

            int i = 0;
            foreach(DataRow row in dtbl.Rows)
            {
                i++;

                currentClient = (string)row["PATID"];


                if (i == dtbl.Rows.Count || started)
                {
                    PreviousClient = currentClient;
                    started = false;

                    Service = (string)row["Service"];
                    Tot = (int)row["Tot"];
                    Program = (string)row["Program"];
                    StaffID = (string)row["STAFFID"];
                    Episode = (int)row["EPISODE"];
                    if (row["DOS"] != DBNull.Value)
                    {
                        DOS = Convert.ToDateTime(row["DOS"]);
                    }
                    else
                    {
                        throw new Exception("No DateTime");
                    }
                }

   /*            if (avConn.CurrentRuleTitle == "ACT")
                {
                    started = false;
                }*/

               if (i == dtbl.Rows.Count)
                {
                    currentClient = "";
                    FoundBillable = false;
                    FTF = 0;
                    Total = 0;
                    NonFTF = 0;

                    if (Service == billableCode)
                    {
                        FoundBillable = true;

                    }
                    else if (Service[Service.Length - 1] == 'F')
                    {
                        FTF += Tot;
                        Total += Tot;
                    }
                    else
                    {
                        NonFTF += Tot;
                        Total += Tot;
                    }

                    PrvDOS = DOS;
                    PrvProgram = Program;
                    PrvService = Service;
                    PrvStaffID = StaffID;
                    PrvEpisode = (int)row["EPISODE"];

                }


                if (PreviousClient != currentClient)
                {

                       //If override staff logic then change staff id HERE since always hit.
                       if (avConn.OverrideStaff)
                       {
                        PrvStaffID = avConn.OverrideStaffID;
                       }
                       //end staff override logic

                    if (!FoundBillable && FTF >= ReqFTF && Total >= ReqTot)
                    {
                        //GenerateCharge
                        rptList.Add(new Report(DOS, PreviousClient, FTF, Total, true, "Charge Created"));
                        avConn.ServList.Add(new ServiceRecord(PrvDOS, PrvEpisode, billableCode, PrvProgram, PrvStaffID, PreviousClient, avConn.currentServiceType));
                        TotalCreated += 1;
                    }
                    else if (FoundBillable)
                    {
                        //WasAlreadyCharges
                        rptList.Add(new Report(DOS, PreviousClient, FTF, Total, false, "Already Created"));
                        TotalAlreadyCreated += 1;
                    } else if (FTF >= ReqFTF && Total < ReqTot)
                    {
                        //Not Enough Services
                        rptList.Add(new Report(DOS, PreviousClient, FTF, Total, false, "Not Enough Services"));
                        TotalError += 1;
                    }
                    else if (FTF <= ReqFTF)
                    {
                        //Not Enough FTF
                        rptList.Add(new Report(DOS, PreviousClient, FTF, Total, false, "Not Enough FTF"));
                        TotalError += 1;
                    }
                    else
                    {
                        //Other Error
                        rptList.Add(new Report(DOS, PreviousClient, FTF, Total, false, "Other Err"));
                        TotalError += 1;
                    }



                    NonFTF = 0;
                    FTF = 0;
                    Total = 0;
                    FoundBillable = false;

                }

                //Always runs -- was miscounted
                Service = (string)row["Service"];
                Tot = (int)row["Tot"];
                Program = (string)row["Program"];
                StaffID = (string)row["STAFFID"];


                if (row["DOS"] != DBNull.Value)
                {
                    DOS = Convert.ToDateTime(row["DOS"]);
                }
                else
                {
                    throw new Exception("No DateTime");
                }
                if (Service == billableCode)
                    {
                        FoundBillable = true;

                    } else if (Service[Service.Length - 1] == 'F')
                    {
                        FTF += Tot;
                        Total += Tot;
                    } else
                    {
                        NonFTF += Tot;
                        Total += Tot;
                    }

                PrvDOS = DOS;
                PrvProgram = Program;
                PrvService = Service;
                PrvStaffID = StaffID;
                PreviousClient = currentClient;
                PrvEpisode = (int)row["EPISODE"];
            }


        }

        private void Test_Test_Load(object sender, EventArgs e)
        {
            this.Text += " Rule: " + avConn.CurrentRuleTitle + " Type: " + avConn.currentServiceType.ToString() + " Billable Code: " + avConn.BillableCode;
            this.comboBox1.DataSource = Enum.GetValues(typeof(Avatar_Solutions.month));

            comboBox2.Text = DateTime.Today.ToString("yyyy");
            comboBox1.SelectedIndex = DateTime.Today.Month - 2;

        }

        private void button1_Click(object sender, EventArgs e)
        {
            dtbl.Clear();
            this.GetData();
            comboBox1.SelectedIndex += 1;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.ProcessData();

            Report_Viewer frm = new Report_Viewer(rptList,TotalCreated,TotalAlreadyCreated,TotalError,avConn.CurrentRuleTitle,From.Value, ThruDate.Value);
            frm.ShowDialog();

           // SendMessage();

        }


    }
}
```