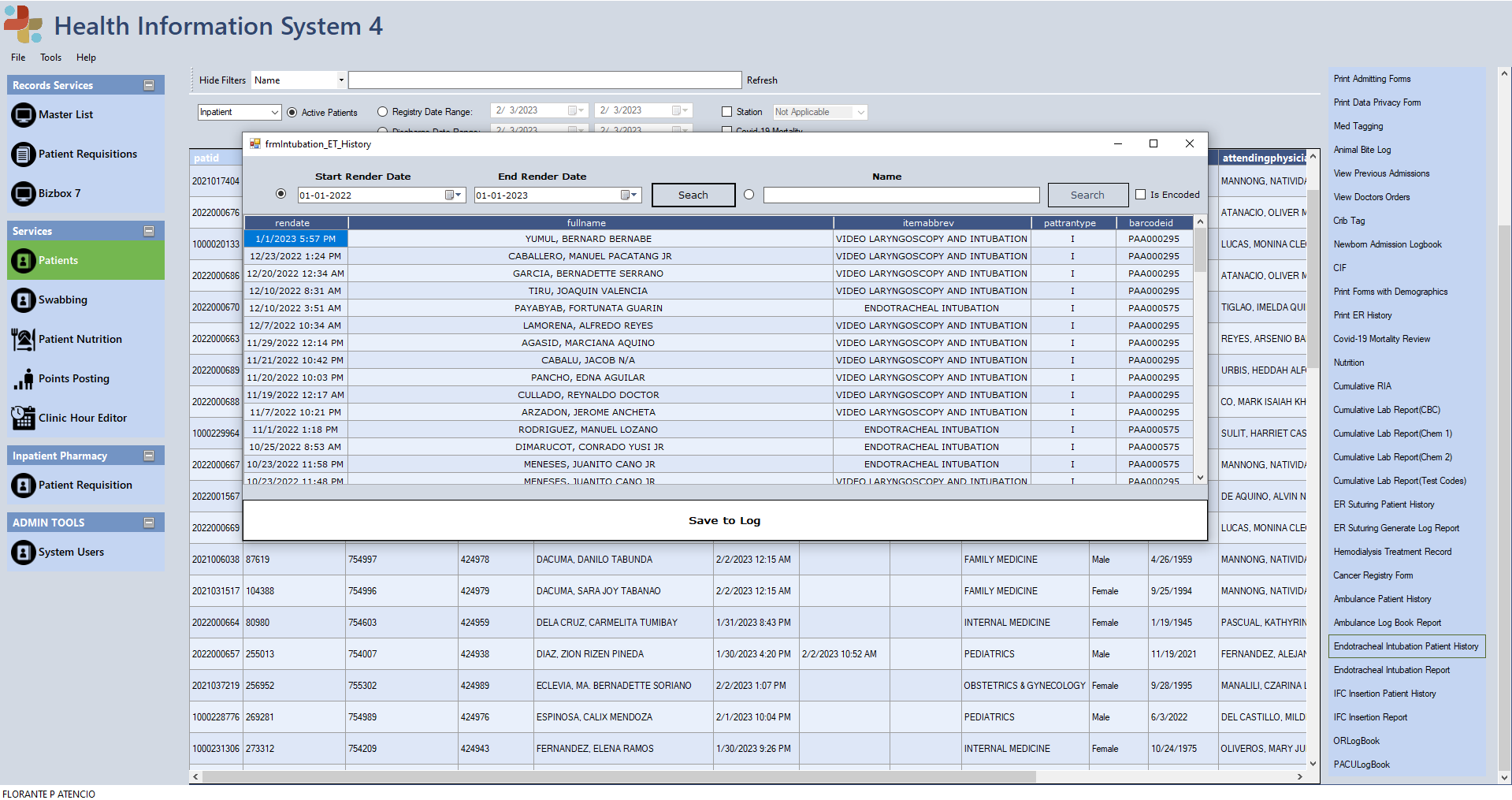
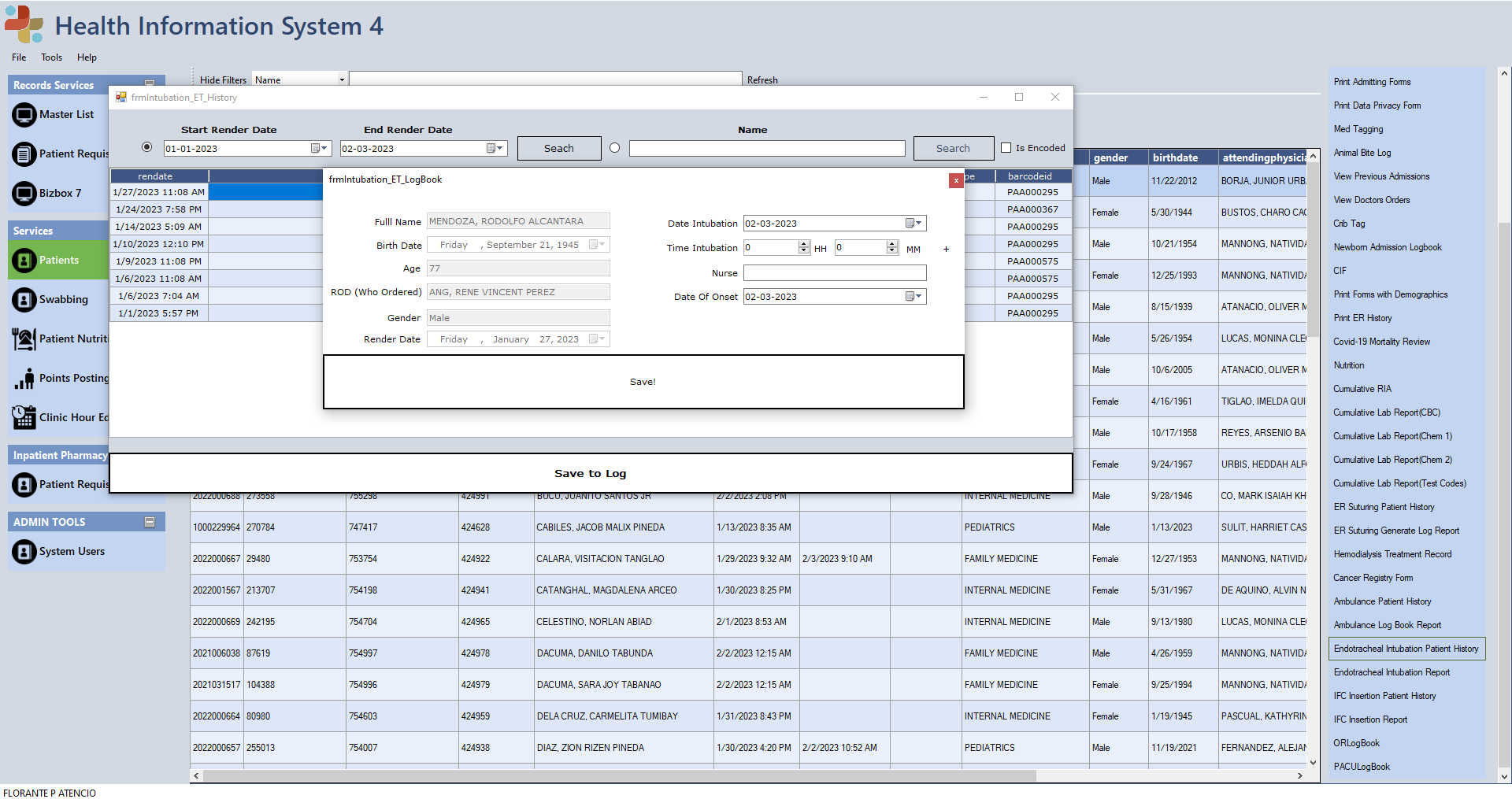
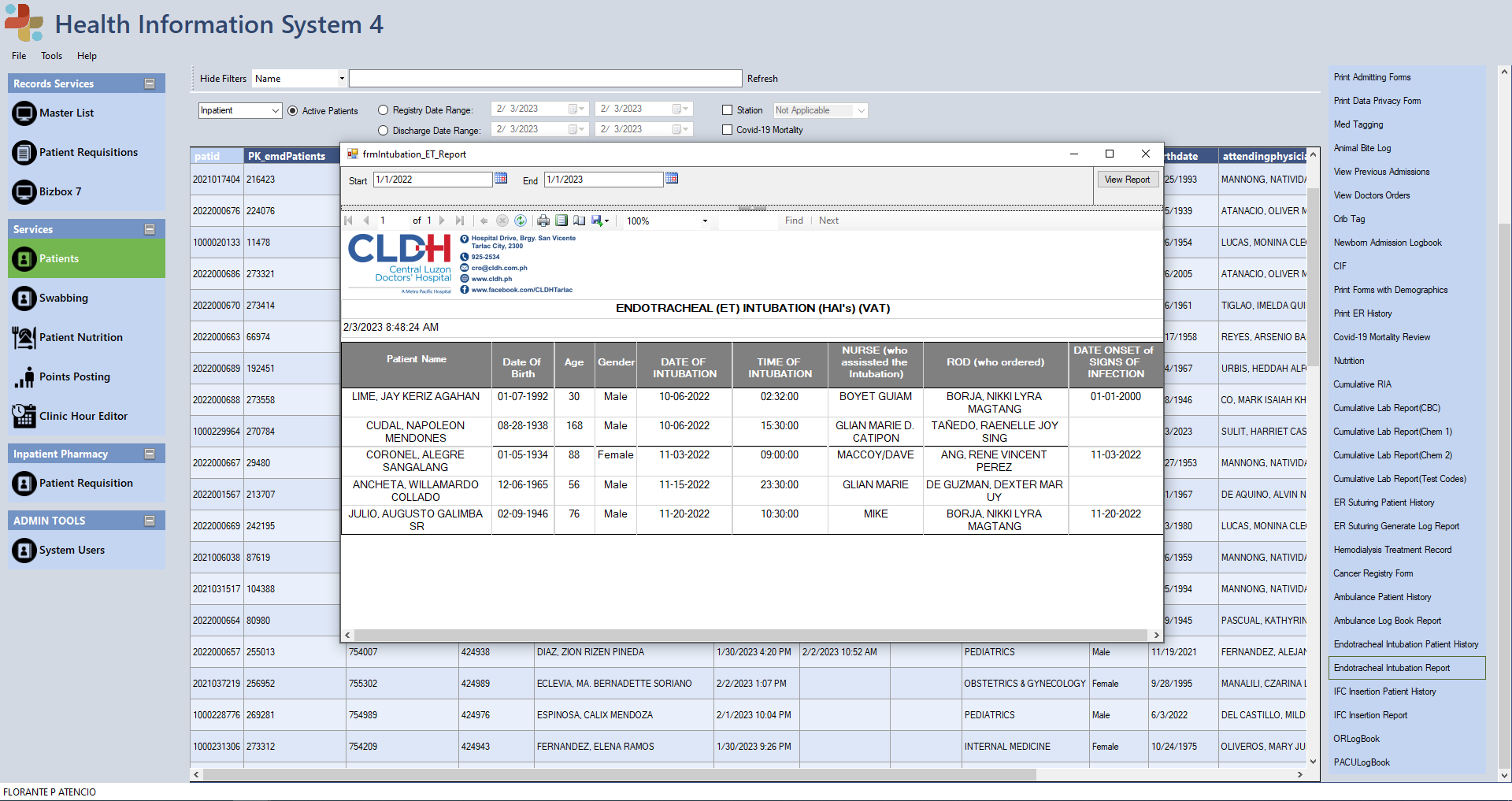
Intubation Endotracheal Module in HIS 4

In Services tag under Patients module Endotracheal intubation patient history click

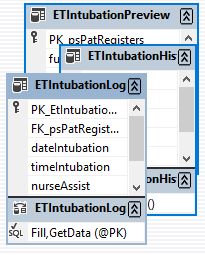
 Endotracheal intubation patient history Window will pop up

In order to save into log First select a patient inside the data table then click the save to log button. It will proceed to encoding date and time. It will required you to fill all the necessary field and click the save button. Then you can repeat the process.

To Access the Report Viewer from Patient module under Services Tab in action tab find the Endotracheal Intubation Report click. Report server 10.4.1.3/reportserver/JUN/frmEndotrachealReport.rdl

**ADO.NET (Table Adapter and DataBindsource)**

Accessing 10.4.1.3/sqlcldh02 (HealthInfoDB) and 10.4.1.3/sqlcldh01 (LIVE\_CLDH)



**Demo Graphics**

**ETIntubationPreview**

**ETIntubationHistory**

**CRUD**

**ETIntubationLog**

**Module Source Code**

**SQL SERVER:**

USE [HealthInfoDB]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ETIntubationLogBook] Script Date: 2/3/2023 8:20:17 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ETIntubationLogBook](

[PK\_EtIntubationLogBook] [int] IDENTITY(1,1) NOT NULL,

[FK\_psPatRegisters] [int] NULL,

[dateIntubation] [date] NULL,

[timeIntubation] [time](7) NULL,

[nurseAssist] [nvarchar](max) NULL,

[dateofOnset] [date] NULL,

[isEncode] [bit] NULL,

PRIMARY KEY CLUSTERED

(

[PK\_EtIntubationLogBook] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

**Visual Studio 2019 C#**

Frmintubation\_ET\_History:

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;

namespace HealthInformationSystem4.Forms

{

public partial class frmIntubation\_ET\_History : Form

{

string isEncoded = "(isEncode is null OR isEncode = 0)";

public frmIntubation\_ET\_History()

{

InitializeComponent();

}

private void frmIntubation\_ET\_History\_Load(object sender, EventArgs e)

{

radioButton1.Checked = true;

int MM = int.Parse(DateTime.Now.ToString("MM"));

int yyyy = int.Parse(DateTime.Now.ToString("yyyy"));

int dd = int.Parse(DateTime.Now.ToString("dd"));

dateTimePicker1.Value = new DateTime(yyyy, MM, 01);

dateTimePicker2.Value = new DateTime(yyyy, MM, dd);

this.eTIntubationHistoryBindingSource.Filter = "rendate >= '" + dateTimePicker1.Value.ToString("MM-dd-yyyy 00:00:00") + "' and rendate <= ' " + dateTimePicker2.Value.ToString("MM-dd-yyyy 23:59:59") + "' AND " + isEncoded;

// TODO: This line of code loads data into the 'cancerRegistry.ETIntubationHistory' table. You can move, or remove it, as needed.

this.eTIntubationHistoryTableAdapter.Fill(this.cancerRegistry.ETIntubationHistory);

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

eTIntubationHistoryBindingSource.Filter = null;

dateTimePicker1.Enabled = true;

dateTimePicker2.Enabled = true;

button1.Enabled = true;

textBox1.Enabled = false;

button3.Enabled = false;

}

private void radioButton3\_CheckedChanged(object sender, EventArgs e)

{

eTIntubationHistoryBindingSource.Filter = null;

dateTimePicker1.Enabled = false;

dateTimePicker2.Enabled = false;

button1.Enabled = false;

textBox1.Enabled = true;

button3.Enabled = true;

}

private void button2\_Click(object sender, EventArgs e)

{

CancerRegistryTableAdapters.ETIntubationLogBookTableAdapter ET = new CancerRegistryTableAdapters.ETIntubationLogBookTableAdapter();

CancerRegistry cancer = new CancerRegistry();

ET.Fill(cancer.ETIntubationLogBook, (int)psPatitemDataGridView.CurrentRow.Cells[1].Value);

if (cancer.ETIntubationLogBook.Rows.Count > 0)

{

frmIntubation\_ET\_LogBook log = new frmIntubation\_ET\_LogBook();

log.pk = psPatitemDataGridView.CurrentRow.Cells[1].Value.ToString();

log.ShowDialog();

}

else {

ET.Insert((int)psPatitemDataGridView.CurrentRow.Cells[1].Value, null, null, null, null,false);

frmIntubation\_ET\_LogBook log = new frmIntubation\_ET\_LogBook();

log.pk = psPatitemDataGridView.CurrentRow.Cells[1].Value.ToString();

log.ShowDialog();

}

this.eTIntubationHistoryTableAdapter.Fill(this.cancerRegistry.ETIntubationHistory);

}

private void button1\_Click(object sender, EventArgs e)

{

this.eTIntubationHistoryBindingSource.Filter = "rendate >= '" + dateTimePicker1.Value.ToString("MM-dd-yyyy 00:00:00") + "' AND rendate <= '" + dateTimePicker2.Value.ToString("MM-dd-yyyy 23:59:59") + "'AND " + isEncoded;

this.eTIntubationHistoryTableAdapter.Fill(this.cancerRegistry.ETIntubationHistory);

}

private void button3\_Click(object sender, EventArgs e)

{

this.eTIntubationHistoryBindingSource.Filter = "fullname like '%" + textBox1.Text + "%' AND " + isEncoded;

this.eTIntubationHistoryTableAdapter.Fill(this.cancerRegistry.ETIntubationHistory);

}

private void KeyPressDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter) {

button3.PerformClick();

}

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (!checkBox1.Checked)

{

eTIntubationHistoryBindingSource.Filter = "(isEncode is null OR isEncode = 0)";

isEncoded = "(isEncode is null OR isEncode = 0)";

}

else

{

eTIntubationHistoryBindingSource.Filter = "(IsEncode = 1)";

isEncoded = "(IsEncode = 1)";

}

}

}

}

frmIntubation\_ET\_Report:

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;

namespace HealthInformationSystem4.Forms

{

public partial class frmIntubation\_ET\_Report : Form

{

public frmIntubation\_ET\_Report()

{

InitializeComponent();

}

private void frmIntubation\_ET\_Report\_Load(object sender, EventArgs e)

{

this.reportViewer1.RefreshReport();

}

}

}

frmIntubation\_ET\_LogBook:

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;

namespace HealthInformationSystem4.Forms

{

public partial class frmIntubation\_ET\_LogBook : Form

{

public string pk { get; set; }

//public bool dateonset = false;

public bool dateCatherer = false;

public frmIntubation\_ET\_LogBook()

{

InitializeComponent();

}

private void frmIntubation\_ET\_LogBook\_Load(object sender, EventArgs e)

{

eTIntubationLogBookTableAdapter.Fill(this.cancerRegistry.ETIntubationLogBook, int.Parse(pk));

eTIntubationPreviewTableAdapter.Fill(this.cancerRegistry.ETIntubationPreview, int.Parse(pk));

if (timeIntubationTextBox1.Text != "") {

string[] cat = timeIntubationTextBox1.Text.Split(':');

numericUpDown1.Value = decimal.Parse(cat[0].ToString());

numericUpDown2.Value = decimal.Parse(cat[1].ToString());

}

dateCatherer = isEncodeCheckBox.Checked ? true : false;

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

textdate();

if (dateCatherer /\*&& dateonset\*/ && timetocheck())

{

isEncodeCheckBox.Checked = true;

this.eTIntubationLogBookBindingSource.EndEdit();

this.eTIntubationLogBookTableAdapter.Update(this.cancerRegistry.ETIntubationLogBook);

MessageBox.Show("Successful");

this.Close();

}

else

{

MessageBox.Show("Please Enter a Date And Fill up all the Form", "Date Required And Other Form");

}

}

catch (Exception)

{

MessageBox.Show("Please Try Again or call for an IT");

}

}

private void textdate()

{

timeIntubationTextBox1.Text = numericUpDown1.Value + ":" + numericUpDown2.Value + ":00";

}

private void DropDownDateCate(object sender, EventArgs e)

{

dateCatherer = true;

}

//private void Dp(object sender, EventArgs e)

//{

// dateonset = true;

//}

private void button2\_Click(object sender, EventArgs e)

{

numericUpDown1.Value = decimal.Parse(DateTime.Now.ToString("HH"));

numericUpDown2.Value = decimal.Parse(DateTime.Now.ToString("mm"));

}

private bool timetocheck()

{

bool pass = nurseAssistTextBox.Text != "" && timeIntubationTextBox1.Text != "" ? true : false;

return pass;

}

}

}

**Report Viewer:**

SELECT a.PK\_psPatRegisters,

e.fullname,

z.birthdate,

dbo.udf\_AgeInYears(z.birthdate, a.registrydate) AS AGE,

z.gender,

y.[dateIntubation],

y.[timeIntubation],

y.[nurseAssist],

dbo.udf\_GetAdmittingDoctors(a.pk\_pspatRegisters) AS AdmittingDoctors,

dbo.udf\_GetFullName(c.FK\_ASURenUser) as Renderer ,

y.[dateofOnset],

b.rendate

--,count(PK\_psPatRegisters)

--format(z.birthdate,'MM-dd-yyyy') as BirthDate,

--z.gender,

--a.dischdiagnosis,

--dbo.udf\_GetFullName(a.FK\_ASURegistry) as AssistBilling,

--a.dischdiagnosis,

--d.itemabbrev

FROM psPatRegisters a

JOIN psDatacenter e

ON e.PK\_psDatacenter = a.FK\_emdPatients

JOIN psPersonaldata z

ON z.PK\_psPersonalData = a.FK\_emdPatients

JOIN psEmergencies x

ON x.FK\_psPatRegisters = a.PK\_psPatRegisters

JOIN psPatitem b

ON a.PK\_psPatRegisters = b.FK\_psPatRegisters

JOIN psPatinv c

ON c.PK\_TRXNO = b.FK\_TRXNO

JOIN iwitems d

ON b.FK\_iwItemsREN = d.PK\_iwItems

JOIN [10.4.1.3\sqlcldh02].[HealthInfoDB].[dbo].[ETIntubationLogBook] y

ON y.FK\_psPatRegisters = a.PK\_psPatRegisters

WHERE d.barcodeid IN ('PAA000575','PAA000295','PAA000367') AND c.doctype <> 'CN' AND a.pattrantype = 'I' AND (y.[dateIntubation] BETWEEN @Start AND @End)

