using schoolmodel;

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace proiect7

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

///

enum ActionState

{

New, Edit, Delete, Nothing

}

public partial class MainWindow : Window

{

ActionState action = ActionState.Nothing;

schoolEntitiesmodel ctx = new schoolEntitiesmodel();

CollectionViewSource studentViewSource;

CollectionViewSource coursViewSource;

CollectionViewSource studentResultsViewSource;

public MainWindow()

{

InitializeComponent();

DataContext = this;

}

private void TabControl\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

this.studentViewSource = ((System.Windows.Data.CollectionViewSource)(this.FindResource("studentViewSource")));

this.studentViewSource.Source = ctx.Students.Local;

System.Windows.Data.CollectionViewSource studentViewSource = ((System.Windows.Data.CollectionViewSource)(this.FindResource("studentViewSource")));

// Load data by setting the CollectionViewSource.Source property:

// studentViewSource.Source = [generic data source]

System.Windows.Data.CollectionViewSource coursViewSource = ((System.Windows.Data.CollectionViewSource)(this.FindResource("coursViewSource")));

System.Windows.Data.CollectionViewSource studentResultsViewSource = ((System.Windows.Data.CollectionViewSource)(this.FindResource("studentResultsViewSource")));

// Load data by setting the CollectionViewSource.Source property:

// coursViewSource.Source = [generic data source]

studentViewSource.Source = ctx.Students.Local;

coursViewSource.Source = ctx.Courses.Local;

studentResultsViewSource.Source = ctx.Results.Local;

//<DataGridTextColumn x:Name="studentIdColumn1" Binding="{Binding StudentId}" Header="Student Id" Width="SizeToHeader"/>

ctx.Students.Load();

ctx.Courses.Load();

ctx.Results.Load();

//preia din baza de date dropdown button

comboCourse.ItemsSource = ctx.Courses.Local;

comboCourse.DisplayMemberPath = "CourseId";

comboStudent.ItemsSource = ctx.Students.Local;

comboStudent.DisplayMemberPath = "StudentId";

}

private void resultsDataGrid\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

private void btnNew\_Click(object sender, RoutedEventArgs e)

{

action = ActionState.New;

btnNew.IsEnabled = false;

btnEdit.IsEnabled = false;

btnDelete.IsEnabled = false;

btnSaveStudent.IsEnabled = true;

btnPrev.IsEnabled = false;

btnNext.IsEnabled = false;

txtLastName.IsEnabled = true;

txtFirstName.IsEnabled = true;

// txtStudentId.IsEnabled = true;

txtFirstName.Text = "";

txtLastName.Text = "";

//txtStudentId.Text = "";

Keyboard.Focus(txtStudentId);

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

Student student = null;

if (action == ActionState.New)

{

try

{

//instantiem Customer entity

student = new Student()

{

FirstName = txtFirstName.Text.Trim(),

LastName = txtLastName.Text.Trim(),

};

//adaugam entitatea nou creata in context

ctx.Students.Add(student);

studentViewSource.View.Refresh();

//salvam modificarile

ctx.SaveChanges();

}

catch (System.Data.DataException ex)

{

MessageBox.Show(ex.Message);

}

btnNew.IsEnabled = true;

btnEdit.IsEnabled = true;

btnSaveStudent.IsEnabled = false;

btnDelete.IsEnabled = true;

btnPrev.IsEnabled = true;

btnNext.IsEnabled = true;

txtStudentId.IsEnabled = false;

txtFirstName.IsEnabled = false;

txtLastName.IsEnabled = false;

}

else

if (action == ActionState.Edit)

{

try

{

student = (Student)studentDataGrid.SelectedItem;

int id\_student = student.StudentId;

Array array = ctx.Students.Local.ToArray();

foreach (Student stud in array)

{

if (stud.StudentId==id\_student)

{

stud.FirstName = firstNameTextBox.Text;

stud.LastName = lastNameTextBox.Text;

studentDataGrid.Items.Refresh();

ctx.SaveChanges();

studentDataGrid.Items.Refresh();

break;

}

}

}

catch (System.Data.DataException ex)

{

MessageBox.Show("Aici"+ex.Message);

}

// studentViewSource.View.Refresh();

// pozitionarea pe item-ul curent

studentViewSource.View.MoveCurrentTo(student);

btnNew.IsEnabled = true;

btnEdit.IsEnabled = true;

btnSaveStudent.IsEnabled = false;

btnDelete.IsEnabled = true;

btnPrev.IsEnabled = true;

btnNext.IsEnabled = true;

txtFirstName.IsEnabled = false;

txtLastName.IsEnabled = false;

}

else if (action == ActionState.Delete)

{

try

{

student = (Student)studentDataGrid.SelectedItem;

ctx.Students.Remove(student);

ctx.SaveChanges();

}

catch (System.Data.DataException ex)

{

MessageBox.Show(ex.Message);

}

studentViewSource.View.Refresh();

btnNew.IsEnabled = true;

btnEdit.IsEnabled = true;

btnSaveStudent.IsEnabled = false;

btnPrev.IsEnabled = true;

btnNext.IsEnabled = true;

txtFirstName.IsEnabled = false;

txtLastName.IsEnabled = false;

}

}

private void btnPrev\_Click(object sender, RoutedEventArgs e)

{

studentViewSource.View.MoveCurrentToPrevious();

}

private void btnNext\_Click(object sender, RoutedEventArgs e)

{

studentViewSource.View.MoveCurrentToNext();

}

private void btnEdit\_Click(object sender, RoutedEventArgs e)

{

studentDataGrid.Items.Refresh();

action = ActionState.Edit;

string tempFirstName = firstNameTextBox.Text.ToString();

string tempLastName = lastNameTextBox.Text.ToString();

btnNew.IsEnabled = false;

btnEdit.IsEnabled = false;

btnDelete.IsEnabled = false;

btnSaveStudent.IsEnabled = true;

btnPrev.IsEnabled = false;

btnNext.IsEnabled = false;

lastNameTextBox.IsEnabled = true;

firstNameTextBox.IsEnabled = true;

lastNameTextBox.Text = tempLastName;

firstNameTextBox.Text = tempFirstName;

Keyboard.Focus(firstNameTextBox);

studentDataGrid.Items.Refresh();

}

private void coursDataGrid\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

private void btnSaveMark\_Click(object sender, RoutedEventArgs e)

{

txtMark.IsEnabled = true;

int parsedValue;

if (string.IsNullOrWhiteSpace(txtMark.Text) || !int.TryParse(txtMark.Text, out parsedValue) )

{

string message = "Mark is empty or contains alphabet";

MessageBox.Show(message);

}

else {

string message = "Mark is saved";

if (parsedValue>10 || parsedValue < 1)

{

message = "Note intre 1 si 10";

}

Result result = new Result();

result.Nota = txtMark.Text;

result.CourseId = int.Parse(comboCourse.Text);

result.StudentId= int.Parse(comboStudent.Text);

try

{

//adaugam entitatea nou creata in context

//salvam modificarile

//

Array array = ctx.Results.Local.ToArray();

int c = 0;

foreach (Result rez in array)

{

if (rez.CourseId==result.CourseId && rez.StudentId==result.StudentId && rez.Nota != null)

{

c = 1;

break;

}

}

if (c == 1)

{

MessageBox.Show("Nota existenta");

}

else

{

ctx.Results.Add(result);

ctx.SaveChanges();

}

}

catch (System.Data.DataException ex)

{

MessageBox.Show(ex.Message);

}

}

}

private void btnDelete\_Click(object sender, RoutedEventArgs e)

{

try

{

int id\_curs = int.Parse(comboCourse.Text);

int id\_student = int.Parse(comboStudent.Text);

Array array = ctx.Results.Local.ToArray();

int c = 0;

foreach (Result rez in array)

{

if (rez.CourseId == id\_curs && rez.StudentId == id\_student)

{

if (MessageBox.Show("Sure you want to delete from student with id " + id\_student + " at course with id " + id\_curs + " ?", "Question", MessageBoxButton.YesNo,

MessageBoxImage.Warning) == MessageBoxResult.Yes)

{

ctx.Results.Remove(rez);

resultsDataGrid.Items.Refresh();

ctx.SaveChanges();

resultsDataGrid.Items.Refresh();

}

}

}

}

catch

{

}

}

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

{

}

private void btnEditMark\_Click(object sender, RoutedEventArgs e)

{

txtMark.IsEnabled = true;

int id\_curs = int.Parse(comboCourse.Text);

int id\_student = int.Parse(comboStudent.Text);

Array array = ctx.Results.Local.ToArray();

int c = 0;

foreach (Result rez in array)

{

if (rez.CourseId == id\_curs && rez.StudentId == id\_student )

{

c = 1;

int parsedValue;

if (string.IsNullOrWhiteSpace(txtMark.Text) || !int.TryParse(txtMark.Text, out parsedValue))

{

string message = "Mark is empty or contains alphabet";

MessageBox.Show(message);

break;

}

else

{

string message = "Bravo Batman";

if (parsedValue > 10 || parsedValue < 1)

{

message = "Note intre 1 si 10";

MessageBox.Show(message);

break;

}

rez.Nota = parsedValue.ToString();

resultsDataGrid.Items.Refresh();

ctx.SaveChanges();

resultsDataGrid.Items.Refresh();

}

break;

}

}

// if (c == 0) {

// MessageBox.Show("Nu exista nota la elevul ales");

// }

}

private void comboStudent\_find\_mark(object sender, EventArgs e)

{

try

{

if (comboCourse.IsDropDownOpen == false)

{

int id\_curs = int.Parse(comboCourse.Text);

int id\_student = int.Parse(comboStudent.Text);

txtMark.Text = this.get\_nota\_for\_given\_courseid\_and\_studentid(id\_curs, id\_student);

}

}

catch { }

}

private void comboCourse\_find\_mark(object sender, EventArgs e)

{

try

{

if (comboCourse.IsDropDownOpen == false)

{

int id\_curs = int.Parse(comboCourse.Text);

int id\_student = int.Parse(comboStudent.Text);

txtMark.Text = this.get\_nota\_for\_given\_courseid\_and\_studentid(id\_curs, id\_student);

}

}catch {

}

}

private string get\_nota\_for\_given\_courseid\_and\_studentid(int idcurs, int idstudent)

{

Array array = ctx.Results.Local.ToArray();

int c = 0;

foreach (Result rez in array)

{

if (rez.CourseId == idcurs && rez.StudentId == idstudent )

{

return rez.Nota;

}

}

return "";

}

private void btnExit\_Click(object sender, RoutedEventArgs e)

{

if (MessageBox.Show("Close Application?", "Question", MessageBoxButton.YesNo,

MessageBoxImage.Warning) == MessageBoxResult.Yes)

{

this.Close();

}

}

}

}