//LogicService

namespace velvetEyebrows

{

class ForService

{

beautyEntities db = new beautyEntities();

public List<Service> ServiceList;

//Вывод всех записей

public void LoadService()

{

this.ServiceList = db.Service.ToList();

}

//Функция сортировки

public void Sort(int Sorting = -1)

{

if(Sorting == 0)

this.ServiceList = this.ServiceList.OrderBy(p => p.Cost).ToList();

else if (Sorting == 1)

this.ServiceList = this.ServiceList.OrderByDescending(p => p.Cost).ToList();

}

//Поиск

public void Search(string ValueSearch)

{

if (string.IsNullOrEmpty(ValueSearch)) return;

List<Service> Newlist = new List<Service>();

foreach (var item in this.ServiceList)

{

if (item.Title.Contains(ValueSearch) || item.Description.Contains(ValueSearch))

{

Newlist.Add(item);

}

this.ServiceList = Newlist;

}

}

//Фильтрация

public void Filtr(int index = -1)

{

switch (index)

{

case 1:

this.ServiceList = this.ServiceList.Where(p => p.Discount >= 0 && p.Discount < 0.05).ToList();

break;

case 2:

this.ServiceList = this.ServiceList.Where(p => p.Discount >= 0.05 && p.Discount < 0.15).ToList();

break;

case 3:

this.ServiceList = this.ServiceList.Where(p => p.Discount >= 0.15 && p.Discount < 0.3).ToList();

break;

case 4:

this.ServiceList = this.ServiceList.Where(p => p.Discount >= 0.3 && p.Discount < 0.7).ToList();

break;

case 5:

this.ServiceList = this.ServiceList.Where(p => p.Discount >= 0.7 && p.Discount <= 1).ToList();

break;

}

}

}

}

//LogicClient

namespace velvetEyebrows

{

class LogicClient

{

beautyEntities db = new beautyEntities();

//Джоин

public List<PsevdoClient> psevdoClients;

public void LoadData()

{

var query = from cli in db.Client

join gen in db.Gender on cli.GenderCode equals gen.Code

select new PsevdoClient

{

ID = cli.ID,

FirstName = cli.FirstName,

LastName = cli.LastName,

Birthday = cli.Birthday,

RegistrationDate = cli.RegistrationDate,

Email = cli.Email,

Phone = cli.Phone,

GenderName = gen.Name,

PhotoPath = cli.PhotoPath

};

psevdoClients = query.ToList();

SetPhotos();

}

//Установка фоты

public void SetPhotos()

{

foreach (var item in this.psevdoClients)

{

string path = Path.Combine(Application.StartupPath, item.PhotoPath);

using (FileStream stream = new FileStream(path, FileMode.Open))

{

item.Photo = Image.FromStream(stream);

item.Photo = new Bitmap(item.Photo, new Size(30, 30));

}

}

}

}

}

// Наследование класса, задание полей

class PsevdoClient : Client

{

public string GenderName { get; set; }

public Image Photo { get; set; }

}

//Форма добавления

public partial class AddService : Form

{

beautyEntities db = new beautyEntities();

string path, pathName;

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

{

GetPhoto();

}

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

{

AddData();

}

//Добавление данных

private void AddData()

{

Service list = new Service

{

Title = TxtTitle.Text,

Cost = Convert.ToDecimal(TxtCost.Text),

DurationInSeconds = Convert.ToInt32(NumDuration.Value),

Description = TxtDiscr.Text,

Discount = Convert.ToDouble(TxtDiscount.Text)/100,

MainImagePath = pathName

};

db.Service.Add(list);

db.SaveChanges();

MessageBox.Show("Успешно", "Запись успешно добавлена");

PicPhoto.Image.Save(path);

this.Close();

new ServicesList(true).Show();

}

//Загрузка фото с компа пользователя

private void GetPhoto()

{

using(OpenFileDialog fileDialog = new OpenFileDialog())

{

if(fileDialog.ShowDialog() == DialogResult.OK)

{

using(FileStream stream = new FileStream(fileDialog.FileName, FileMode.Open))

{

var img = Image.FromStream(stream);

PicPhoto.Image = new Bitmap(img);

path = Path.Combine(Application.StartupPath, fileDialog.SafeFileName);

pathName = fileDialog.SafeFileName;

}

}

//Форма авторизации

public partial class Authorization : Form

{

beautyEntities db = new beautyEntities();

string Captcha;

Random Random = new Random();

//Нажатие на кнопку войти

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

{

try

{

if (string.Empty != txtLog.Text && string.Empty != txtPassword.Text)

{

if (db.Client.Any(p => p.Email == txtLog.Text && p.FirstName == txtPassword.Text))

{

var user = db.Client.Where(p => p.Email == txtLog.Text && p.FirstName == txtPassword.Text).FirstOrDefault();

if (txtCaptcha.Text == Captcha) new TimeForm(user).Show();

else

{

MessageBox.Show("Каптча введена неверно!");

txtCaptcha.Text = "";

lblCaptcha.Text = "";

GenCaptcha();

}

}

else

{

MessageBox.Show("Проверьте корректность введенных данных");

}

}

else

{

MessageBox.Show("Поля не должны быть пустыми!");

}

}

catch (Exception ex) { }

}

//Что-то вроде капчи

private void GenCaptcha()

{

Captcha = "";

string Symbols = "q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z,x,c,v,b,n,m";

Symbols += "Q,W,E,R,T,Y,U,I,O,P,A,S,D,F,G,H,J,K,L,Z,X,C,V,B,N,M";

Symbols += "1,2,3,4,5,6,7,8,9";

string [] ListSymbols = Symbols.Split(',');

for(int i = 0; i < 5; i++)

{

Captcha += ListSymbols[Random.Next(0, ListSymbols.Length - 1)];

}

lblCaptcha.Text = Captcha;

}

}

//Форма изменения

public partial class Edit : Form

{

beautyEntities db = new beautyEntities();

Service ServiceL;

string PathPhoto;

string path;

public Edit(int id)

{

InitializeComponent();

ServiceL = db.Service.Where(p => p.ID == id).FirstOrDefault();

}

//Добавление фото с компа пользователя при изменении

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

{

using (OpenFileDialog openFile = new OpenFileDialog())

{

if(openFile.ShowDialog() == DialogResult.OK)

{

using(FileStream stream = new FileStream(openFile.FileName, FileMode.Open))

{

var img = Image.FromStream(stream);

PicPhoto.Image = new Bitmap(img);

PathPhoto = openFile.SafeFileName;

path = Path.Combine(Application.StartupPath, openFile.SafeFileName);

}

}

}

}

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

{

LoadDate();

GetPhoto();

}

//Подгрузка данных в поля

private void LoadDate()

{

TxtCost.Text = (ServiceL.Cost.ToString().Remove(ServiceL.Cost.ToString().Length-5));

TxtDiscount.Text = (ServiceL.Discount\*100).ToString();

TxtTitle.Text = ServiceL.Title;

NumDuration.Value = ServiceL.DurationInSeconds;

TxtDiscr.Text = ServiceL.Description;

}

//Изменение данных

private void EditData()

{

try

{

ServiceL.Title = TxtTitle.Text;

ServiceL.Cost = Convert.ToDecimal(TxtCost.Text);

ServiceL.Description = TxtDiscr.Text;

ServiceL.Discount = Convert.ToDouble(TxtDiscount.Text) \* 100;

ServiceL.DurationInSeconds = Convert.ToInt32(NumDuration.Value);

ServiceL.MainImagePath = PathPhoto;

db.SaveChanges();

PicPhoto.Image.Save(path);

MessageBox.Show("Данные успешно изменены", "Изменено");

this.Close();

new ServicesList(true).Show();

}

catch (Exception)

{

MessageBox.Show("Что-то пошло не так", "Кажется, произошла оишбка!");

}

}

//Вывод фото при загрузке формы

private void GetPhoto()

{

string PathPhotoForEdit = Path.Combine(Application.StartupPath, ServiceL.MainImagePath);

using(FileStream stream = new FileStream(PathPhotoForEdit, FileMode.Open))

{

var Img = Image.FromStream(stream);

PicPhoto.Image = new Bitmap(Img);

PathPhoto = ServiceL.MainImagePath;

}

}

}

//Доброе утро и так далее вариант 1

private void GetTime()

{

DateTime TimeNow = DateTime.Now;

if ((TimeNow.Hour >= 9 && TimeNow.Hour < 11) || (TimeNow.Minute == 0 && TimeNow.Hour == 11)) lblTime.Text = "Доброе утро";

else if ((TimeNow.Hour >= 11 && TimeNow.Hour < 18) || (TimeNow.Minute == 0 && TimeNow.Hour == 18)) lblTime.Text = "Добрый день";

else if (TimeNow.Hour >= 18) lblTime.Text = "Добрый вечер";

}

//Доброе утро и так далее вариант 2

DateTime TimeNow = DateTime.Now;

TimeSpan time = new TimeSpan(TimeNow.Hour, TimeNow.Minute, 0);

if (time >= new TimeSpan(9, 0, 0) && time < new TimeSpan(11, 1, 0) )

TimeLab.Text = "Доброе утро";

else if(time >= new TimeSpan(11, 1, 0) && time < new TimeSpan(18, 1, 0))

TimeLab.Text = "Добрый день";

else if (time >= new TimeSpan(18, 1, 0) && time <= new TimeSpan(23, 59, 0) || time == new TimeSpan(0, 0, 0))

TimeLab.Text = "Добрый вечер";

//Фильтрация по направлению

public void FilterDirection(string nameDir)

{

try

{

if (nameDir == "Все") return;

else if (nameDir != "") this.Events = this.Events.Where(p => p.DirectionPS == nameDir).ToList();

}

catch (Exception)

{

}

При вызове отправляем текст в прошлую функцию:

logic.FilterDirection(FiltrDirection.Text);

//Задача комбобокса с пунктом «Все»

void LoadCOmboBox()

{

var DataList = db.Direction.ToList();

DataList.Insert(0, new Direction

{

ID = 0,

Name = "Все"

});

FiltrDirection.DataSource = DataList;

FiltrDirection.DisplayMember = "Name";

FiltrDirection.ValueMember = "ID";

FiltrDirection.SelectedIndex = 0;

}

//Переименовать столбцы, скрыть, задать порядок с нуля

void Rename()

{

tableEvent.Columns["ID"].Visible = false;

tableEvent.Columns["Name"].HeaderText = "Название";

tableEvent.Columns["Name"].DisplayIndex = 0;

tableEvent.Columns["StartEvent"].HeaderText = "Начало";

tableEvent.Columns["EndEvent"].HeaderText = "Конец";

tableEvent.Columns["CityPs"].HeaderText = "Город";

tableEvent.Columns["DirectionPs"].HeaderText = "Направление";

tableEvent.Columns["Photo"].HeaderText = "Логотип";

tableEvent.Columns["Photo"].DisplayIndex = 2;

tableEvent.Columns["DirectionPs"].DisplayIndex = 1;

tableEvent.Columns["DirectionId"].Visible = false;

tableEvent.Columns["CityId"].Visible = false;

tableEvent.Columns["PhotoPath"].Visible = false;

tableEvent.Columns["ActivityEvents"].Visible = false;

tableEvent.Columns["City"].Visible = false;

tableEvent.Columns["Direction"].Visible = false;

tableEvent.Columns["UserEvent"].Visible = false;

}