**1 Разработка автоматизированной информационной системы**

**1.1 Постановка задачи**

Разработать информационную систему по ведению базы данных «Автомойка» содержащую информацию о услугах мойки, которые заказали клиенты в автомойке.

База данных должна содержать следующую информацию:

* таблица **«Список Клиентов»: номер, фамилия, имя, отчество, модель машины клиента.**
* таблица **«Список услуг»: номер, наименование услуги, стоимость услуги.**
* таблица **«Список работников»: номер, фамилия, имя, отчество, дата рождения работника .** Данная таблица должна быть связана с номером книги и номером читательского билета.
* таблица **«Оказанных услуг»: номер клиента, номер услуги, номер работника.** Данная таблица должна быть связана с номером клиента, номером услуги и номером работника-исполнителя.

Определить ключи (если необходимо добавьте поля). Автоматизировать поля ввода данных с помощью выпадающих списков или маски ввода. Предусмотреть контроль ввода данных, применив блокировку некорректного ввода данных.

Создать унифицированную форму для добавления, удаления и редактирования данных в существующих таблицах.

Создать форму «Предоставление услуги», которая должна содержать следующие поля: номер, ФИО клиента и модель машины клиента, наименование и цена услуги, ФИО и дата рождения работника.

**1.2 Структура таблиц в БД**

Таблица 1.1 - Структура таблицы «user»

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название поля** | **Тип данных** | **Описание поля** |
| 1 | idUser | integer | Идентификатор поля. Первичный ключ |
| 2 | FullName | varchar(45) | Имя пользователя |
| 3 | Login | Varchar(45) | логин |
| 4 | Password | Varchar(45) | Пароль |
| 5 | AccessRight | Enum | Роль(права доступа) |

Таблица 1.2 - Структура таблицы «listClients»

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название поля** | **Тип данных** | **Описание поля** |
| 1 | idClient | integer | Идентификатор поля. Первичный ключ |
| 2 | Name | varchar(15) | Имя клиента |
| 3 | Surname | varchar(15) | Фамилия |
| 4 | Patronomic | varchar(15) | Отчество |
| 5 | CarModel | varchar(15) | Модель машины |

Таблица 1.3 - Структура таблицы «listWashers»

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название поля** | **Тип данных** | **Описание поля** |
| 1 | idWasher | integer | Идентификатор поля. Первичный ключ |
| 2 | Name | varchar(15) | Имя работника (мойщика) |
| 3 | Surname | varchar(15) | Фамилия |
| 4 | Patronomic | varchar(15) | Отчество |
| 5 | DataOB | Date | Дата рождения |

Таблица 1.4- Структура таблицы «listServices»

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название поля** | **Тип данных** | **Описание поля** |
| 1 | idService | integer | Идентификатор поля. Первичный ключ |
| 2 | ServiceName | varchar(45) | Наименование услуги |
| 3 | ServicePrice | integer | Цена услуги |

Таблица 1.5 - Структура таблицы «ProvisionServices»

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название поля** | **Тип данных** | **Описание поля** |
| 1 | idProvisionService | integer | Идентификатор поля. Первичный ключ |
| 2 | idClient | integer | Идентификатор клиента |
| 3 | idService | integer | Идентификатор услуги |
| 4 | idWasher | integer | Идентификатор работника(мойщика) |

**1.3 Схема базы данных**

На рисунке 1.1 представлена схема базы данных и используемые связи.

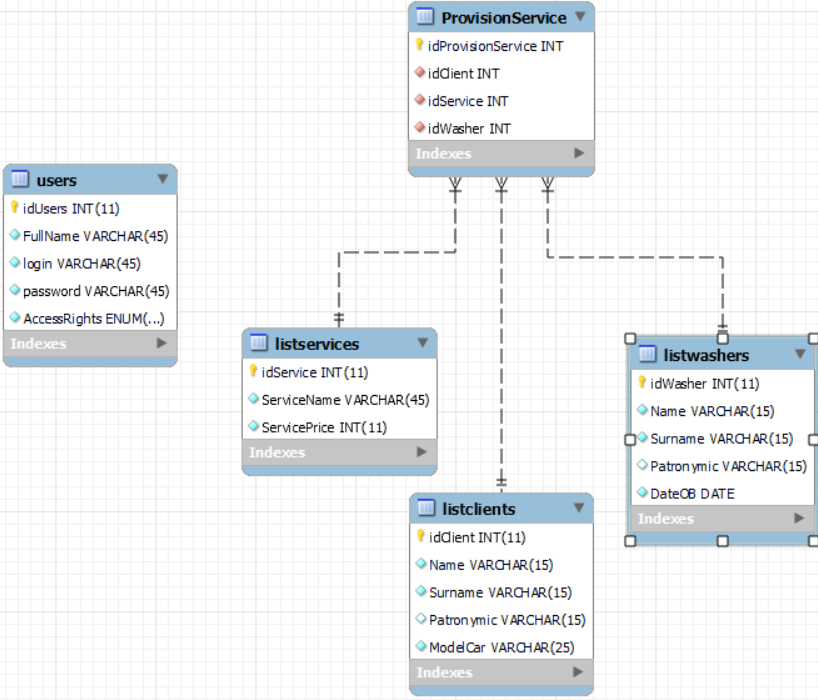


Рисунок 1.1 - Схема базы данных

**1.4 Описание программных модулей**

Главная форма программы представлена на рисунке 1.2.

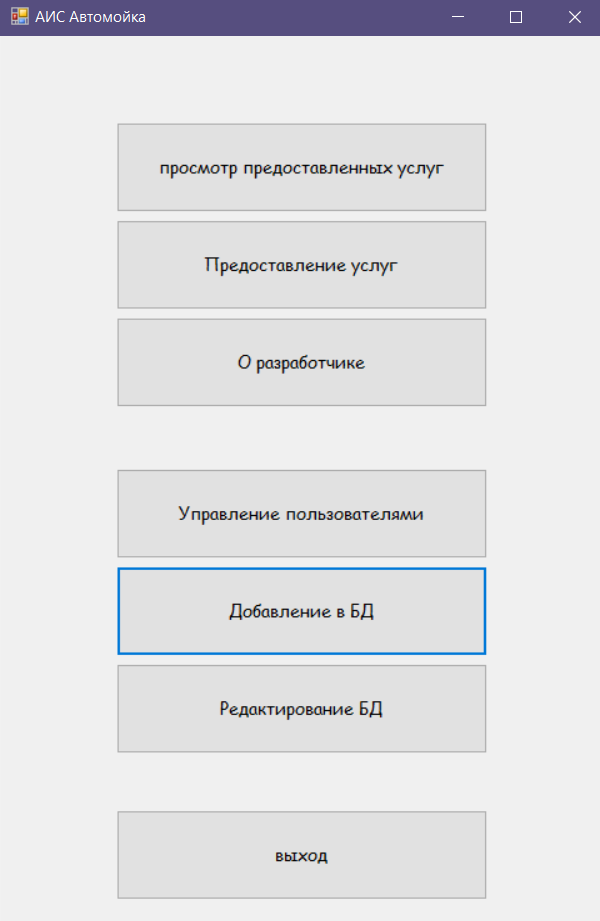


Рисунок 1.2 – Главная форма программы

Форма добавления данных в таблицы представлена на рисунке 1.3.

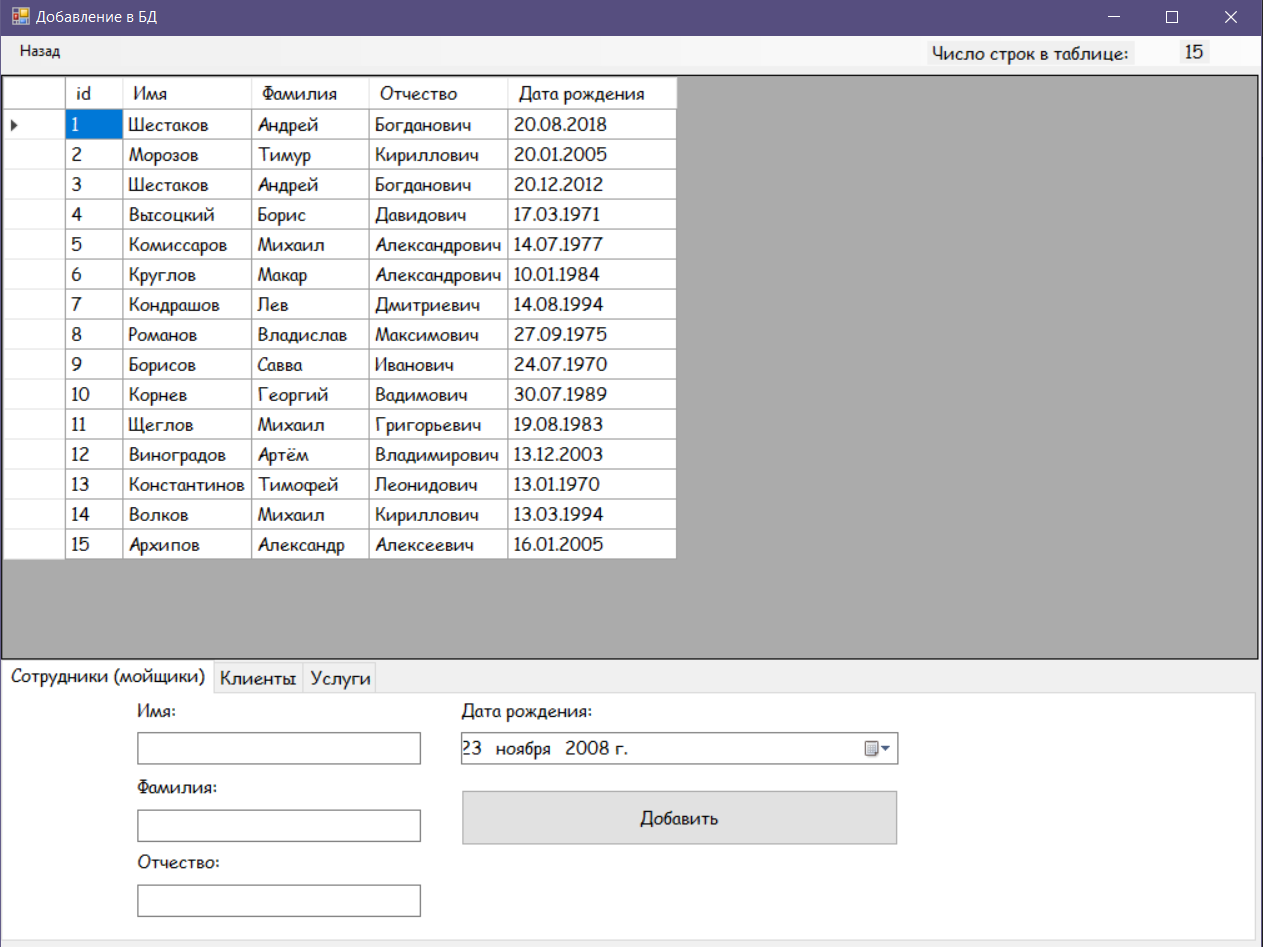


Рисунок 1.3 – Форма добавления информации в базу данных

Форма редактирование данных в таблицах представлена на рисунке 1.4.

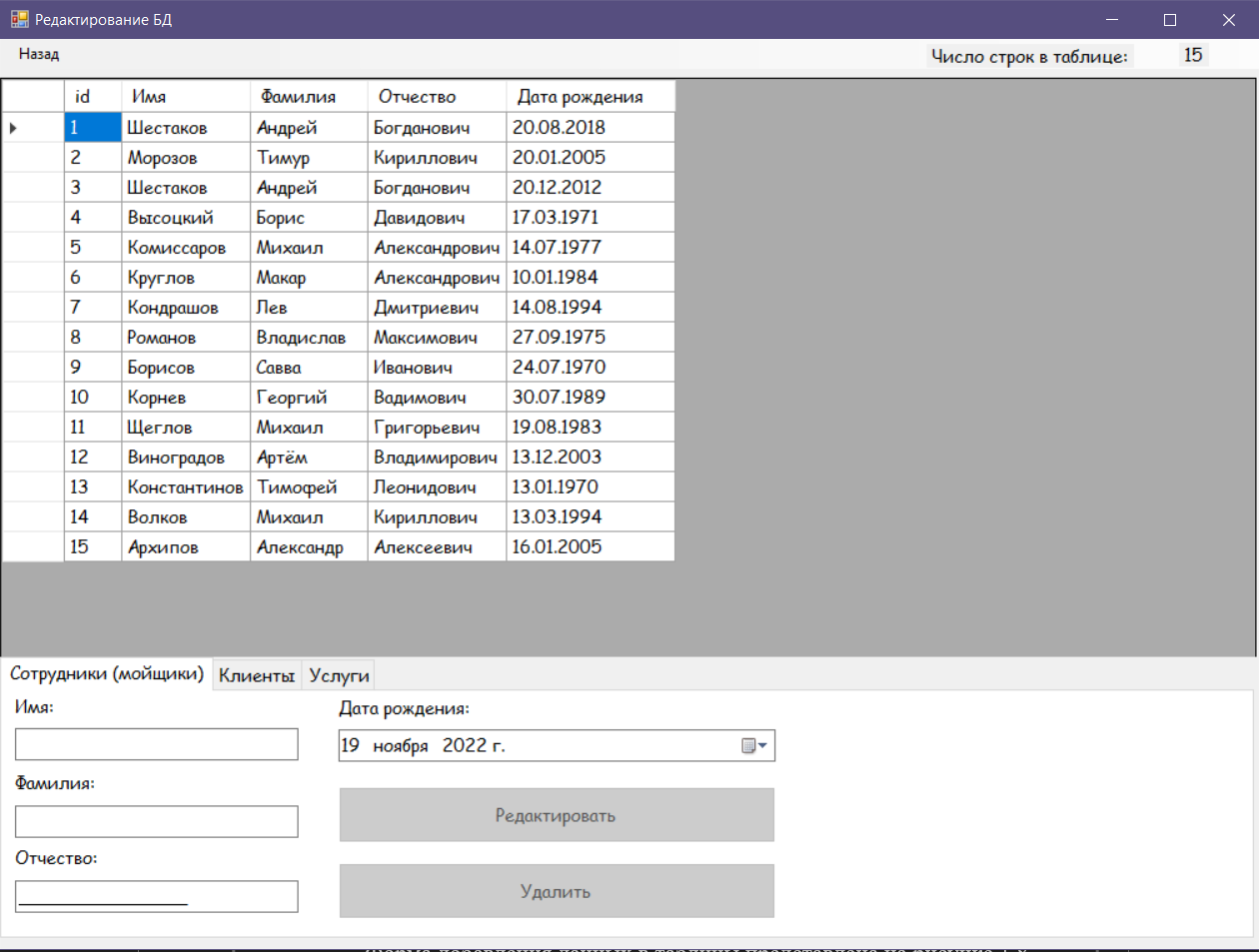


Рисунок 1.4 – Форма редактирование данных в таблицах

Форма управления пользователями представлена на рисунке 1.5.

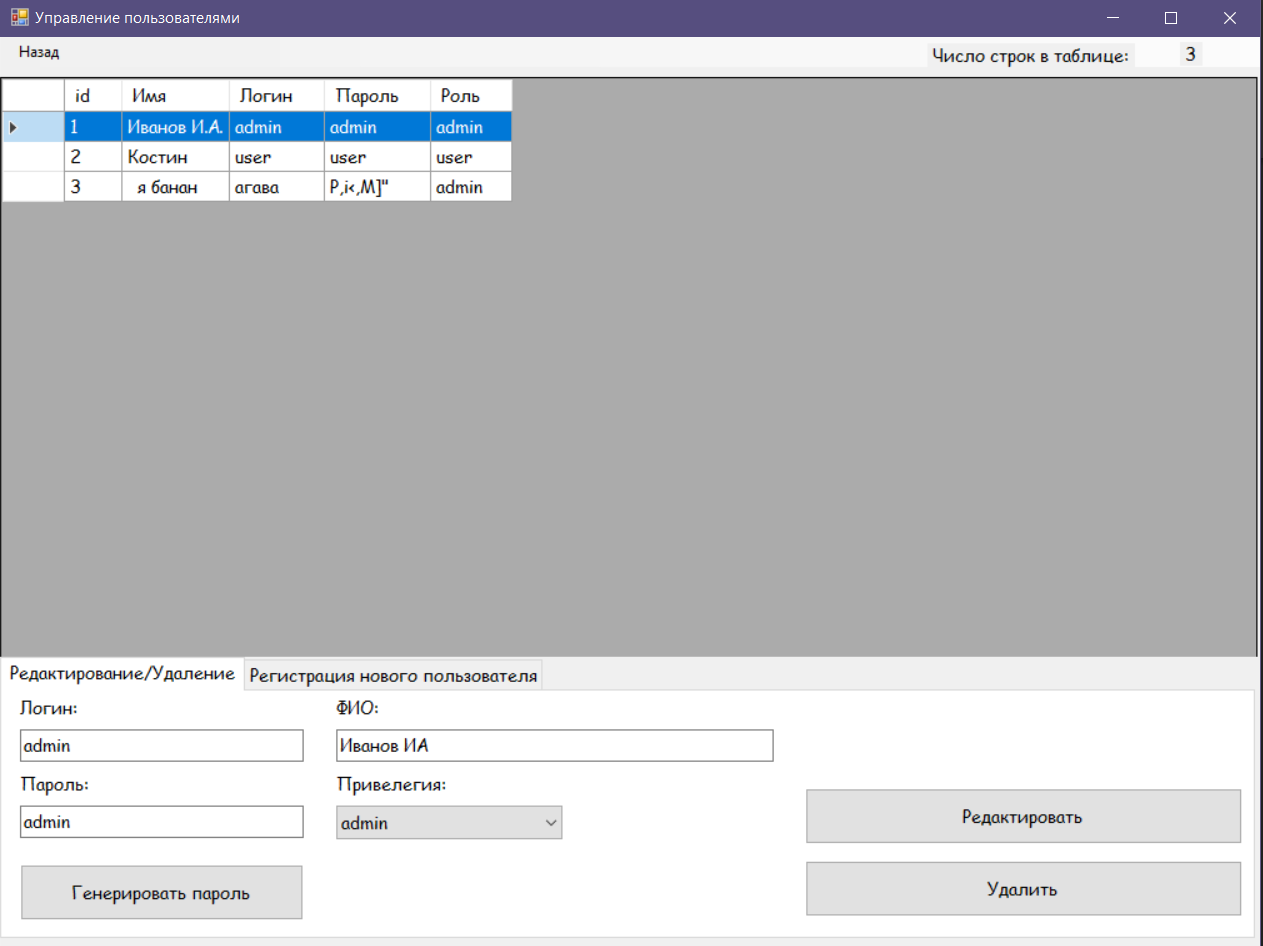


Рисунок 1.5 – Форма управления пользователями

Форма оказание услуги представлена на рисунке 1.6.

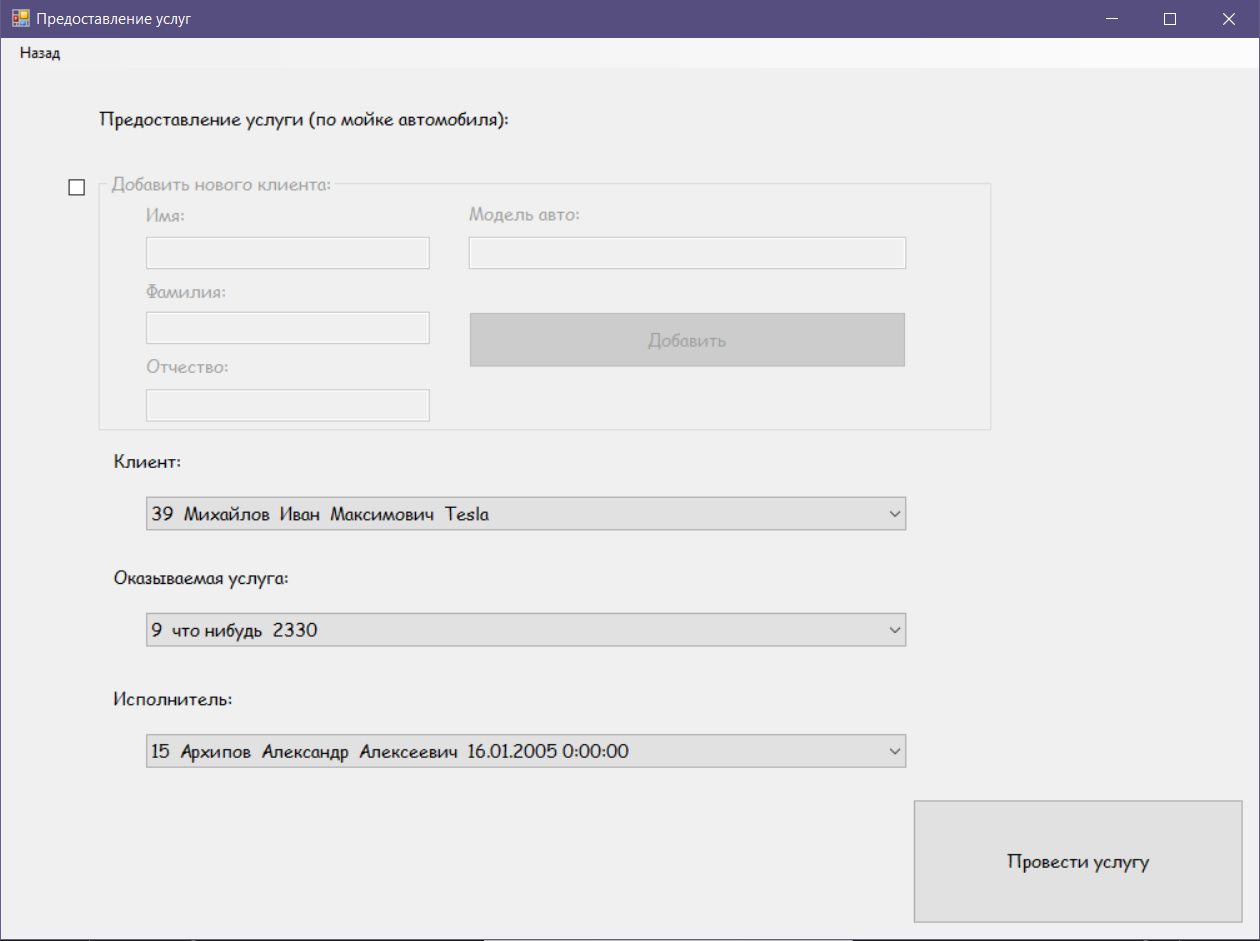


Рисунок 1.6 – Форма оказание услуги

Форма просмотра таблицы оказанных услуг представлена на рисунке 1.7.

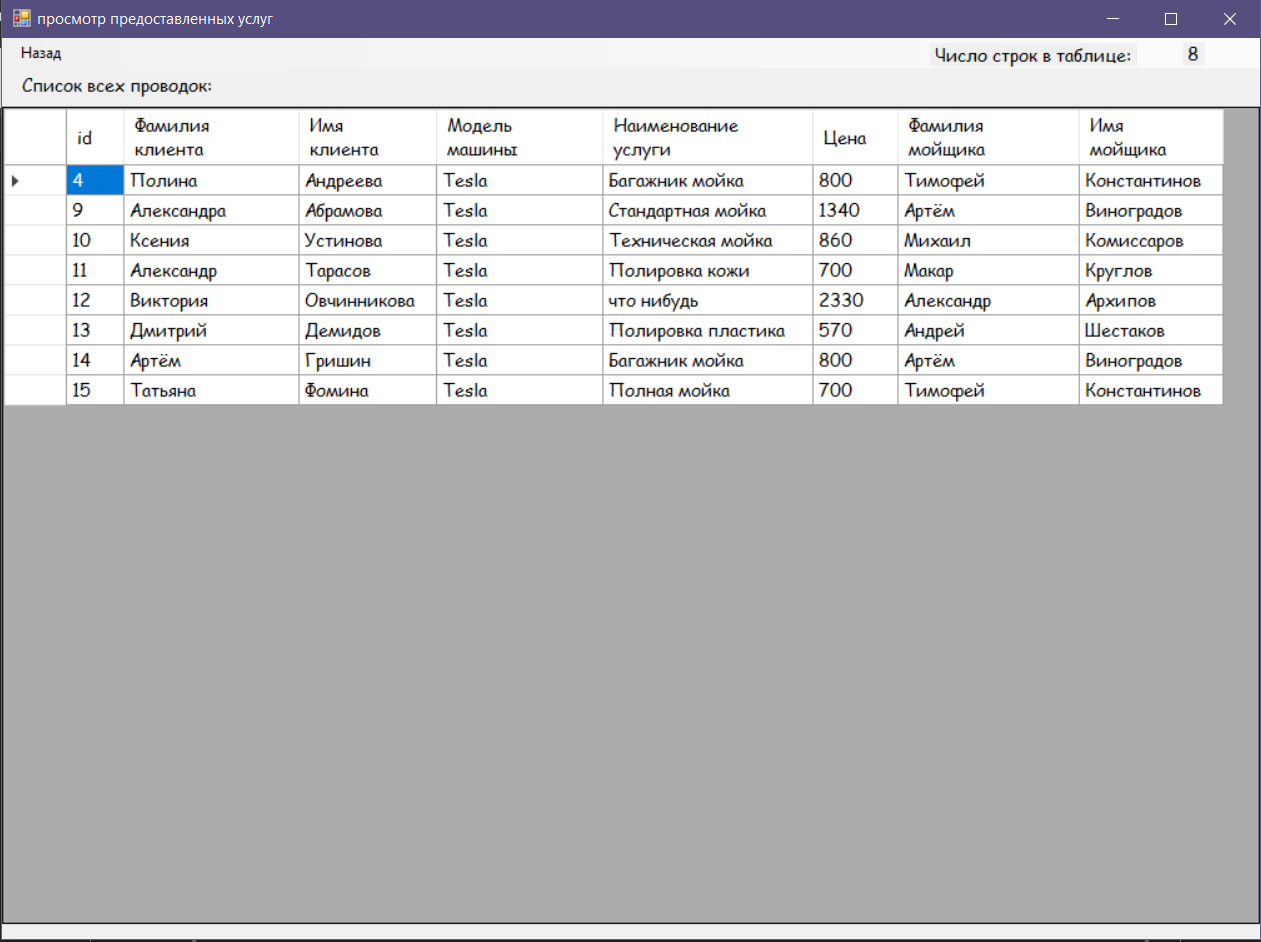


Рисунок 1.7 – Форма просмотра таблицы оказанных услуг

Форма авторизации представлена на рисунке 1.8.

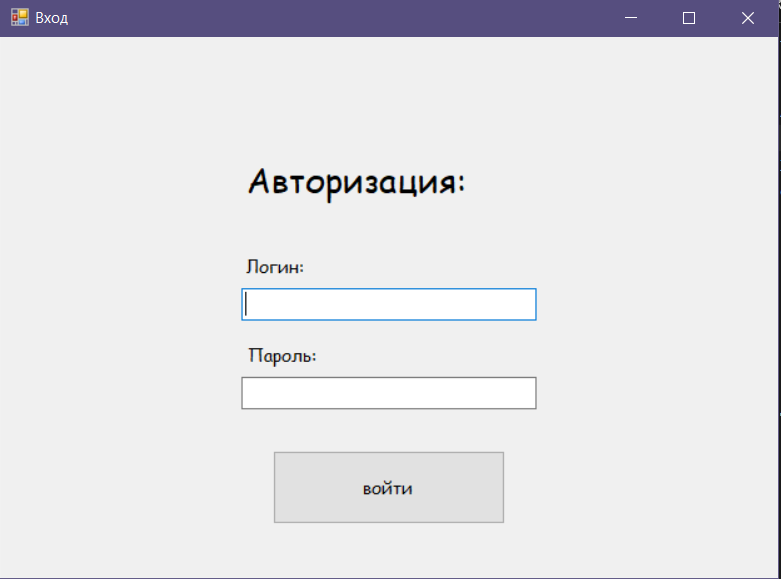


Рисунок 1.8 – Форма авторизации

Форма о разработчике представлена на рисунке 1.9.

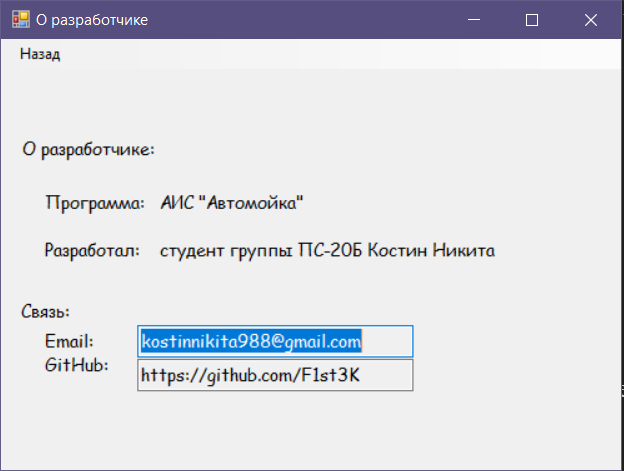


Рисунок 1.9 – Форма о разработчике

**1.5 Листинг программы**

**MainForm.cs** – главная форма

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 AISCarwash

{

public partial class MainForm : Form

{

private string \_mode;

public MainForm(string mode)

{

InitializeComponent();

this.CenterToScreen();

if (mode == "user" || mode == "admin")

\_mode = mode;

else Close();

}

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

{

if (\_mode == "user")

ActivateUserMode();

}

private void ActivateUserMode()

{

buttonUserControl.Enabled = false;

buttonUserControl.Visible = false;

buttonAdd.Enabled = false;

buttonAdd.Visible = false;

buttonChange.Enabled = false;

buttonChange.Visible = false;

}

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

{

Close();

Application.Exit();

}

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

{

this.Close();

UserControlForm form = new UserControlForm(\_mode);

form.Text = buttonUserControl.Text;

form.Show();

}

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

{

this.Close();

AddForm form = new AddForm(\_mode);

form.Text = buttonAdd.Text;

form.Show();

}

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

{

this.Close();

ChangeForm form = new ChangeForm(\_mode);

form.Text = buttonChange.Text;

form.Show();

}

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

{

this.Close();

ProvisionServies form = new ProvisionServies(\_mode);

form.Text = buttonProvisionServies.Text;

form.Show();

}

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

{

this.Close();

ViewForm form = new ViewForm(\_mode);

form.Text = buttonView.Text;

form.Show();

}

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

{

this.Close();

DeveloperInfoForm form = new DeveloperInfoForm(\_mode);

form.Text = buttonInfo.Text;

form.Show();

}

}

}

**AutorizationForm.cs** – форма авторизации

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 AISCarwash

{

public partial class AutorizationForm : Form

{

public AutorizationForm()

{

InitializeComponent();

this.CenterToScreen();

textPassword.PasswordChar = '•';

}

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

{

}

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

{

string column = "AccessRights";

string tableName = "users";

string condition = $"login = '{textLogin.Text}' AND password = '{textPassword.Text}'";

DataTable resultTable = MySqlConnecter.QueryReturnTable(column, tableName, condition);

if (resultTable.Rows.Count == 0)

MessageBox.Show("\*неверный логин и/или пароль\*", "неудачный вход", MessageBoxButtons.OK, MessageBoxIcon.Error);

else

{

string mode = resultTable.Rows[0][0].ToString();

MessageBox.Show($"\*Успех вы вошли как {mode}\*", "удачный вход", MessageBoxButtons.OK);

LogIn(mode);

}

}

private void LogIn(string mode)

{

this.Hide();

MainForm form = new MainForm(mode);

form.Show();

}

}

}

**UserControlForm.cs** – форма управления пользователями

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 AISCarwash

{

public partial class UserControlForm : Form

{

private string \_mode;

private DataGridViewRow \_currentRow;

public UserControlForm(string mode)

{

\_mode = mode;

InitializeComponent();

this.CenterToScreen();

}

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

{

UpdateGridView();

string[] modes = new string[] { "user", "admin"};

ChangeMode.DataSource = modes;

CreateMode.DataSource = modes;

ChangeMode.DropDownStyle = ComboBoxStyle.DropDownList;

CreateMode.DropDownStyle = ComboBoxStyle.DropDownList;

}

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

{

this.Close();

MainForm form = new MainForm(\_mode);

form.Show();

}

private void ResetChange()

{

textChangeFullname.ResetText();

textChangeLogin.ResetText();

textChangePassword.ResetText();

ChangeMode.ResetText();

}

private void ResetCreate()

{

textCreateFullname.ResetText();

textCreateLogin.ResetText();

textCreatePassword.ResetText();

CreateMode.ResetText();

}

private void UpdateGridView()

{

string tableName = "users";

string column = "\*";

string condition = $"EXISTS (SELECT {column} FROM {tableName})";

dataGridView.DataSource = MySqlConnecter.QueryReturnTable(column, tableName, condition);

dataGridView = MySqlConnecter.ChangeColumnsName(dataGridView, "id", "Имя", "Логин", "Пароль", "Роль");

counterTable.Text = dataGridView.RowCount.ToString();

}

private void dataGridView\_CellClick(object sender, DataGridViewCellEventArgs e)

{

\_currentRow = dataGridView.Rows[dataGridView.CurrentCell.RowIndex];

\_currentRow.Selected = true;

buttonChange.Enabled = true;

buttonDelete.Enabled = true;

GetDateChange();

}

private void GetDateChange()

{

textChangeFullname.Text = \_currentRow.Cells[1].Value.ToString();

textChangeLogin.Text = \_currentRow.Cells[2].Value.ToString();

textChangePassword.Text = \_currentRow.Cells[3].Value.ToString();

ChangeMode.Text = \_currentRow.Cells[4].Value.ToString();

}

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

{

if (MySqlConnecter.StringsIsEmpty(textChangeLogin.Text, textChangePassword.Text))

return;

if (ExistUserLogin(textChangeLogin.Text, 1))

return;

string table = "users";

string values = $" {\_currentRow.Cells[1].OwningColumn.Name} = '{textChangeFullname.Text}', " +

$"{\_currentRow.Cells[2].OwningColumn.Name} = '{textChangeLogin.Text}', " +

$"{\_currentRow.Cells[3].OwningColumn.Name} = '{textChangePassword.Text}', " +

$"{\_currentRow.Cells[4].OwningColumn.Name} = '{ChangeMode.Text}'";

string condition = $"{\_currentRow.Cells[0].OwningColumn.Name} = {\_currentRow.Cells[0].Value.ToString()}";

MySqlConnecter.QueryChangeInTable(table, values, condition);

UpdateGridView();

ResetChange();

}

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

{

if(MySqlConnecter.StringsIsEmpty(textCreateLogin.Text, textCreatePassword.Text))

return;

if (ExistUserLogin(textCreateLogin.Text, 0))

return;

string table = "users";

string values = $"DEFAULT, '{textCreateFullname.Text}', '{textCreateLogin.Text}', '{textCreatePassword.Text}', '{CreateMode.Text}'";

MySqlConnecter.QueryAddInTable(table, values);

UpdateGridView();

ResetCreate();

}

private bool ExistUserLogin(string login, int countLogin)

{

string column = "\*";

string tableName = "users";

string condition = $"login = '{login}'";

DataTable table = MySqlConnecter.QueryReturnTable(column, tableName, condition);

if (table.Rows.Count > countLogin)

{

MessageBox.Show("Пользователь с таким логином уже существует", "EROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

return true;

}

return false;

}

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

{

string row = "";

for (int i = 0; i < dataGridView.Columns.Count; i++) row += "\t" + \_currentRow.Cells[i].Value.ToString();

DialogResult dialogResult = MessageBox.Show($"Вы действительно хотите удалить следующую запись?: \n{row}", "Внимание! УДАЛЕНИЕ!", MessageBoxButtons.YesNo, MessageBoxIcon.Asterisk);

if (dialogResult == DialogResult.Yes)

DeleteUser();

}

private void DeleteUser()

{

string table = "users";

string condition = $"{\_currentRow.Cells[0].OwningColumn.Name} = {\_currentRow.Cells[0].Value.ToString()}";

MySqlConnecter.QueryDeleteInTable(table, condition);

UpdateGridView();

ResetChange();

}

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

{

textChangePassword.Text = new RandomPassword(8, 25).Password;

}

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

{

textCreatePassword.Text = new RandomPassword(8, 25).Password;

}

}

}

**RandomPassword.cs** – класс для генерации пароля

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AISCarwash

{

internal class RandomPassword

{

public string Password { get; set; }

private Random rand = new Random();

int lenght;

public RandomPassword(int minLenght, int maxLenght)

{

lenght = rand.Next(minLenght, maxLenght);

CreatePassword();

}

private void CreatePassword()

{

lenght = 8;

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

{

Password += (char)rand.Next(0x0021, 0x007A);

}

}

}

}

**AddForm.cs** – форма добавления в бд

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 AISCarwash

{

public partial class AddForm : Form

{

private string \_mode;

public AddForm(string mode)

{

InitializeComponent();

\_mode = mode;

this.CenterToScreen();

}

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

{

this.Close();

MainForm form = new MainForm(\_mode);

form.Show();

}

private string GetCurrentTable(int SelectedIndex)

{

string tableName = "";

switch (SelectedIndex)

{

case 0: tableName = "ListWashers"; break;

case 1: tableName = "ListClients"; break;

case 2: tableName = "listServices"; break;

default:

break;

}

return tableName;

}

private string[] GetCurrentNameColumns(int SelectedIndex)

{

string[] NameColumns = new string[] { };

switch (SelectedIndex)

{

case 0: NameColumns = new string[] { "id", "Имя", "Фамилия", "Отчество", "Дата рождения" }; break;

case 1: NameColumns = new string[] { "id", "Имя", "Фамилия", "Отчество", "Модель машины" }; break;

case 2: NameColumns = new string[] { "id", "Наименование улуги", "Цена" }; break;

default:

break;

}

return NameColumns;

}

private void UpdateGridView()

{

string tableName = GetCurrentTable(tabControl.SelectedIndex);

string column = "\*";

string condition = $"EXISTS (SELECT {column} FROM {tableName})";

dataGridView.DataSource = MySqlConnecter.QueryReturnTable(column, tableName, condition);

dataGridView = MySqlConnecter.ChangeColumnsName(dataGridView, GetCurrentNameColumns(tabControl.SelectedIndex));

counterTable.Text = dataGridView.RowCount.ToString();

}

private void tabControl\_SelectedIndexChanged(object sender, EventArgs e)

{

UpdateGridView();

}

private void ResetAll()

{

textNameWasher.ResetText();

textSurnameWasher.ResetText();

textPatronymicWasher.ResetText();

dateOBWasher.ResetText();

textNameClient.ResetText();

textSurnameClient.ResetText();

textPatronymicClient.ResetText();

textModelCarClient.ResetText();

textNameService.ResetText();

textPriceService.ResetText();

}

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

{

if (MySqlConnecter.StringsIsEmpty(textNameWasher.Text, textSurnameWasher.Text))

return;

string table = GetCurrentTable(tabControl.SelectedIndex);

string values = $"DEFAULT, '{textNameWasher.Text}', '{textSurnameWasher.Text}', '{textPatronymicWasher.Text}', '{dateOBWasher.Value.ToString("yyyy-MM-dd")}'";

MySqlConnecter.QueryAddInTable(table, values);

UpdateGridView();

ResetAll();

}

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

{

if (MySqlConnecter.StringsIsEmpty(textNameClient.Text, textSurnameClient.Text, textModelCarClient.Text))

return;

string table = GetCurrentTable(tabControl.SelectedIndex);

string values = $"DEFAULT, '{textNameClient.Text}', '{textSurnameClient.Text}', '{textPatronymicClient.Text}', '{textModelCarClient.Text}'";

MySqlConnecter.QueryAddInTable(table, values);

UpdateGridView();

ResetAll();

}

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

{

if (MySqlConnecter.StringsIsEmpty(textNameService.Text))

return;

string table = GetCurrentTable(tabControl.SelectedIndex);

string values = $"DEFAULT, '{textNameService.Text}', {textPriceService.Text}";

MySqlConnecter.QueryAddInTable(table, values);

UpdateGridView();

ResetAll();

}

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

{

DateTime Now = DateTime.Today;

dateOBWasher.MaxDate = new DateTime(Now.Year - 14, Now.Month, Now.Day);

UpdateGridView();

}

}

}

**ChangeForm.cs** – форма редактирования и удаления в бд

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 AISCarwash

{

public partial class ChangeForm : Form

{

private string \_mode;

private DataGridViewRow \_currentRow;

public ChangeForm(string mode)

{

\_mode = mode;

InitializeComponent();

this.CenterToScreen();

}

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

{

this.Close();

MainForm form = new MainForm(\_mode);

form.Show();

}

private string GetCurrentTable(int SelectedIndex)

{

string tableName = "";

switch (SelectedIndex)

{

case 0: tableName = "ListWashers"; break;

case 1: tableName = "ListClients"; break;

case 2: tableName = "listServices"; break;

default:

break;

}

return tableName;

}

private string[] GetCurrentNameColumns(int SelectedIndex)

{

string[] NameColumns = new string[] {};

switch (SelectedIndex)

{

case 0: NameColumns = new string[] { "id", "Имя", "Фамилия", "Отчество", "Дата рождения" }; break;

case 1: NameColumns = new string[] { "id", "Имя", "Фамилия", "Отчество", "Модель машины" }; break;

case 2: NameColumns = new string[] { "id", "Наименование улуги", "Цена" }; break;

default:

break;

}

return NameColumns;

}

private void UpdateGridView()

{

string tableName = GetCurrentTable(tabControl.SelectedIndex);

string column = "\*";

string condition = $"EXISTS (SELECT {column} FROM {tableName})";

dataGridView.DataSource = MySqlConnecter.QueryReturnTable(column, tableName, condition);

dataGridView = MySqlConnecter.ChangeColumnsName(dataGridView, GetCurrentNameColumns(tabControl.SelectedIndex));

counterTable.Text = dataGridView.RowCount.ToString();

}

private void tabControl\_SelectedIndexChanged(object sender, EventArgs e)

{

UpdateGridView();

}

private void SelectCurrentRow()

{

\_currentRow = dataGridView.Rows[dataGridView.CurrentCell.RowIndex];

\_currentRow.Selected = true;

buttonChangeClient.Enabled = true;

buttonChangeService.Enabled = true;

buttonChangeWasher.Enabled = true;

buttonDelClient.Enabled = true;

buttonDelService.Enabled = true;

buttonDelWasher.Enabled = true;

}

private void dataGridView\_CellClick(object sender, DataGridViewCellEventArgs e)

{

SelectCurrentRow();

switch (tabControl.SelectedIndex)

{

case 0: GetDataFromTableListWashers(); break;

case 1: GetDataFromTableListClients(); break;

case 2: GetDataFromTableListService(); break;

default:

break;

}

}

private void ResetAll()

{

textNameWasher.ResetText();

textSurnameWasher.ResetText();

textPatronymicWasher.ResetText();

dateOBWasher.ResetText();

textNameClient.ResetText();

textSurnameClient.ResetText();

textPatronymicClient.ResetText();

textModelCarClient.ResetText();

textNameService.ResetText();

textPriceService.ResetText();

}

private void GetDataFromTableListWashers()

{

textNameWasher.Text = \_currentRow.Cells[1].Value.ToString();

textSurnameWasher.Text = \_currentRow.Cells[2].Value.ToString();

textPatronymicWasher.Text = \_currentRow.Cells[3].Value.ToString();

dateOBWasher.Text = \_currentRow.Cells[4].Value.ToString();

}

private void GetDataFromTableListClients()

{

textNameClient.Text = \_currentRow.Cells[1].Value.ToString();

textSurnameClient.Text = \_currentRow.Cells[2].Value.ToString();

textPatronymicClient.Text = \_currentRow.Cells[3].Value.ToString();

textModelCarClient.Text = \_currentRow.Cells[4].Value.ToString();

}

private void GetDataFromTableListService()

{

textNameService.Text = \_currentRow.Cells[1].Value.ToString();

textPriceService.Text = \_currentRow.Cells[2].Value.ToString();

}

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

{

if (MySqlConnecter.StringsIsEmpty(textNameWasher.Text, textSurnameWasher.Text))

return;

string table = GetCurrentTable(tabControl.SelectedIndex);

string values = $" {\_currentRow.Cells[1].OwningColumn.Name} = '{textNameWasher.Text}', " +

$" {\_currentRow.Cells[2].OwningColumn.Name} = '{textSurnameWasher.Text}', " +

$" {\_currentRow.Cells[3].OwningColumn.Name} = '{textPatronymicWasher.Text}', " +

$" {\_currentRow.Cells[4].OwningColumn.Name} = '{dateOBWasher.Value.ToString("yyyy-MM-dd")}'";

string condition = $"{\_currentRow.Cells[0].OwningColumn.Name} = {\_currentRow.Cells[0].Value.ToString()}";

MySqlConnecter.QueryChangeInTable(table, values, condition);

UpdateGridView();

ResetAll();

}

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

{

if (MySqlConnecter.StringsIsEmpty(textNameClient.Text, textSurnameClient.Text, textModelCarClient.Text))

return;

string table = GetCurrentTable(tabControl.SelectedIndex);

string values = $" {\_currentRow.Cells[1].OwningColumn.Name} = '{textNameClient.Text}', " +

$" {\_currentRow.Cells[2].OwningColumn.Name} = '{textSurnameClient.Text}', " +

$" {\_currentRow.Cells[3].OwningColumn.Name} = '{textPatronymicClient.Text}', " +

$" {\_currentRow.Cells[4].OwningColumn.Name} = '{textModelCarClient.Text}'";

string condition = $"{\_currentRow.Cells[0].OwningColumn.Name} = {\_currentRow.Cells[0].Value.ToString()}";

MySqlConnecter.QueryChangeInTable(table, values, condition);

UpdateGridView();

ResetAll();

}

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

{

if (MySqlConnecter.StringsIsEmpty(textNameService.Text))

return;

string table = GetCurrentTable(tabControl.SelectedIndex);

string values = $" {\_currentRow.Cells[1].OwningColumn.Name} = '{textNameService.Text}', " +

$" {\_currentRow.Cells[2].OwningColumn.Name} = {textPriceService.Text}";

string condition = $"{\_currentRow.Cells[0].OwningColumn.Name} = {\_currentRow.Cells[0].Value.ToString()}";

MySqlConnecter.QueryChangeInTable(table, values, condition);

UpdateGridView();

ResetAll();

}

private void DeleteCurrentRow()

{

string table = GetCurrentTable(tabControl.SelectedIndex);

string condition = $"{\_currentRow.Cells[0].OwningColumn.Name} = {\_currentRow.Cells[0].Value.ToString()}";

MySqlConnecter.QueryDeleteInTable(table, condition);

UpdateGridView();

ResetAll();

}

private void WarningDeleteCurrentRow()

{

string row = "";

for (int i = 0; i < dataGridView.Columns.Count; i++) row += "\t" + \_currentRow.Cells[i].Value.ToString();

DialogResult dialogResult = MessageBox.Show($"Вы действительно хотите удалить следующую запись?: \n{row}", "Внимание! УДАЛЕНИЕ!", MessageBoxButtons.YesNo, MessageBoxIcon.Asterisk);

if (dialogResult == DialogResult.Yes)

DeleteCurrentRow();

}

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

{

WarningDeleteCurrentRow();

}

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

{

WarningDeleteCurrentRow();

}

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

{

WarningDeleteCurrentRow();

}

private void dataGridView\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

SelectCurrentRow();

WarningDeleteCurrentRow();

}

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

{

dateOBWasher.MaxDate = DateTime.Today;

UpdateGridView();

}

}

}

**MySqlConnecter.cs** –класс для работы с sql запросами

using Microsoft.SqlServer.Server;

using MySql.Data.MySqlClient;

using MySqlX.XDevAPI.Relational;

using System;

using System.Collections;

using System.Collections.Generic;

using System.Data;

using System.Data.Common;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Web;

using System.Windows.Forms;

namespace AISCarwash

{

static class MySqlConnecter

{

private static string \_connectionString = "host='localhost';database='carwah\_shema';uid='root';pwd='';charset=utf8;";

private static MySqlConnection \_dbConection = new MySqlConnection(\_connectionString);

public static DataTable QueryReturnTable(string column, string tableName, string condition)

{

DataTable table = new DataTable();

string query = $"SELECT {column} FROM {tableName} WHERE {condition};";

try

{

\_dbConection.Open();

MySqlCommand command = new MySqlCommand(query, \_dbConection);

MySqlDataAdapter dataAdapter = new MySqlDataAdapter(command);

dataAdapter.Fill(table);

\_dbConection.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message + "\n" + query, "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

return table;

}

public static void QueryAddInTable(string tableName, string values)

{

string query = $"INSERT INTO {tableName} VALUES ({values});";

try

{

\_dbConection.Open();

MySqlCommand command = new MySqlCommand(query, \_dbConection);

command.ExecuteNonQuery();

\_dbConection.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message + "\n" + query, "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

}

public static void QueryChangeInTable(string tableName, string values, string condition)

{

string query = $"UPDATE {tableName} SET {values} WHERE {condition};";

try

{

\_dbConection.Open();

MySqlCommand command = new MySqlCommand(query, \_dbConection);

command.ExecuteNonQuery();

\_dbConection.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message + "\n" + query, "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

}

public static void QueryDeleteInTable(string table, string condition)

{

string query = $"DELETE FROM {table} WHERE {condition};";

try

{

\_dbConection.Open();

MySqlCommand command = new MySqlCommand(query, \_dbConection);

command.ExecuteNonQuery();

\_dbConection.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message + "\n" + query, "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

}

public static DataGridView ChangeColumnsName(DataGridView dgv, params string[] Names)

{

int lenght;

if (dgv.ColumnCount < Names.Length)

lenght = dgv.ColumnCount;

else

lenght = Names.Length;

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

dgv.Columns[i].HeaderText = Names[i];

return dgv;

}

public static bool StringsIsEmpty(params string[] parameters)

{

for (int i = 0; i < parameters.Length; i++)

{

if (parameters[i] == string.Empty)

{

MessageBox.Show("Слишком мало данных для формирования записи", "EROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

return true;

}

}

return false;

}

}

}

**ProvisionServies.cs** –форма для оказания услуг

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 AISCarwash

{

public partial class ProvisionServies : Form

{

private string \_mode;

string[] idClients;

string[] idServices;

string[] idWashers;

public ProvisionServies(string mode)

{

InitializeComponent();

\_mode = mode;

this.CenterToScreen();

}

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

{

groupAddClient.Enabled = selectAddClient.Checked;

}

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

{

this.Close();

MainForm form = new MainForm(\_mode);

form.Show();

}

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

{

if (MySqlConnecter.StringsIsEmpty(textNameClient.Text, textSurnameClient.Text, textModelCarClient.Text))

return;

string table = "listclients";

string values = $"DEFAULT, '{textNameClient.Text}', '{textSurnameClient.Text}', '{textPatronymicClient.Text}', '{textModelCarClient.Text}'";

MySqlConnecter.QueryAddInTable(table, values);

selectAddClient.Checked = false;

UpdateBoxes();

ResetAddClient();

}

private void ResetAddClient()

{

textNameClient.ResetText();

textSurnameClient.ResetText();

textPatronymicClient.ResetText();

textModelCarClient.ResetText();

}

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

{

UpdateBoxes();

}

private void UpdateBoxes()

{

selectClient.DropDownStyle = ComboBoxStyle.DropDownList;

selectClient.DataSource = ParseData("\*", "listClients");

idClients = ParseData("idClient", "listClients");

selectClient.SelectedIndex = selectClient.Items.Count - 1;

selectService.DropDownStyle = ComboBoxStyle.DropDownList;

selectService.DataSource = ParseData("\*", "listServices");

idServices = ParseData("idService", "listServices");

selectService.SelectedIndex = selectService.Items.Count - 1;

selectWasher.DropDownStyle = ComboBoxStyle.DropDownList;

selectWasher.DataSource = ParseData("\*", "listWashers");

idWashers = ParseData("idWasher", "listWashers");

selectWasher.SelectedIndex = selectWasher.Items.Count - 1;

}

private string[] ParseData(string column, string tableName)

{

string condition = $"EXISTS (SELECT {column} FROM {tableName})";

DataTable dt = MySqlConnecter.QueryReturnTable(column, tableName, condition);

string[] data = new string[dt.Rows.Count];

for (int i = 0; i < dt.Rows.Count; i++)

{

for (int j = 0; j < dt.Columns.Count; j++)

data[i] += dt.Rows[i][j].ToString() + " \t ";

}

return data;

}

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

{

if (MySqlConnecter.StringsIsEmpty(selectClient.Text, selectService.Text, selectWasher.Text))

return;

string tableName = "provisionService";

string values = $"DEFAULT, {idClients[selectClient.SelectedIndex]}, {idServices[selectService.SelectedIndex]}, {idWashers[selectWasher.SelectedIndex]}";

MySqlConnecter.QueryAddInTable(tableName, values);

MessageBox.Show("проводка завершена успешно");

BackToolStripMenuItem\_Click(sender, e);

}

}

}

**ViewForm.cs** – форма просмотра оказаных услуг

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 AISCarwash

{

public partial class ViewForm : Form

{

private string \_mode;

private string table = "provisionService";

private DataGridViewRow \_currentRow;

public ViewForm(string mode)

{

InitializeComponent();

\_mode = mode;

this.CenterToScreen();

}

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

{

this.Close();

MainForm form = new MainForm(\_mode);

form.Show();

}

private void UpdateGridView()

{

string column =

"ProvisionService.idProvisionService, " +

"listClients.Surname, " +

"listClients.Name, " +

"listClients.ModelCar, " +

"listServices.ServiceName, " +

"listServices.ServicePrice, " +

"listWashers.Surname, " +

"listWashers.Name";

string tableName = "ProvisionService, listClients, listServices, listWashers";

string condition =

"ProvisionService.idClient = listClients.idClient AND " +

"ProvisionService.idService = listServices.idService AND " +

"ProvisionService.idWasher = listWashers.idWasher";

dataGridView.DataSource = MySqlConnecter.QueryReturnTable(column, tableName, condition);

counterTable.Text = dataGridView.RowCount.ToString();

dataGridView = MySqlConnecter.ChangeColumnsName(dataGridView,

"id", "Фамилия клиента", "Имя клиента",

"Модель машины", "Наименование услуги",

"Цена", "Фамилия мойщика", "Имя мойщика");

}

private void DeleteCurrentRow()

{

string condition = $"{\_currentRow.Cells[0].OwningColumn.Name} = {\_currentRow.Cells[0].Value.ToString()}";

MySqlConnecter.QueryDeleteInTable(table, condition);

UpdateGridView();

}

private void WarningDeleteCurrentRow()

{

string row = "";

for (int i = 0; i < dataGridView.Columns.Count; i++) row += "\t" + \_currentRow.Cells[i].Value.ToString();

DialogResult dialogResult = MessageBox.Show($"Вы действительно хотите удалить следующую запись?: \n{row}", "Внимание! УДАЛЕНИЕ!", MessageBoxButtons.YesNo, MessageBoxIcon.Asterisk);

if (dialogResult == DialogResult.Yes)

DeleteCurrentRow();

}

private void dataGridView\_CellClick(object sender, DataGridViewCellEventArgs e)

{

\_currentRow = dataGridView.Rows[dataGridView.CurrentCell.RowIndex];

\_currentRow.Selected = true;

}

private void dataGridView\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

\_currentRow = dataGridView.Rows[dataGridView.CurrentCell.RowIndex];

\_currentRow.Selected = true;

WarningDeleteCurrentRow();

}

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

{

UpdateGridView();

}

}

}

**Program.cs** – точка входа в программу

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AISCarwash

{

internal static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new AutorizationForm());

}

}

}