**ПРИЛОЖЕНИЕ Д. КОД ПРОГРАММЫ**

**АННОТАЦИЯ**

В данном разделе представлены описания модулей приложения, код программы со встроенными комментариями модулей, а также код запроса БД.

СОДЕРЖАНИЕ

[1. ВВЕДЕНИЕ 3](#_Toc159326143)

[1.1. Описание модулей 3](#_Toc159326144)

[1.2. AdminWindow 3](#_Toc159326145)

[1.3. AnalyticsPage 4](#_Toc159326146)

[1.4. AuthorizationWindow 5](#_Toc159326147)

[1.5. ClientsReqPage 6](#_Toc159326148)

[1.6. ClientWindow 9](#_Toc159326149)

[1.7. HistroryPage 10](#_Toc159326150)

[1.8. MainWindow 11](#_Toc159326151)

[1.9. ModerWindow 12](#_Toc159326152)

[1.10. RegistrationWindow 13](#_Toc159326153)

[1.11. Код запроса БД 14](#_Toc159326154)

# ВВЕДЕНИЕ

## Описание модулей

Модули программы и их описание в алфавитном порядке:

1. AdminWindow – Главное окно администратора
2. AnalyticsPage – Страница с таблицами аналитики запросов
3. AuthorizationWindow – Окно авторизации
4. ClientsReqPage – Страница с таблицой запросов и полями для ввода
5. ClientWindow – Главное окно клиента
6. HistoryPage – Страница с историей запросов от текущего пользователя
7. MainWindow – Стартовое окно программы с переходом на Авторизацию и Регистрацию
8. ModerWindow – Главное окно модератора
9. RegistrationWindow – Окно регистрации клиента

## AdminWindow

using System;

using System.Collections.Generic;

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.Shapes;

namespace NidzhatUP

{

/// <summary>

/// Логика взаимодействия для AdminWindow.xaml

/// </summary>

public partial class AdminWindow : Window

{

int id;

/// <summary>

/// Конструктор класса окна Администратора. Устанавливает отображение страницы и сохраняет id.

/// </summary>

/// <param name="admin\_id">Собственное id из окна авторизации</param>

public AdminWindow(int admin\_id)

{

InitializeComponent();

ClientsReqPage clientsReqPage = new ClientsReqPage();

Frame.Content = clientsReqPage;

id = admin\_id;

}

/// <summary>

/// Переключает отображение фрейма на страницу истории.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

HistoryPage historyPage = new HistoryPage(id);

Frame.Content = historyPage;

}

/// <summary>

/// Переключает отображение фрейма на страницу аналитики.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

AnalyticsPage analyticsPage = new AnalyticsPage();

Frame.Content = analyticsPage;

}

/// <summary>

/// Переключает отображение фрейма на страницу запросов.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

ClientsReqPage clientsReqPage = new ClientsReqPage();

Frame.Content = clientsReqPage;

}

}

}

## AnalyticsPage

using System;

using System.Collections.Generic;

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;

using NidzhatUP.nidzhatDataSetTableAdapters;

namespace NidzhatUP

{

public partial class AnalyticsPage : Page

{

client\_requestTableAdapter clientRequestTableAdapter = new client\_requestTableAdapter();

/// <summary>

/// Заполняет таблицу данными из БД, и устанаваливает надписи текстовым полям.

/// </summary>

public AnalyticsPage()

{

InitializeComponent();

DoneTable.ItemsSource = clientRequestTableAdapter.GetDataByStatus1();

NotDoneTable.ItemsSource = clientRequestTableAdapter.GetDataByStatus2();

DoneText.Text = $"Решено: {DoneTable.Items.Count-1}";

NotDoneText.Text = $"Не решено: {NotDoneTable.Items.Count-1}";

}

}

}

## AuthorizationWindow

using System;

using System.Data;

using System.Collections.Generic;

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.Shapes;

using NidzhatUP.nidzhatDataSetTableAdapters;

namespace NidzhatUP

{

/// <summary>

/// Логика взаимодействия для Authorization.xaml

/// </summary>

public partial class AuthorizationWindow : Window

{

usersTableAdapter usersTableAdapter = new usersTableAdapter();

/// <summary>

/// Обычный конструктор окна.

/// </summary>

public AuthorizationWindow()

{

InitializeComponent();

}

/// <summary>

/// Выполнение авторизации по сопоставлению введенных данных и данных из БД.

/// В зависимости от роли открывает то или иное окно.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

string login = Login.Text;

string password = Password.Text;

bool reg = false;

DataRowCollection AllLogins = usersTableAdapter.GetData().Rows;

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

{

if (AllLogins[i][1].ToString() == login && AllLogins[i][2].ToString() == password)

{

reg = true;

switch (AllLogins[i][3])

{

case 1:

AdminWindow adminWindow = new AdminWindow(Convert.ToInt32(AllLogins[i][0]));

adminWindow.Show();

Close();

break;

case 2:

ModerWindow moderWindow = new ModerWindow();

moderWindow.Show();

Close();

break;

case 3:

ClientWindow clientWindow = new ClientWindow(Convert.ToInt32(AllLogins[i][0]));

clientWindow.Show();

Close();

break;

default:

MessageBox.Show("Произошел сбой в системе или функционал для вашей роли еще не разработан.");

break;

}

}

}

if (!reg) MessageBox.Show("Неверный логин или пароль!");

}

}

}

## ClientsReqPage

using System;

using System.Collections.Generic;

using System.Data;

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;

using NidzhatUP.nidzhatDataSetTableAdapters;

namespace NidzhatUP

{

/// <summary>

/// Логика взаимодействия для ClientsReqPage.xaml

/// </summary>

public partial class ClientsReqPage : Page

{

client\_requestTableAdapter crta = new client\_requestTableAdapter();

request\_statusTableAdapter rsta = new request\_statusTableAdapter();

request\_historyTableAdapter rhta = new request\_historyTableAdapter();

/// <summary>

/// Конструктор окна с получением данных из БД.

/// </summary>

public ClientsReqPage()

{

InitializeComponent();

Table.ItemsSource = crta.GetData();

Status.ItemsSource = rsta.GetData();

Status.DisplayMemberPath = "request\_status";

}

/// <summary>

/// Очистка текстового поля, отвечающего за текст запроса.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

private void Clear\_Click\_Text(object sender, RoutedEventArgs e)

{

Text.Clear();

}

/// <summary>

/// Очистка текстового поля, отвечающего за id клиента.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

private void Clear\_Click\_Client(object sender, RoutedEventArgs e)

{

Client.Clear();

}

/// <summary>

/// Изменение выделенного объекта в таблице на данные, указанные в текстовых полях.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

if (Table.SelectedIndex != -1)

{

try

{

crta.UpdateQuery(Convert.ToInt32(Client.Text), Convert.ToInt32((Status.SelectedItem as DataRowView).Row[0]), Text.Text, Convert.ToInt32(Moderator.Text),

Convert.ToInt32((Table.SelectedItem as DataRowView).Row[4]));

Refresh();

}

catch (Exception)

{

MessageBox.Show("Неверно введены данные! Данные должны быть другого типа.");

}

}

}

/// <summary>

/// Удаление из БД выделенного объекта таблицы.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

try

{

if (Table.SelectedIndex != -1)

{

crta.DeleteQuery(Convert.ToInt32((Table.SelectedItem as DataRowView).Row[4]));

Refresh();

}

}

catch (Exception)

{

MessageBox.Show($"Непредиденная ошибка!");

throw;

}

}

/// <summary>

/// Добавление нового объекта в БД с данными, введенными в текстовых полях.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

try

{

int statusId = 0;

if (Status.SelectedItem != null)

{

DataRowView selectedStatus = (DataRowView)Status.SelectedItem;

statusId = Convert.ToInt32(selectedStatus.Row[0]);

}

crta.InsertQuery(Convert.ToInt32(Client.Text), statusId, Text.Text, Convert.ToInt32(Moderator.Text));

rhta.InsertQuery(Convert.ToInt32(Client.Text), (int)crta.GetMaxId());

Refresh();

}

catch (Exception)

{

MessageBox.Show("Неверно введены данные! Данные должны быть другого типа.");

}

}

/// <summary>

/// Очистка текстового поля, отвечающего за id модератора.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

private void Clear\_Click\_Moderator(object sender, RoutedEventArgs e)

{

Moderator.Clear();

}

/// <summary>

/// Событие, вызывающееся при выделении какого-либо объекта таблицы(DataGrid).

/// Заполняет текстовые поля и выпадающий список в соответствии с данными выделенного объекта

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

if (Table.SelectedIndex != -1)

{

try

{

Client.Text = Convert.ToString((Table.SelectedItem as DataRowView).Row[0]);

int statusId = Convert.ToInt32((Table.SelectedItem as DataRowView).Row[1]);

foreach (DataRowView item in Status.Items)

{

if (Convert.ToInt32(item.Row[0]) == statusId)

{

Status.SelectedItem = item;

break;

}

}

Text.Text = Convert.ToString((Table.SelectedItem as DataRowView).Row[2]);

Moderator.Text = Convert.ToString((Table.SelectedItem as DataRowView).Row[3]);

}

catch (Exception)

{

MessageBox.Show("Выбран неверный элемент!");

}

}

}

/// <summary>

/// Данный метод обновляет содержание таблицы, подгружая новые данные из БД.

/// </summary>

private void Refresh()

{

Table.ItemsSource = null;

Table.ItemsSource = crta.GetData();

}

}

}

## ClientWindow

using System;

using System.Collections.Generic;

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.Shapes;

namespace NidzhatUP

{

/// <summary>

/// Логика взаимодействия для ClientWindow.xaml

/// </summary>

public partial class ClientWindow : Window

{

int id;

/// <summary>

/// Конструктор окна клиента, устанавливающий отображение контента и удаляющий из него кнопки удаления и изменения.

/// </summary>

/// <param name="client\_id">Собственный id, полученный из окна авторизации</param>

public ClientWindow(int client\_id)

{

InitializeComponent();

ClientsReqPage clientsReqPage = new ClientsReqPage();

clientsReqPage.Change.Visibility = Visibility.Collapsed;

clientsReqPage.Delete.Visibility = Visibility.Collapsed;

Frame.Content = clientsReqPage;

id = client\_id;

}

/// <summary>

/// Меняет отображение на страницу запросов и удаляет из нее кнопки удаления и изменения.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

ClientsReqPage clientsReqPage = new ClientsReqPage();

clientsReqPage.Change.Visibility = Visibility.Collapsed;

clientsReqPage.Delete.Visibility = Visibility.Collapsed;

Frame.Content = clientsReqPage;

}

/// <summary>

/// Меняет отображение на страницу запросов и удаляет из нее кнопки удаления и изменения.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

HistoryPage historyPage = new HistoryPage(id);

Frame.Content = historyPage;

}

}

}

## HistroryPage

using System;

using System.Collections.Generic;

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;

using NidzhatUP.nidzhatDataSetTableAdapters;

namespace NidzhatUP

{

/// <summary>

/// Логика взаимодействия для HistoryPage.xaml

/// </summary>

public partial class HistoryPage : Page

{

request\_historyTableAdapter requestHistoryTableAdapter = new request\_historyTableAdapter();

/// <summary>

/// Конструктор, получающий данные из БД и устанавливающий их отображение в таблице.

/// </summary>

/// <param name="id"></param>

public HistoryPage(int id)

{

InitializeComponent();

Table.ItemsSource = requestHistoryTableAdapter.GetDataBy1(id);

}

}

}

## MainWindow

using System;

using System.Collections.Generic;

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 NidzhatUP

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

/// <summary>

/// Конструктор окна, устанавливающий отображение визуальных элементов

/// </summary>

public MainWindow()

{

InitializeComponent();

}

/// <summary>

/// Открывает окно регистрации, закрывая текущее.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

RegistrationWindow window = new RegistrationWindow();

window.Show();

Close();

}

/// <summary>

/// Открывает окно авторизации, закрывая текущее

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

AuthorizationWindow window = new AuthorizationWindow();

window.Show();

Close();

}

}

}

## ModerWindow

using System;

using System.Collections.Generic;

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.Shapes;

namespace NidzhatUP

{

/// <summary>

/// Логика взаимодействия для ModerWindow.xaml

/// </summary>

public partial class ModerWindow : Window

{

/// <summary>

/// Конструктор окна модератора, устанавливающий отображение контента и удаляющий из него кнопку изменения.

/// </summary>

public ModerWindow()

{

InitializeComponent();

ClientsReqPage clientsReqPage = new ClientsReqPage();

clientsReqPage.Change.Visibility = Visibility.Collapsed;

Frame.Content = clientsReqPage;

}

/// <summary>

/// Меняет отображение на страницу запросов и удаляет из нее кнопку изменения.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

ClientsReqPage clientsReqPage = new ClientsReqPage();

clientsReqPage.Change.Visibility = Visibility.Collapsed;

Frame.Content = clientsReqPage;

}

/// <summary>

/// Меняет отображение на страницу аналитики запросов.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

AnalyticsPage analyticsPage = new AnalyticsPage();

Frame.Content = analyticsPage;

}

}

}

## RegistrationWindow

using System;

using System.Collections.Generic;

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.Shapes;

using NidzhatUP.nidzhatDataSetTableAdapters;

namespace NidzhatUP

{

/// <summary>

/// Логика взаимодействия для Registration.xaml

/// </summary>

public partial class RegistrationWindow : Window

{

usersTableAdapter usersTableAdapter = new usersTableAdapter();

/// <summary>

/// Обычный конструктор окна

/// </summary>

public RegistrationWindow()

{

InitializeComponent();

}

/// <summary>

/// Регистрирует нового пользователя в БД с ролью Клиент, используя данные из введенных полей, после чего открывает окно авторизации.

/// </summary>

/// <param name="sender">Объект, откуда вызывался метод</param>

/// <param name="e">Событие, которое вызвало метод</param>

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

{

string login = Login.Text;

string password = Password.Text;

string password2 = Password2.Text;

if (password != password2)

{

MessageBox.Show("Пароли не совпадают!");

return;

}

if (password.Length < 5)

{

MessageBox.Show("Пароль должен состоять из 5-ти или более символов.");

return;

}

if (login.Length < 4)

{

MessageBox.Show("Логин должен состоять из 4-х или более символов.");

return;

}

usersTableAdapter.InsertQuery(login, password, 3);

MessageBox.Show("Вы успешно зарегистрированы!");

MainWindow window = new MainWindow();

window.Show();

Close();

}

}

}

## Код запроса БД

--Create

CREATE TABLE roles (

id INT PRIMARY KEY IDENTITY(1,1),

role\_name VARCHAR(20) NOT NULL

);

CREATE TABLE users (

id INT PRIMARY KEY IDENTITY(1,1),

login VARCHAR(20) NOT NULL,

password VARCHAR(50) NOT NULL,

role\_id INT NOT NULL,

FOREIGN KEY (role\_id) REFERENCES roles(id)

);

CREATE TABLE request\_status (

status\_id INT PRIMARY KEY IDENTITY(1,1),

request\_status VARCHAR(20) NOT NULL

);

CREATE TABLE client\_request (

id INT PRIMARY KEY IDENTITY(1,1),

client\_id INT NOT NULL,

status\_id INT NOT NULL,

request\_text VARCHAR(255) NOT NULL,

moderator\_user\_id INT,

FOREIGN KEY (client\_id) REFERENCES users(id),

FOREIGN KEY (status\_id) REFERENCES request\_status(status\_id),

FOREIGN KEY (moderator\_user\_id) REFERENCES users(id)

);

CREATE TABLE request\_history (

id INT PRIMARY KEY IDENTITY(1,1),

client\_id INT NOT NULL,

request\_id INT NOT NULL,

FOREIGN KEY (client\_id) REFERENCES users(id),

FOREIGN KEY (request\_id) REFERENCES client\_request(id) ON DELETE CASCADE

);

CREATE TABLE request\_analysis (

id INT PRIMARY KEY IDENTITY(1,1),

request\_id INT NOT NULL,

FOREIGN KEY (request\_id) REFERENCES client\_request(id) ON DELETE CASCADE

);

--Insert

insert into roles values ('Администратор');

insert into roles values ('Модератор');

insert into roles values ('Клиент');

insert into users values ('admin', 'admin', 1);

insert into users values ('moder', 'moder', 2);

insert into client\_request values (3, 2, 'Нужен тело. Хранитель.', 2);

insert into request\_status values ('Решено')

insert into request\_status values ('Не решено')

--Select

select \* from roles

select \* from users

select \* from request\_status

select \* from client\_request

select \* from request\_history

select \* from request\_analysis

--Drop

drop table roles

drop table users

drop table request\_status

drop table client\_request

drop table request\_history

drop table request\_analysis