Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

Специальность 09.02.07 «Информационные системы и программирование»

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

Профессиональный модуль ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

Выполнил:

обучающийся учебной группы №\_\_\_619\_\_\_

\_ДеревягинЕгор\_4ИСИП-619

Проверили:

Руководители практики от колледжа:

Сибирев И.В

*(должность)*

Аксёнова Т.Г.

*(должность)*

Костиков П.А.

*(должность)*

Ельшакова Е.Л.

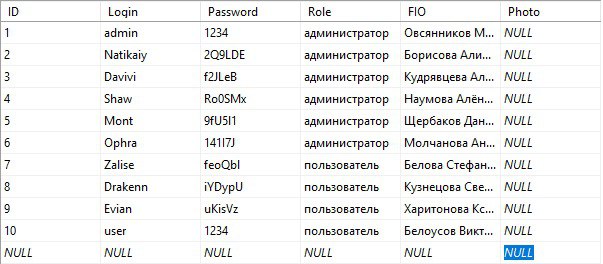
*(должность)*

*(оценка) (подпись)*

**Москва**

**20­­­­22**

Цель работы: приобрести практические навыки проектирования интерфейса приложения WPF, навигации по страницам WPF, подключения базы данных с помощью технологии Entity Framework ADO.NET, реализации страницы авторизации пользователей, создания и настройки словаря стилей.



### Рисунок 1.Таблица БД User

Изображение выглядит как стол

Автоматически созданное описание

### Рисунок 2. Страница Аунтификации

Изображение выглядит как стол

Автоматически созданное описание

### Рисунок 3. Вход на Страницу Администратора

Изображение выглядит как текст

Автоматически созданное описание

### Рисунок 4. Вход на Страницу Администратора (2)

Изображение выглядит как стол

Автоматически созданное описание

### Рисунок 5. Вход на Страницу Пользователя

Изображение выглядит как текст

Автоматически созданное описание

### Рисунок 6. Вход на Страницу Пользователя (2)

## Листинг программы

### MainWindow.xaml

<Window x:Name="MainFrame1" x:Class="\_619\_Voronov\_.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:\_619\_Voronov\_"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="45"/>

<RowDefinition/>

<RowDefinition Height="45"/>

</Grid.RowDefinitions>

<Button Grid.Row="0" Name="Back\_btn" HorizontalAlignment="left" Width="70" Height="35" Margin="10,0,0,0" Click="Back\_btn\_Click">Назад</Button>

<TextBlock Grid.Row="2" Name="DataTimer" HorizontalAlignment="Right" Margin="10,10,10,0"/>

<Frame Grid.Row="1" Name="MainFrame" Source="Pages/AuthPage.xaml" Navigated="MainFrame\_OnNavigated"/>

</Grid>

</Window>

### Administrator.xaml

<Page x:Class="\_619\_Voronov.Pages.Administrator"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:\_619\_Voronov.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Администратор">

<Grid Background="white">

<TextBlock

HorizontalAlignment="Center"

VerticalAlignment="Center"

FontSize="28pt"

FontWeight="Bold"

>Вы на странице администратора</TextBlock>

</Grid>

</Page>

### User.xaml

<Page x:Class="\_619\_Voronov.Pages.Users"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:\_619\_Voronov.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Пользователь">

<Grid Background="white">

<TextBlock

HorizontalAlignment="Center"

VerticalAlignment="Center"

FontSize="28pt"

FontWeight="Bold"

>Вы на странице пользователя</TextBlock>

</Grid>

</Page>

### AuthPage.xaml

<Page x:Class="\_619\_Voronov.Pages.AuthPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:\_619\_Voronov.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Авторизация">

<Grid Background="#b5d5ca">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="34\*"/>

<ColumnDefinition Width="89\*"/>

<ColumnDefinition Width="177\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="52\*"/>

<RowDefinition Height="193\*"/>

<RowDefinition Height="55\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Column="1" Grid.Row="1" HorizontalAlignment="Center" Width="237" Grid.RowSpan="2" Height="290" VerticalAlignment="Top">

<Label Content="Логин"/>

<Label Content="Пароль"/>

</StackPanel>

<TextBlock Name="txtHintLogin" Grid.Column="2" Grid.Row="1" Text="Логин или почта"

Margin="10,20,212,0" Foreground="Gray" FontSize="12pt" Width="250"

Height="30" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<TextBlock Name="txtHintPassword" Grid.Column="2" Grid.Row="1" Text="Пароль"

Margin="10,70" Foreground="Gray" FontSize="12pt" Width="250"

Height="30" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<StackPanel Grid.Row="1" Grid.Column="2" HorizontalAlignment="Left" Grid.RowSpan="2" Width="472" Height="290" VerticalAlignment="Top">

<TextBox x:Name="TextBoxLogin" ToolTip="Введите логин или почту" Background="Transparent" TextChanged="TextBoxLogin\_Changed"/>

<PasswordBox x:Name="PasswordBox" Margin="10" Width="250" Height="30" HorizontalAlignment="Left" PasswordChar="\*" ToolTip="Введите свой пароль"

Background="Transparent" PasswordChanged="Password\_Changed"/>

<Button Content="Вход" Margin="10, 30, 10, 5" Width="250" HorizontalAlignment="Left" FontSize="22pt"

ToolTip="Нажмите, чтобы войти в свой аккаунт" Click="ButtonEnter\_OnClick"/>

<Button Content="Регистрация" Width="250" HorizontalAlignment="Left" FontSize="22pt" ToolTip="Нажмите, чтобы зарегистрировать аккаунт"/>

</StackPanel>

</Grid>

</Page>

### Dictionary.xaml

<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

<Style

TargetType="Button">

<Setter Property="Height" Value="40"/>

<Setter Property="Padding" Value="0"/>

<Setter Property="Margin" Value="10"/>

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="FontSize" Value="12pt"/>

<Setter Property="Background" Value="Gray"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style

TargetType="Label">

<Setter Property="Height" Value="30"/>

<Setter Property="Width" Value="150"/>

<Setter Property="Margin" Value="10"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

<Setter Property="FontSize" Value="18pt"/>

<Setter Property="FontFamily" Value="Tahoma"/>

<Setter Property="Padding" Value="0"/>

</Style>

<Style

TargetType="TextBox">

<Setter Property="FontSize" Value="18pt"/>

<Setter Property="Foreground" Value="Black"/>

<Setter Property="Padding" Value="0"/>

<Setter Property="Margin" Value="10"/>

<Setter Property="Width" Value="250"/>

<Setter Property="Height" Value="30"/>

<Setter Property="HorizontalAlignment" Value="Left"/>

<Setter Property="Background" Value="White"/>

</Style>

<Style

TargetType="PasswordBox">

<Setter Property="FontSize" Value="18pt"/>

<Setter Property="Foreground" Value="Black"/>

<Setter Property="Padding" Value="0"/>

<Setter Property="Margin" Value="10"/>

<Setter Property="Width" Value="250"/>

<Setter Property="Height" Value="30"/>

<Setter Property="HorizontalAlignment" Value="Left"/>

<Setter Property="Background" Value="White"/>

</Style>

</ResourceDictionary>

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

# ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЙ №4-6

**Тема:** РАЗРАБОТКА СТРАНИЦЫ РЕГИСТРАЦИИ ПОЛЬЗОВАТЕЛЕЙ.

ЭКСПОРТ ДАННЫХ ИЗ БАЗЫ ДАННЫХ В ТАБЛИЦУ DATAGRID

Студенты:

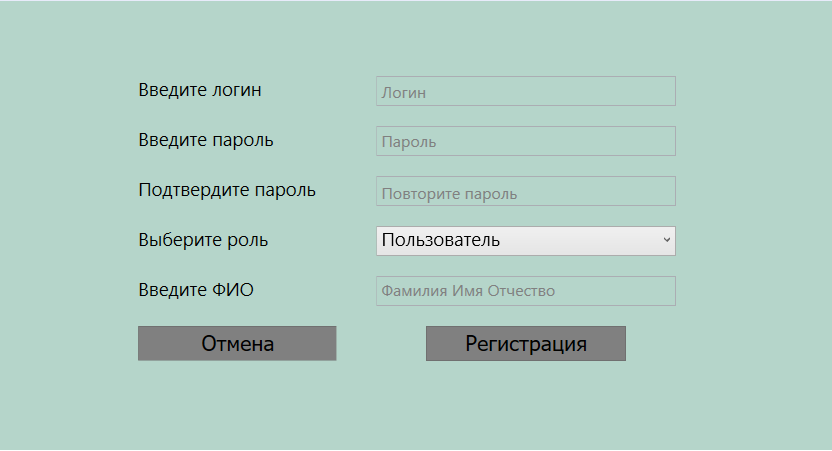
Воронов А.В

Группа: 4ИСИП-619

Преподаватель: Т.Г.Аксёнова

Дата: 12.09.2022

Цель работы: приобрести практические навыки проектирования и реализации страницы регистрации пользователей в приложении WPF, научиться экспортировать данные из базы данных в таблицу DataGrid с помощью технологии ADO.NET Entity Framework.



### Рисунок 1. Создание страницы регистрации.

Изображение выглядит как текст, стол

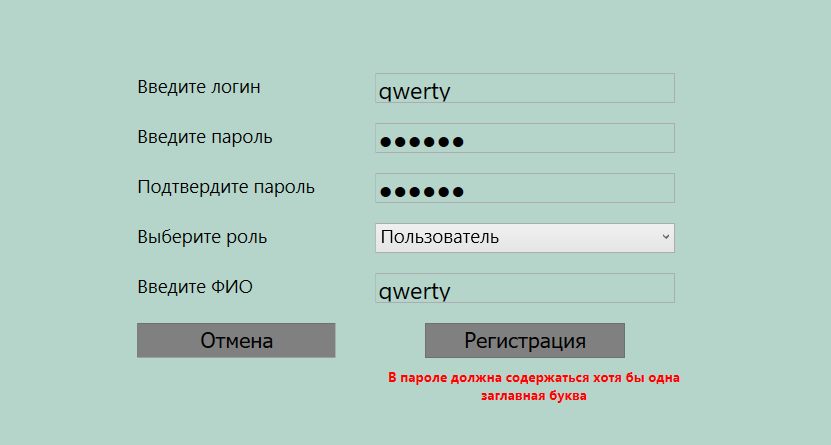
Автоматически созданное описание

### Рисунок 2. Создание выпадающего списка.

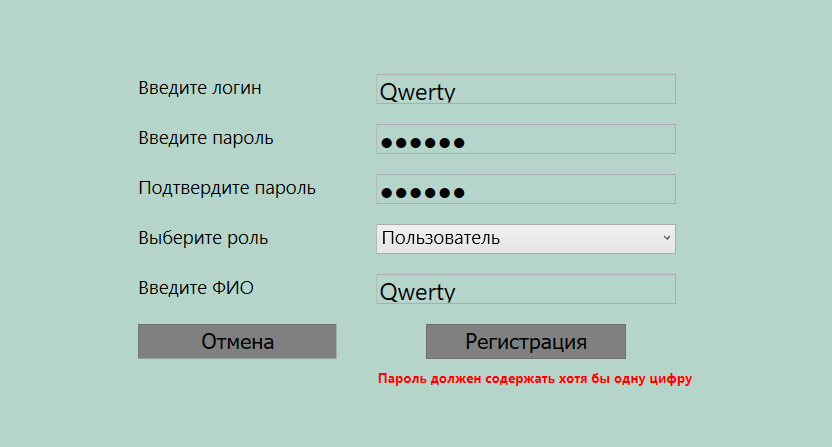
Изображение выглядит как текст

Автоматически созданное описание

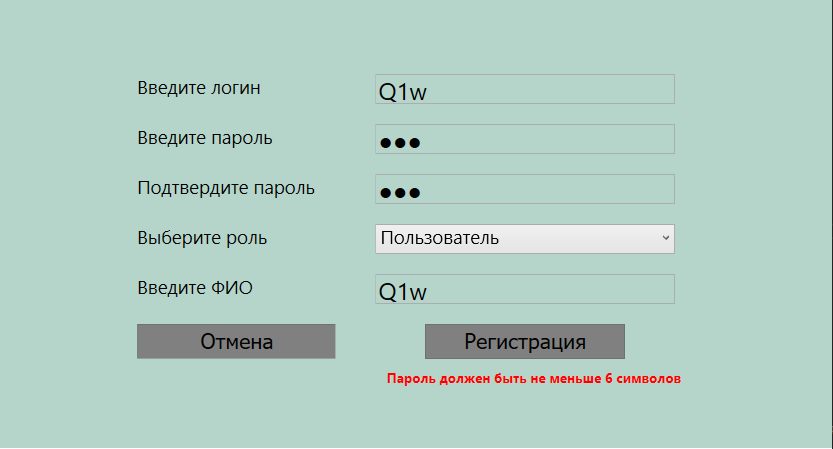
### Рисунок 3. Проверка 1 (Не все поля заполнены).



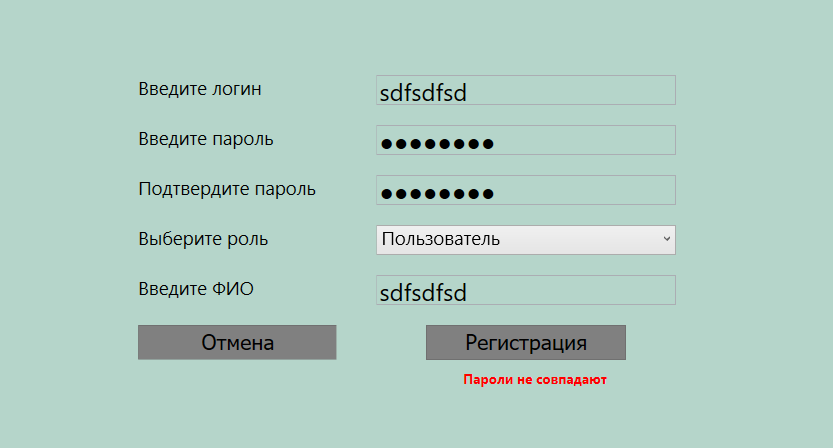
### Рисунок 4. Проверка 2 (Пароль «qwerty»).



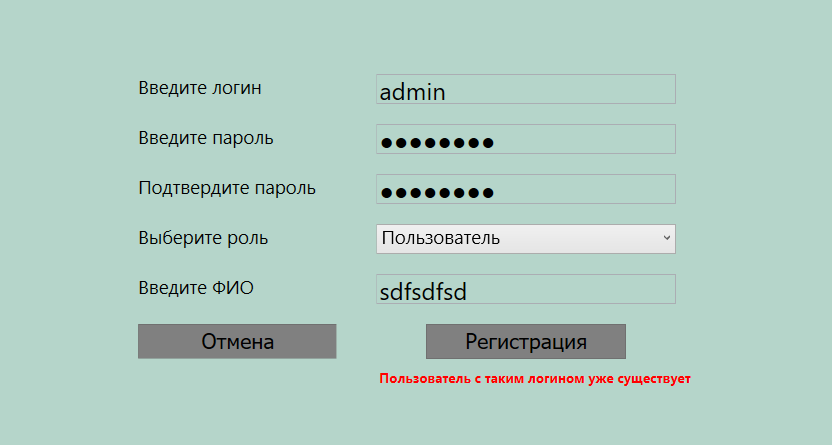
### Рисунок 5. Проверка 3 (Пароль «Qwerty»).



### Рисунок 6. Проверка 4 (Пароль «Q1w»).



### Рисунок 7. Проверка 5 (Пароли не совпадают).



### Рисунок 8. Проверка 6 (Логин уже существует в БД).

Изображение выглядит как стол

Автоматически созданное описание

### Рисунок 9. Страница Администратора.

## **Листинг программы**

### **RegPage.xaml**

<Page x:Class="\_619\_Voronov.Pages.RegPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:\_619\_Voronov.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="RegPage">

<Grid Background="#b5d5ca" Height="450" VerticalAlignment="Center">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="3\*"/>

<ColumnDefinition Width="1\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="45"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="45"/>

</Grid.RowDefinitions>

<StackPanel Grid.Column="1" Grid.Row="1" Margin="20">

<StackPanel.Resources>

<Style

TargetType="TextBlock">

<Setter Property="FontSize" Value="14pt"/>

<Setter Property="Margin" Value="0,10"/>

<Setter Property="Height" Value="30"/>

</Style>

</StackPanel.Resources>

<TextBlock Text="Введите логин" Margin="0, 10" Height="30"/>

<TextBlock Text="Введите пароль"/>

<TextBlock Text="Подтвердите пароль"/>

<TextBlock Text="Выберите роль"/>

<TextBlock Text="Введите ФИО"/>

<Button Name="RegCansel\_btn" Content="Отмена" Height="35" Width="200" HorizontalAlignment="Left" FontSize="16pt" Click="RegCansel\_btn\_Click"/>

</StackPanel>

<TextBlock Text="Логин"

x:Name="txtRegLog"

Style="{StaticResource RegTextBlock}"

Width="220" Height="30" VerticalAlignment="Top" HorizontalAlignment="Left" Margin="25,35"/>

<TextBlock Text="Пароль"

x:Name="txtRegPas"

Style="{StaticResource RegTextBlock}"

Width="220" Height="30" VerticalAlignment="Top" HorizontalAlignment="Left" Margin="25,84"/>

<TextBlock Text="Повторите пароль"

x:Name="txtRegPasRepeat"

Style="{StaticResource RegTextBlock}"

Width="220" Height="30" VerticalAlignment="Top" HorizontalAlignment="Left" Margin="25,136"/>

<TextBlock Text="Фамилия Имя Отчество"

x:Name="txtRegFio"

Style="{StaticResource RegTextBlock}"

Width="220" Height="30" VerticalAlignment="Top" HorizontalAlignment="Left" Margin="25,233,23,0"/>

<StackPanel Grid.Row="1" Grid.Column="2" Margin="20,20,20,38">

<TextBox x:Name="RegLogin" Width="300" Background="Transparent" TextChanged="RegLogin\_TextChanged"/>

<PasswordBox x:Name="RegPassword" Width="300" Background="Transparent" PasswordChanged="RegPassword\_PasswordChanged"/>

<PasswordBox x:Name="RegPasswordRepeat" Width="300" Background="Transparent" PasswordChanged="RegPasswordRepeat\_PasswordChanged"/>

<ComboBox Name="RoleList" Margin="0,10" Width="300" FontSize="14pt">

<ComboBoxItem>

<TextBlock Text="Администратор" FontSize="14pt" Margin="5,0" Width="220"/>

</ComboBoxItem>

<ComboBoxItem IsSelected="True" >

<TextBlock Text="Пользователь" FontSize="14pt" Margin="5,0" Width="220"/>

</ComboBoxItem>

</ComboBox>

<TextBox x:Name="RegFIO" Width="300" Background="Transparent" TextChanged="RegFIO\_TextChanged"/>

<Button x:Name="RegRegister\_btn" Content="Регистрация" Height="35" Width="200"

FontSize="16pt" HorizontalAlignment="Left" Margin="50,10" Click="RegRegister\_btn\_Click"/>

</StackPanel>

<TextBlock x:Name="HidenLable" Text="" Grid.Row="1" Grid.Column="2" FontWeight="Bold"

FontSize="10pt" Foreground="Red" Width="343" Height="70" TextWrapping="Wrap" TextAlignment="Center"

Margin="0,325,0,40" Visibility="Visible" Grid.RowSpan="2"/>

</Grid>

</Page>

### **RegPage.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

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 \_619\_Voronov.Pages

{

/// <summary>

/// Логика взаимодействия для RegPage.xaml

/// </summary>

///

public partial class RegPage : Page

{

public RegPage()

{

InitializeComponent();

}

private void RegLogin\_TextChanged(object sender, TextChangedEventArgs e)

{

txtRegLog.Visibility = Visibility.Visible;

if (RegLogin.Text.Length > 0)

{

txtRegLog.Visibility = Visibility.Hidden;

}

}

private void RegFIO\_TextChanged(object sender, TextChangedEventArgs e)

{

txtRegFio.Visibility = Visibility.Visible;

if (RegFIO.Text.Length > 0)

{

txtRegFio.Visibility = Visibility.Hidden;

}

}

private void RegCansel\_btn\_Click(object sender, RoutedEventArgs e)

{

NavigationService?.Navigate(new AuthPage());

}

private void RegRegister\_btn\_Click(object sender, RoutedEventArgs e)

{

if (string.IsNullOrEmpty(RegLogin.Text) ||

string.IsNullOrEmpty(RegPassword.Password) ||

string.IsNullOrEmpty(RegPasswordRepeat.Password) ||

string.IsNullOrEmpty(RegFIO.Text))

{

HidenLable.Text = "Заполните все поля";

HidenLable.Visibility = Visibility.Visible;

return;

}

using (var db = new Entities\_db())

{

var userCheck = db.USER

.AsNoTracking()

.FirstOrDefault(

u => u.Login == RegLogin.Text);

int scoreUpper = 0;

int scoreNumber = 0;

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

{

if (Char.IsUpper(RegPassword.Password[i]))

scoreUpper++;

if (int.TryParse(RegPassword.Password[i].ToString(), out int n))

scoreNumber++;

if (scoreNumber > 0 && scoreUpper > 0)

break;

}

//уникальность логина

if (userCheck != null)

{

HidenLable.Text = "Пользователь с таким логином уже существует";

HidenLable.Visibility = Visibility.Visible;

return;

}

//порверка на заглавную букву

if (scoreUpper == 0)

{

HidenLable.Text = "В пароле должна содержаться хотя бы одна заглавная буква";

HidenLable.Visibility = Visibility.Visible;

return;

}

if (scoreNumber == 0)

{

HidenLable.Text = "Пароль должен содержать хотя бы одну цифру";

HidenLable.Visibility = Visibility.Visible;

return;

}

//допускается только английская раскладка и содержит хотя бы одну цифру

if (!(Regex.IsMatch(RegPassword.Password, "^[a-zA-Z0-9. -\_?]\*$") || Regex.IsMatch(RegPassword.Password, "^[a-zA-Z0-9]\*$")))

{

HidenLable.Text = "Пароль должен состоять из латинских символов и хотя бы одной цифры";

HidenLable.Visibility = Visibility.Visible;

return;

}

//пароль должен содержать 6 или более символов

if (RegPassword.Password.Length < 6)

{

HidenLable.Text = "Пароль должен быть не меньше 6 символов";

HidenLable.Visibility = Visibility.Visible;

return;

}

//совпадение паролей

if (RegPassword.Password != RegPasswordRepeat.Password)

{

HidenLable.Text = "Пароли не совпадают";

HidenLable.Visibility = Visibility.Visible;

return;

}

if (RoleList.Text is "")

{

RoleList.Text = "Пользователь";

}

Entities\_db DB = new Entities\_db();

USER userObject = new USER()

{

FIO = RegFIO.Text,

Login = RegLogin.Text,

Password = RegPassword.Password,

Role = RoleList.Text

};

DB.USER.Add(userObject);

DB.SaveChanges();

switch (RoleList.Text)

{

case "Администратор":

MessageBox.Show("Регистрация прошла успешно");

NavigationService?.Navigate(new Administrator());

break;

case "Пользователь":

MessageBox.Show("Регистрация прошла успешно");

NavigationService?.Navigate(new Users());

break;

default:

MessageBox.Show("Роли не подошли");

break;

}

RegLogin.Text = "";

RegFIO.Text = "";

HidenLable.Text = "";

}

}

private void RegPassword\_PasswordChanged(object sender, RoutedEventArgs e)

{

txtRegPas.Visibility = Visibility.Visible;

if (RegPassword.Password.Length > 0)

{

txtRegPas.Visibility = Visibility.Hidden;

}

}

private void RegPasswordRepeat\_PasswordChanged(object sender, RoutedEventArgs e)

{

txtRegPasRepeat.Visibility = Visibility.Visible;

if (RegPasswordRepeat.Password.Length > 0)

{

txtRegPasRepeat.Visibility = Visibility.Hidden;

}

}

}

}

### **Administrator.xaml**

<Page x:Class="\_619\_Voronov.Pages.Administrator"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:\_619\_Voronov.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Администратор">

<Grid Background="White">

<Grid.RowDefinitions>

<RowDefinition Height="377"/>

<RowDefinition Height="73\*"/>

</Grid.RowDefinitions>

<DataGrid Name="DataGridUser" Grid.Row="0" AutoGenerateColumns="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn Header="Логин" Binding="{Binding Login}" Width="auto"/>

<DataGridTextColumn Header="Пароль" Binding="{Binding Password}" Width="auto"/>

<DataGridTextColumn Header="Роль" Binding="{Binding Role}" Width="auto"/>

<DataGridTextColumn Header="Ф.И.О" Binding="{Binding FIO}" Width="auto"/>

<DataGridTextColumn Header="Фото" Binding="{Binding Photo}" Width="100"/>

<DataGridTemplateColumn Width="\*">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Name="ButtonEdit" Content="Редактировать" Background="Transparent" BorderThickness="0" Width="auto" Height="20" HorizontalAlignment="Left" Margin="5, 0, 0, 0"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Grid.Row="1" Name="ButtonAdd" Content="Добавить" HorizontalAlignment="Left" Width="70" Height="35"

ToolTip="Нажмите, чтобы добавить пользователя" Margin="5" Click="ButtonAdd\_Click"/>

<Button Grid.Row="1" Name="ButtonDel" Content="Удалить" HorizontalAlignment="Right" Width="70" Height="35"

ToolTip="Нажмите, чтобы удалить пользователя" Margin="5" Click="ButtonDel\_Click"/>

</Grid>

</Page>

### **Administrator.xaml.cs**

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 \_619\_Voronov.Pages

{

/// <summary>

/// Логика взаимодействия для Administrator.xaml

/// </summary>

public partial class Administrator : Page

{

public Administrator()

{

InitializeComponent();

DataGridUser.ItemsSource = Entities\_db.GetContext().USER.ToList();

}

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

{

}

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

{

}

}

}

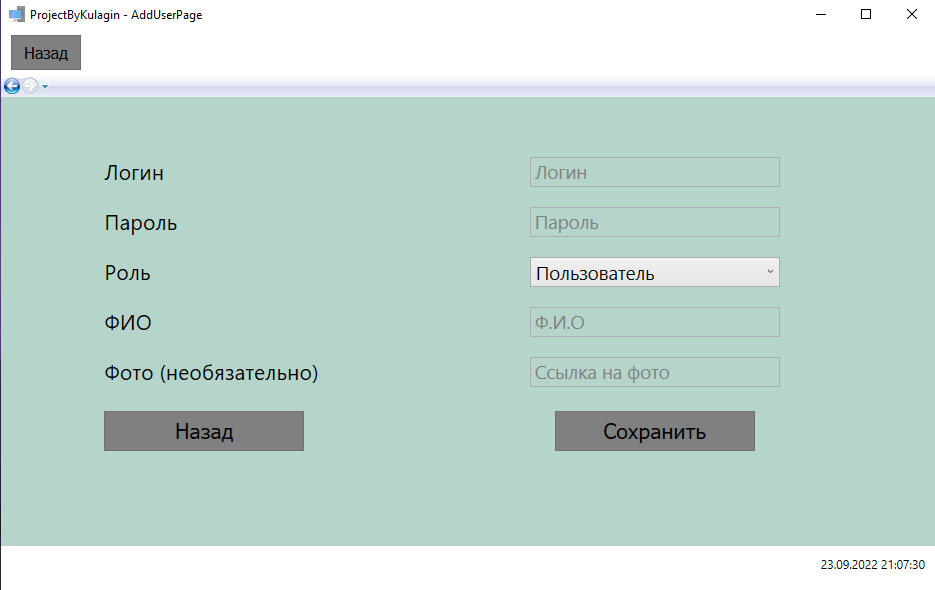


Рисунок 10. Страница AddUserPage

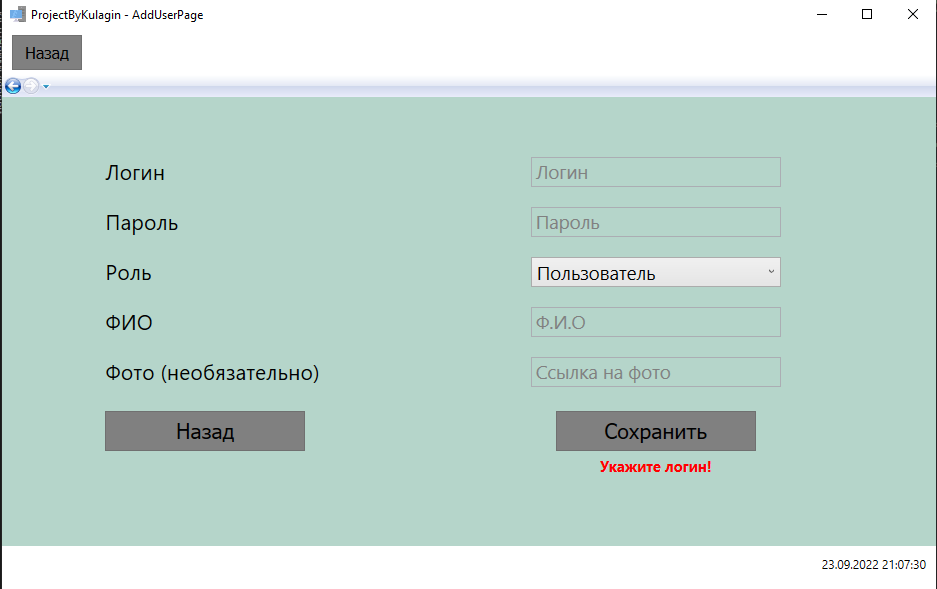


Рисунок 11. Проверка ввода данных

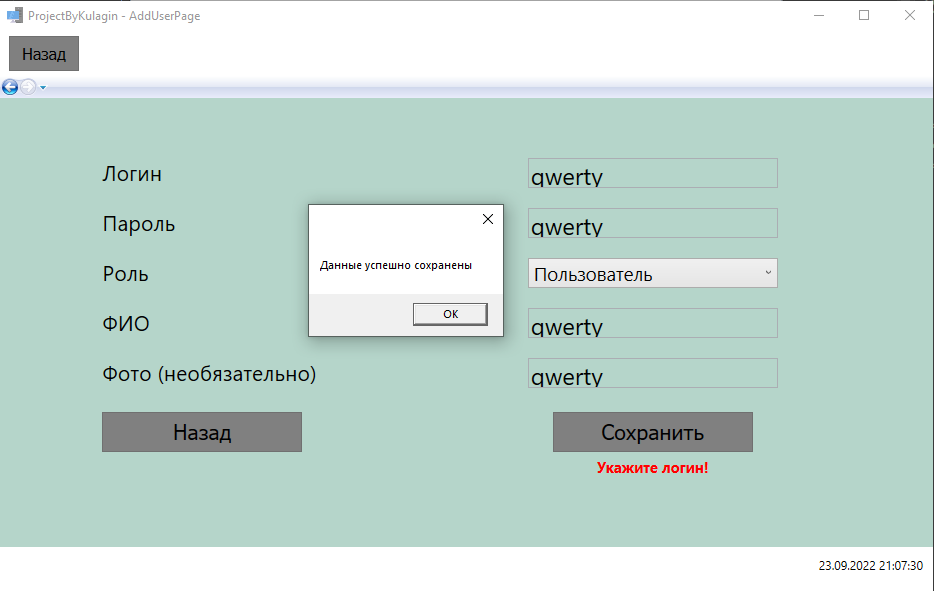


Рисунок 12. Добавление нового пользователя

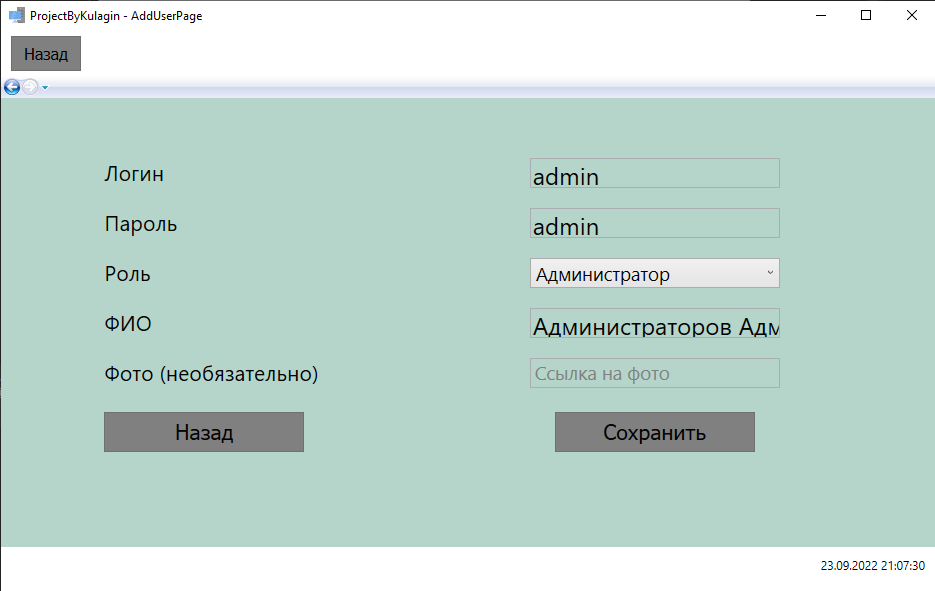


Рисунок 13. Редактирование пользователя

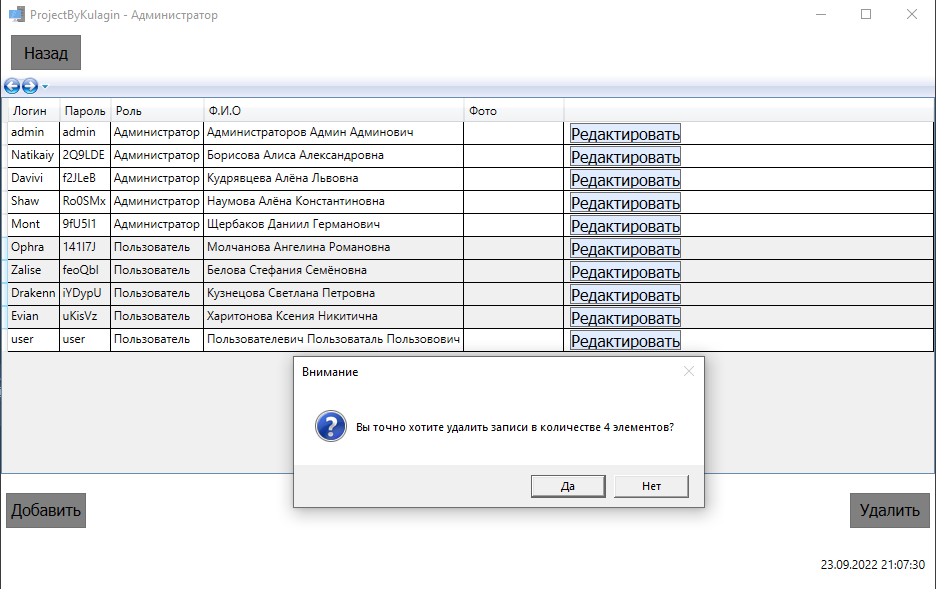


Рисунок 14. Удаление нескольких пользователей

## **Листинг программы**

AddUserPage.xaml

<Page x:Class="\_619\_Voronov.Pages.AddUserPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:\_619\_Voronov.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AddUserPage">

<Grid Background="#b5d5ca">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="4\*"/>

<ColumnDefinition Width="4\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition/>

<RowDefinition Height="145"/>

</Grid.RowDefinitions>

<StackPanel Grid.Column="1" Grid.Row="1">

<StackPanel.Resources>

<Style

TargetType="TextBlock">

<Setter Property="FontSize" Value="16pt"/>

<Setter Property="Margin" Value="10"/>

<Setter Property="Height" Value="30"/>

</Style>

</StackPanel.Resources>

<TextBlock Text="Логин"/>

<TextBlock Text="Пароль"/>

<TextBlock Text="Роль"/>

<TextBlock Text="ФИО"/>

<TextBlock Text="Фото (необязательно)"/>

</StackPanel>

<TextBlock Grid.Row="1" Grid.Column="2" Name="txtLogin"

Text="Логин" Height="30" Margin="0,12,0,0"

VerticalAlignment="Top" FontSize="14pt" Foreground="Gray"

HorizontalAlignment="Center" Width="240"/>

<TextBlock Grid.Row="1" Grid.Column="2" Name="txtPassword"

Text="Пароль" Height="30" Margin="0,62,0,0"

VerticalAlignment="Top" FontSize="14pt" Foreground="Gray"

HorizontalAlignment="Center" Width="240"/>

<TextBlock Grid.Row="1" Grid.Column="2" Name="txtFIO"

Text="Ф.И.О" Height="30" Width="240" Margin="0,162,0,0"

VerticalAlignment="Top" FontSize="14pt" Foreground="Gray"

HorizontalAlignment="Center"/>

<TextBlock Grid.Row="1" Grid.Column="2" Name="txtPhoto"

Text="Ссылка на фото" Height="30" Width="240" Margin="0,212,0,0"

VerticalAlignment="Top" FontSize="14pt" Foreground="Gray"

HorizontalAlignment="Center"/>

<StackPanel Grid.Column="2" Grid.Row="1">

<TextBox Text="{Binding Login}" Name="AddLogin"

HorizontalAlignment="Center" Background="Transparent" TextChanged="AddLogin\_TextChanged"/>

<TextBox Text="{Binding Password}" Name="AddPassword"

HorizontalAlignment="Center" Background="Transparent" TextChanged="AddPassword\_TextChanged"/>

<ComboBox Text="{Binding Role}" Name="AddRole" HorizontalAlignment="Center" FontSize="14pt">

<ComboBoxItem Content="Администратор"/>

<ComboBoxItem IsSelected="True" Content="Пользователь"/>

</ComboBox>

<TextBox Text="{Binding FIO}" Name="AddFIO"

HorizontalAlignment="Center" Background="Transparent" TextChanged="AddFIO\_TextChanged"/>

<TextBox Text="{Binding Photo}" Name="AddPhoto"

HorizontalAlignment="Center" Background="Transparent" TextChanged="AddPhoto\_TextChanged"/>

</StackPanel>

<Button Name="ButtonCancel" Content="Назад" Grid.Column="1" Grid.Row="2" FontSize="16pt"

Width="200" VerticalAlignment="Top" HorizontalAlignment="Left" Click="ButtonCancel\_Click"/>

<Button Name="ButtonSave" Content="Сохранить" Grid.Column="2" Grid.Row="2" FontSize="16pt"

Width="200" VerticalAlignment="Top" Click="ButtonSave\_Click"/>

<TextBlock Name="HideError" Grid.Row="2" Grid.Column="2" TextWrapping="Wrap" Text=""

TextAlignment="Center" FontSize="11pt" Foreground="Red" FontWeight="Bold"

HorizontalAlignment="Center" VerticalAlignment="Bottom" Width="300" Height="90"/>

</Grid>

</Page>

AddUserPage.xaml.cs

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 \_619\_Voronov.Pages

{

/// <summary>

/// Логика взаимодействия для AddUserPage.xaml

/// </summary>

public partial class AddUserPage : Page

{

private USER \_currentUser = new USER();

public AddUserPage(USER selectedUser)

{

InitializeComponent();

if (selectedUser != null)

\_currentUser = selectedUser;

DataContext = \_currentUser;

if (\_currentUser.Role == null)

\_currentUser.Role = "Пользователь";

}

private void AddLogin\_TextChanged(object sender, TextChangedEventArgs e)

{

txtLogin.Visibility = Visibility.Visible;

if (AddLogin.Text.Length > 0)

{

txtLogin.Visibility = Visibility.Hidden;

}

}

private void AddPassword\_TextChanged(object sender, TextChangedEventArgs e)

{

txtPassword.Visibility = Visibility.Visible;

if (AddPassword.Text.Length > 0)

{

txtPassword.Visibility = Visibility.Hidden;

}

}

private void AddFIO\_TextChanged(object sender, TextChangedEventArgs e)

{

txtFIO.Visibility = Visibility.Visible;

if (AddFIO.Text.Length > 0)

{

txtFIO.Visibility = Visibility.Hidden;

}

}

private void AddPhoto\_TextChanged(object sender, TextChangedEventArgs e)

{

txtPhoto.Visibility = Visibility.Visible;

if (AddPhoto.Text.Length > 0)

{

txtPhoto.Visibility = Visibility.Hidden;

}

}

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

{

if (string.IsNullOrWhiteSpace(\_currentUser.Login))

{

HideError.Text = "Укажите логин!";

return;

}

if (string.IsNullOrWhiteSpace(\_currentUser.Password))

{

HideError.Text = "Укажите пароль!";

return;

}

if (AddRole.Text == "")

{

HideError.Text = "Выберете роль!";

return;

}

else

{

\_currentUser.Role = AddRole.Text;

}

if (string.IsNullOrWhiteSpace(\_currentUser.FIO))

{

HideError.Text = "Укажите Ф.И.О!";

return;

}

if (\_currentUser.ID == 0)

Entities\_db.GetContext().USER.Add(\_currentUser);

try

{

Entities\_db.GetContext().SaveChanges();

MessageBox.Show("Данные успешно сохранены");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

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

{

NavigationService?.Navigate(new Administrator());

}

}

}

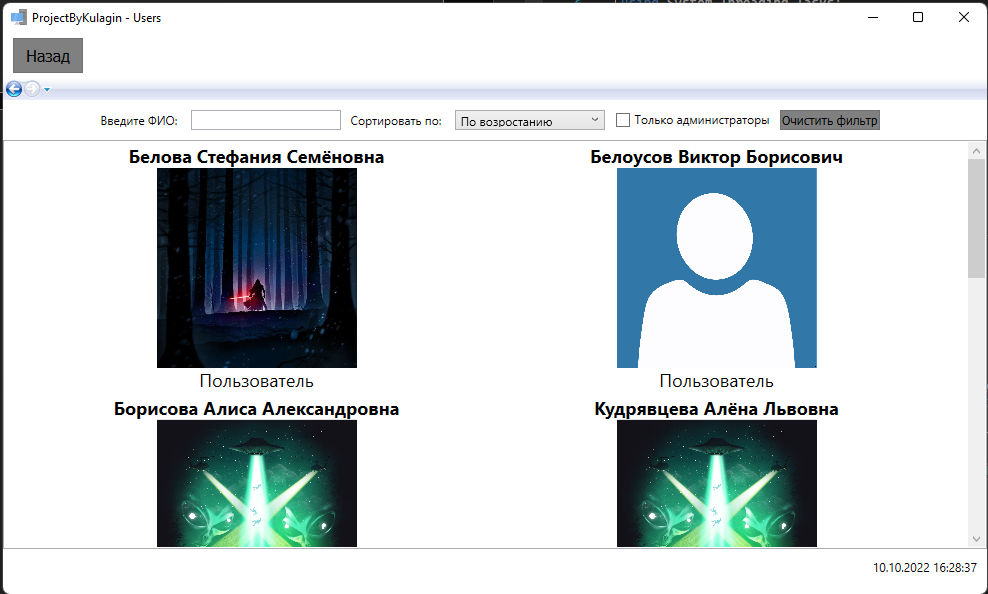


Рисунок 15. Страница Пользователя

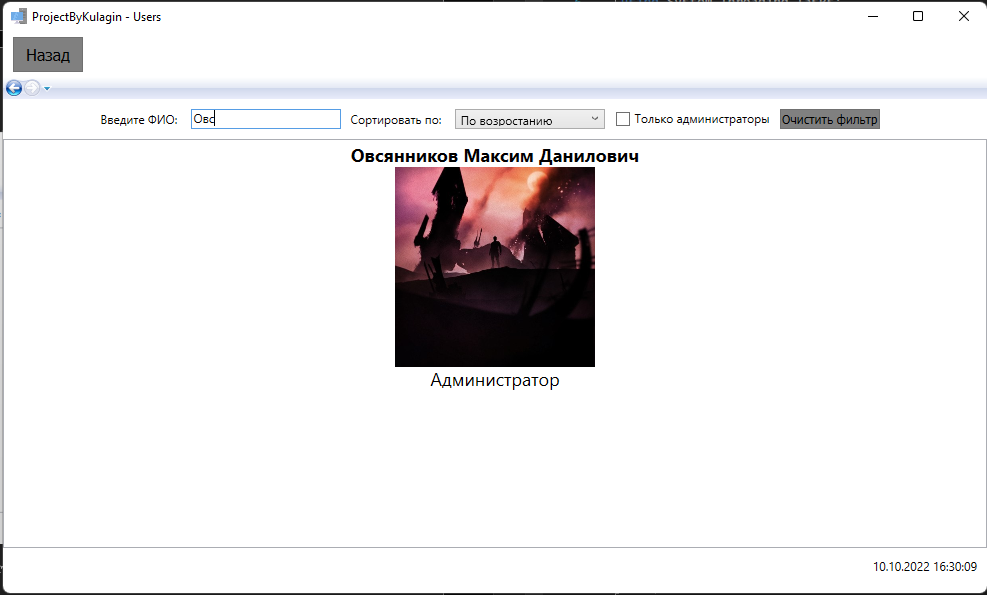


Рисунок 16. Поиск по ФИО

**Листинг программы**

User.xaml

<Page x:Class="\_619\_Voronov.Pages.Users"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:\_619\_Voronov.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Users">

<Grid Background="white">

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<WrapPanel Grid.Row="0" HorizontalAlignment="Center">

<TextBlock Text="Введите ФИО: " VerticalAlignment="Center"/>

<TextBox Name="FindText" Height="20" Width="150" FontSize="12px" TextChanged="FindText\_TextChanged"/>

<TextBlock Text="Сортировать по: " VerticalAlignment="Center"/>

<ComboBox Height="20" Width="150" Name="Cmb" SelectionChanged="ComboBox\_SelectionChanged">

<ComboBoxItem>

<TextBlock Text="По возростанию"/>

</ComboBoxItem>

<ComboBoxItem>

<TextBlock Text="По убыванию"/>

</ComboBoxItem>

</ComboBox>

<CheckBox Content="Только администраторы"

VerticalAlignment="Center" Name="CheckAdm"

Checked="CheckBox\_Checked" Unchecked="CheckBox\_Unchecked"/>

<Button Content="Очистить фильтр" Height="20" Width="100" FontSize="12px" Click="Button\_Click"/>

</WrapPanel>

<ListView Name="ListUser" Grid.Row="1"

ScrollViewer.HorizontalScrollBarVisibility="Disabled"

HorizontalContentAlignment="Center">

<ListView.ItemsPanel>

<ItemsPanelTemplate>

<WrapPanel Orientation="Horizontal" HorizontalAlignment="Center"/>

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="200"/>

<RowDefinition Height="auto"/>

</Grid.RowDefinitions>

<Image Width="200" Grid.Row="1" Stretch="UniformToFill"

HorizontalAlignment="Center" VerticalAlignment="Center">

<Image.Source>

<Binding Path="Photo">

<Binding.TargetNullValue>

<ImageSource>/Img/DefaultPhoto.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

<TextBlock VerticalAlignment="Center" TextAlignment="Center" Width="450"

Grid.Row="0" TextWrapping="Wrap" Text="{Binding FIO}"

HorizontalAlignment="Center" FontSize="18" FontWeight="Bold"/>

<TextBlock VerticalAlignment="Center" TextAlignment="Center" Width="200"

Grid.Row="2" HorizontalAlignment="Center" FontSize="18"

Text="{Binding Role}"/>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</Page>

User.xaml.cs

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 \_619\_Voronov.Pages

{

/// <summary>

/// Логика взаимодействия для Users.xaml

/// </summary>

public partial class Users : Page

{

public Users()

{

InitializeComponent();

var currentUsers = Entities.GetContext().USER.ToList();

ListUser.ItemsSource = currentUsers;

Cmb.SelectedIndex = 0;

CheckAdm.IsChecked = false;

UpdateUsers();

}

private void UpdateUsers()

{

var currentUsers = Entities.GetContext().USER.ToList();

currentUsers = currentUsers.Where

(x => x.FIO.ToLower().Contains(FindText.Text.ToLower())).ToList();

if (CheckAdm.IsChecked.Value)

currentUsers = currentUsers.Where(x => x.Role.Contains("Пользователь")).ToList();

if (Cmb.SelectedIndex == 0)

ListUser.ItemsSource = currentUsers.OrderBy(x => x.FIO).ToList();

else

ListUser.ItemsSource = currentUsers.OrderByDescending(x => x.FIO).ToList();

}

private void FindText\_TextChanged(object sender, TextChangedEventArgs e)

{

UpdateUsers();

}

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

{

UpdateUsers();

}

private void CheckBox\_Checked(object sender, RoutedEventArgs e)

{

UpdateUsers();

}

private void CheckBox\_Unchecked(object sender, RoutedEventArgs e)

{

UpdateUsers();

}

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

{

Cmb.SelectedIndex = 0;

CheckAdm.IsChecked = false;

FindText.Text = "";

UpdateUsers();

}

}

}

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

09.02.07 Информационные системы и программирование

## **ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ Профессиональный модуль - УП.01.01 Учебная практика**

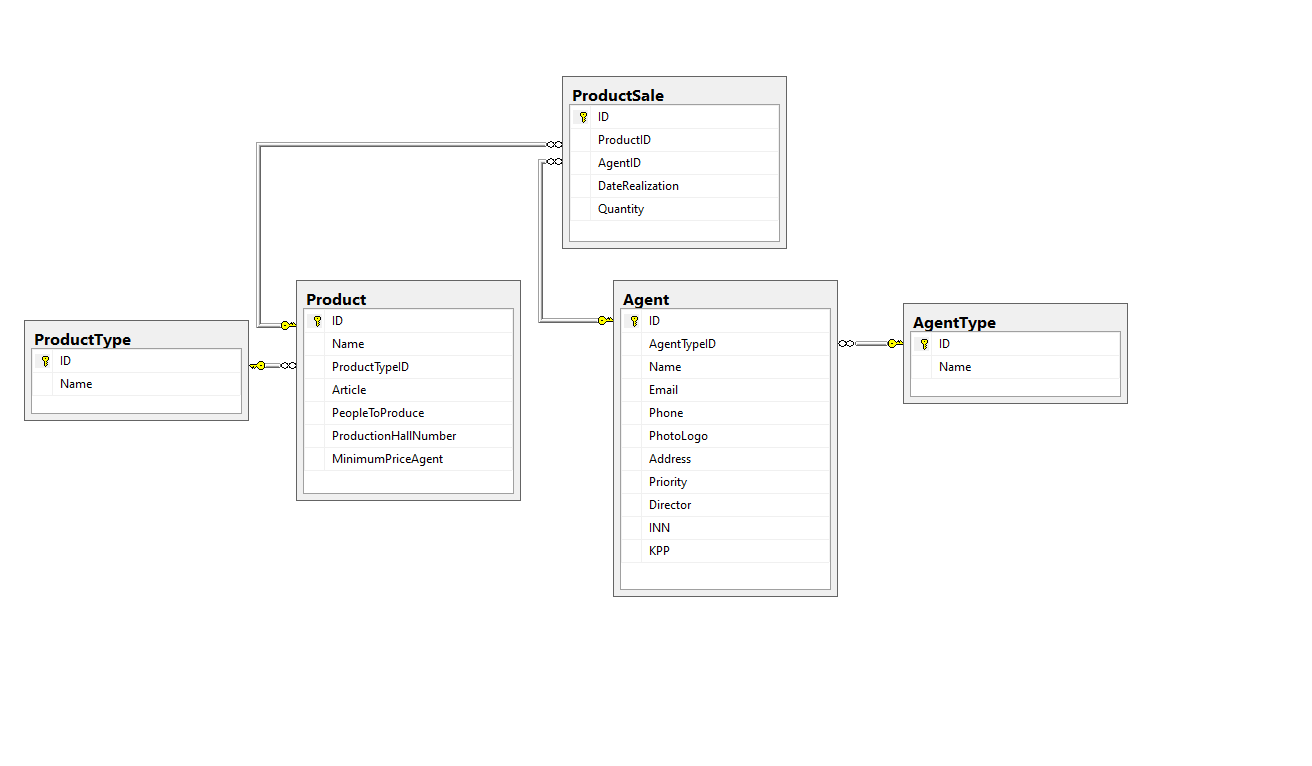
Выполнила: студент учебной группы 4ИСИП-619

Воронов А.В.

Руководитель практики от колледжа: Альшакова Е.Л.

**Москва – 2021**

**Цель работы:** отформатировать Excel файлы, создать базу данных и импортировать

 Рисунок 1. Диаграмма базы данных

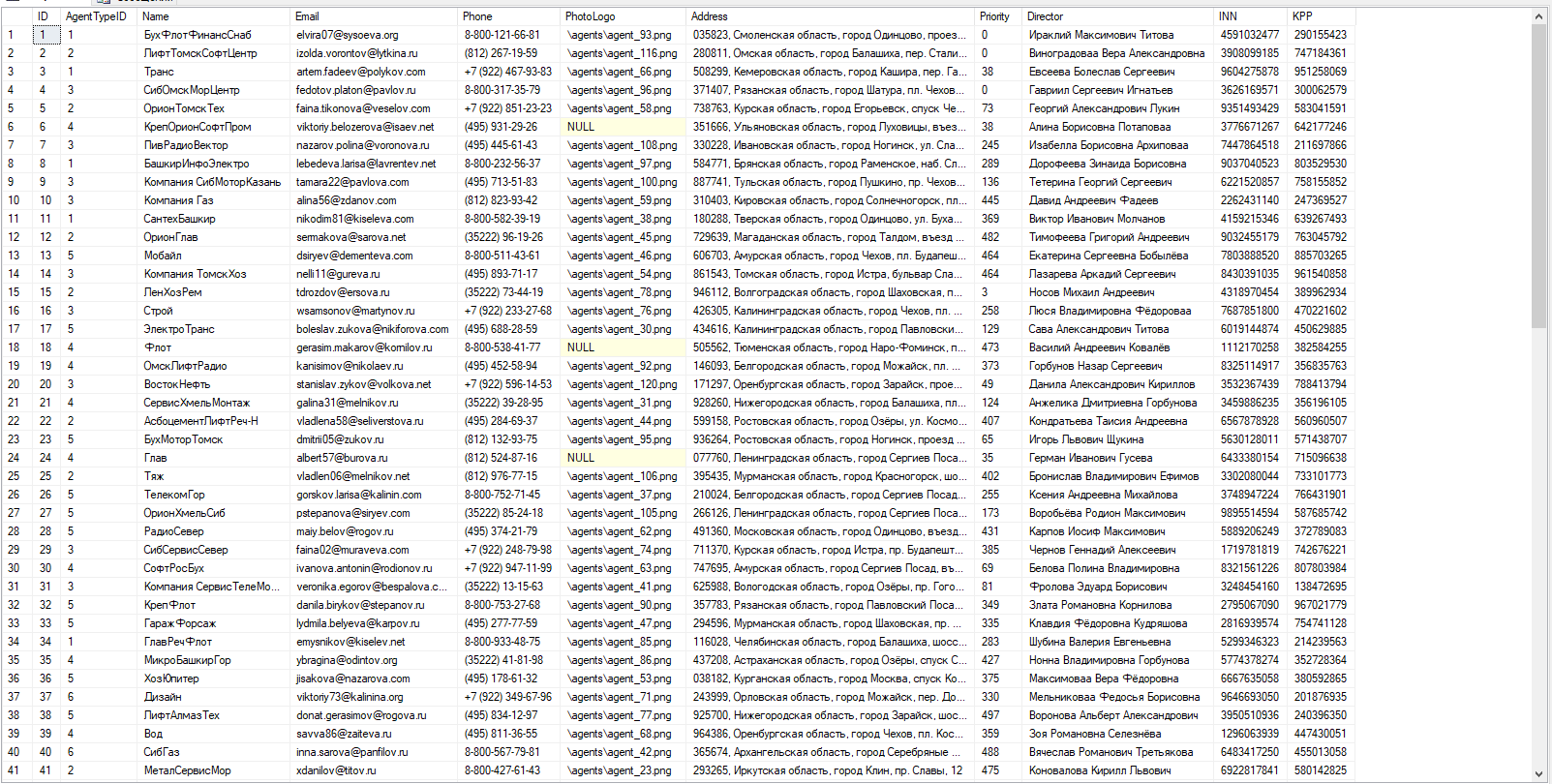


Рисунок 2. Таблица agent

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3. Таблица AgentType

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4. Таблица Product

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 5. Таблица ProductSale

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 6. Таблица ProductType

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 7. Данные из Excel файла Agent

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 8. Данные из Excel файла AgentType

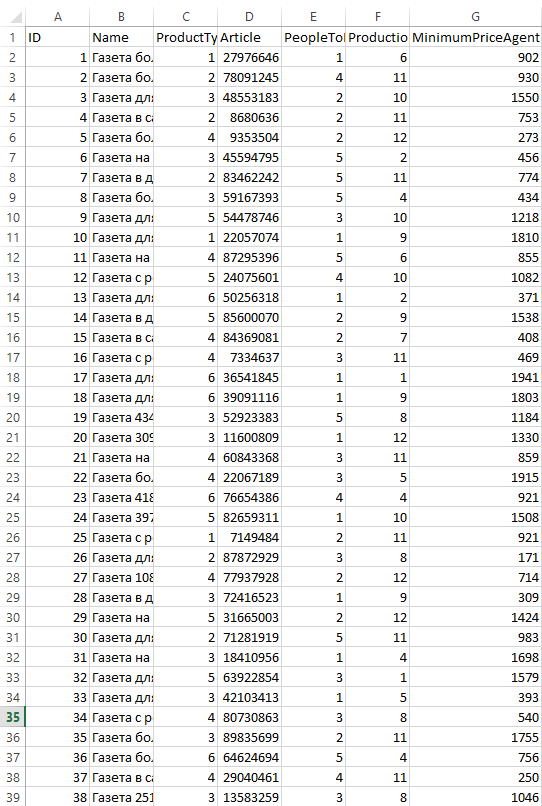


Рисунок 9. Данные из Excel файла Product

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 10. Данные из Excel файла ProductSale

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 11. Данные из Excel файла ProductType

**Федеральное государственное образовательное бюджетное учреждение**

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

Специальность 09.02.07 «Информационные системы и программирование»

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

Профессиональный модуль ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

Выполнил:

обучающийся учебной группы №\_\_\_619\_\_\_

Воронов А.В

*(подпись) (И.О. Фамилия)*

*(подпись) (И.О. Фамилия)*

Проверили:

Руководители практики от колледжа:

Сибирев И. В.

*(должность) (И.О. Фамилия)*

*(оценка) (подпись)*

**Москва**

**20­­­­22**

**Лабораторная работа №1-2**

**Цель работы:**

**Задание:** Спроектируйте базу данных. Создайте необходимые объекты базы данных

**Ход работы:** Создаем базу данных



**Таблица 1**

|  |  |  |
| --- | --- | --- |
| **Сотрудники** | | |
| **Название** | **Тип** | **Разрешить** |
| Код страдника | Int | + |
| Фамилия | varchar(50) | - |
| Имя | varchar(50) | - |
| Отчество | varchar(50) | - |
| Оклад | Int | - |

Таблица «Сотрудники» хранит в себе информацию: ФИО сотрудника, его оклад

**Таблица 2**

|  |  |  |
| --- | --- | --- |
| **Работы** | | |
| **Название** | **Тип** | **Разрешить** |
| Код страдника | Int | + |
| Код вида | varchar(50) | - |
| Дата начал | Date | - |
| Дата окончания | Date | - |

Таблица хранит в себе информацию о доп.работе сотрудника, вид его работы, код отрдника.

**Таблица 3**

|  |  |  |
| --- | --- | --- |
| **Работы** | | |
| **Название** | **Тип** | **Разрешить** |
| Код страдника | Int | + |
| Описание | varchar(50) | - |
| Оплата за день | Money | - |

Таблица хранит в себе информацию о виде работы и ее оплаты

**Вывод:** научились проектировать базу данных, создали ER-диаграмму и описали сущности

**Лабораторная работа №4**

**Цель работы:** Изучить создание запросов и представлений, использование механизма объединений.

**Задание:**

• Для базы данных согласно варианту задания (из Лабораторной

работы №1) написать следующие SQL запросы из Приложения 2

• Реализовать запросы из лабораторной работы 1 по варианту,

оформить два из них в виде представлений.

**Ход работы:** Создаем зпросы

a)

select \* from [dbo].[Вид\_работы]

select \* from [dbo].[Работы]

select \* from [dbo].[Сотрудники]

b)

SELECT [ID\_Sotrudnika],[Last\_name],[Name],[Patronic],[ID\_vida], [opisanie]

FROM [dbo].[Сотрудники]

INNER JOIN [dbo].[Вид\_работы] ON [ID\_vida] = [ID\_Sotrudnika]

c)

select [Last\_name], [Name], [Patronic] from [dbo].[Сотрудники]

order by [Last\_name], [Name], [Patronic]

**Вывод:** Научились создавать запросы для Бд

Создаем скрипт бд

USE [master]

GO

/\*\*\*\*\*\* Object: Database [Voronov619] Script Date: 27.10.2022 12:36:02 \*\*\*\*\*\*/

CREATE DATABASE [Voronov619]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'Voronov619', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL\DATA\Voronov619.mdf' , SIZE = 5120KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )

LOG ON

( NAME = N'Voronov619\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL\DATA\Voronov619\_log.ldf' , SIZE = 1024KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)

GO

ALTER DATABASE [Voronov619] SET COMPATIBILITY\_LEVEL = 120

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [Voronov619].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [Voronov619] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [Voronov619] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [Voronov619] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [Voronov619] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [Voronov619] SET ARITHABORT OFF

GO

ALTER DATABASE [Voronov619] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [Voronov619] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [Voronov619] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [Voronov619] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [Voronov619] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [Voronov619] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [Voronov619] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [Voronov619] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [Voronov619] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [Voronov619] SET DISABLE\_BROKER

GO

ALTER DATABASE [Voronov619] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [Voronov619] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [Voronov619] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [Voronov619] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [Voronov619] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [Voronov619] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [Voronov619] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [Voronov619] SET RECOVERY SIMPLE

GO

ALTER DATABASE [Voronov619] SET MULTI\_USER

GO

ALTER DATABASE [Voronov619] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [Voronov619] SET DB\_CHAINING OFF

GO

ALTER DATABASE [Voronov619] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [Voronov619] SET TARGET\_RECOVERY\_TIME = 0 SECONDS

GO

ALTER DATABASE [Voronov619] SET DELAYED\_DURABILITY = DISABLED

GO

USE [Voronov619]

GO

/\*\*\*\*\*\* Object: Table [dbo].[vid\_rabot] Script Date: 27.10.2022 12:36:03 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[vid\_rabot](

[ID\_vida] [int] NOT NULL,

[opisanie1] [varchar](1) NULL,

[pay\_for\_work] [int] NULL,

UNIQUE NONCLUSTERED

(

[pay\_for\_work] ASC

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

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

**Вывод:** Сделали бакап БД

**Задание:**

Для базы данных согласно варианту задания (из Лабораторной работы

№2) написать:

**Ход работы:** Создаем хранимые процедуры

a)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

alter PROCEDURE zarplata

-- Add the parameters for the stored procedure here

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

select SUM([Salary]) + (select sum([pay\_for\_work]) from [dbo].[Вид\_работы])

from [dbo].[Сотрудники]

END

GO

**Вывод:** Повторил создание хромых процедур для Бд

**Задание:**

Для базы данных из лабораторной работы 2

1 Добавить ограничение DEFAULT

2 Создать триггеры индикаторы:

Добавление, выводящий на экран сообщение "Запись добавлена"

при добавлении новой записи в таблицу

Изменение, выводящий на экран с сообщение "Запись изменена"

при изменении записи в таблице

Удаление, выводящий на экран с сообщение "Запись удалена" при

удалении записи из таблицы

3 Создать триггер по удалении, когда данные удаляются сразу из двух связанных таблиц.

4 Проверить и продемонстрировать созданных триггеров.

**Ход работы:** Создаем триггер

1. Создаем новую таблицу

create table vid\_rabot([ID\_vida] [int] not null,

[opisanie1] [varchar] DEFAULT (0),

[pay\_for\_work] [int] UNIQUE)

1. Новая запись

Create TRIGGER tr\_add on [dbo].[Сотрудники] after insert

as

begin SET NOCOUNT ON;

PRINT 'НОВАЯ ЗАПИСЬ'

END

GO

2. Изменения

CREATE TRIGGER tr\_change ON [dbo].[Сотрудники]

AFTER UPDATE AS BEGIn

SET NOCOUNT ON;

PRINT 'Запись изменена'

END

1. Удаление записи

CREATE TRIGGER tr\_DELETE ON [dbo].[Работы]

AFTER DELETE AS BEGIN

SET NOCOUNT ON;

PRINT 'Запись удалена'

END

GO

**Вывод:** Создали триггеры в бд

**Федеральное государственное образовательное бюджетное учреждение  
высшего образования**

**«Финансовый университет при Правительстве Российской Федерации»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ПМ.01 Разработка мобильных приложений**

**Группа: 4ИСиП-619**

**УТВЕРЖДАЮ**

**Председатель цикловой-комиссии**

**информационных систем и программирования**

**\_\_\_\_\_\_\_\_\_\_\_\_\_ / Т.Г. Титов /**

**\_\_\_\_.\_\_\_\_. 2022**

**ПРОЕКТ КУРСОВОЙ**

**На тему:** Разработка Мобильного приложения информирования о динамике учетных цен на драгоценные металлы.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Руководитель курсового проекта

\_\_\_\_\_\_\_\_\_\_\_\_\_ /П.А. Костиков/

Исполнитель курсового проекта

\_\_\_\_\_\_\_\_\_\_\_\_\_ /А.В.Воронов/

Оценка за проект: **\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2022**

СОДЕРЖАНИЕ

ВВЕДЕНИЕ………………………………………………………………………..3

[ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc118286677)

[1.1 Проектное исследование предметной области 4](#_Toc118286678)

[1.2 Обзор предшествующих решений. 5](#_Toc118286679)

[1.3 Постановка задачи 7](#_Toc118286680)

[1.4 Характеристика инструментальных средств разработки 8](#_Toc118286681)

[ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ 11](#_Toc118286687)

[2.1 Анализ требований и определение спецификаций программного обеспечения 11](#_Toc118286688)

[2.2 Проектирование программного обеспечения 13](#_Toc118286689)

[2.3 Разработка программного обеспечения 14](#_Toc118286690)

[2.4 Отладка и тестирование программы 24](#_Toc118286691)

[2.5 Руководство по использованию программы 25](#_Toc118286692)

[ЗАКЛЮЧЕНИЕ 28](#_Toc118286694)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 27](#_Toc118286695)

[ПРИЛОЖЕНИЕ 1 28](#_Toc118286696)

ВВЕДЕНИЕ

Драгоценные металлы — это редкие химические элементы, не подверженные, в отличие от других металлов, коррозии и окислению. К ним относят золото, серебро, платину и палладий. Они редко встречаются в природе и имеют ювелирную ценность. Их добыча и производство достаточно трудоемки. Все это делает их средствами накопления и обмена.

Несомненно полезной для пользователя является приложение которое информирует его. Установив и включив данное приложение, пользователь будет всегда знать о ценах на драгоценные металлы. Приложение было создано для информирования о ценах на драгоценные металлы на сайте ЦБ РФ.

Объектом исследования является: Приложение банка

Предметом исследования будут являться: инструментальные средства и технологии для разработки мобильного приложения.

Цель данной курсовой работы, заключается в разработке мобильного приложения для информирования о международных резервах Российской Федерации.

# ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Проектное исследование предметной области

Драгоценные металлы — это редкие химические элементы, не подверженные, в отличие от других металлов, коррозии и окислению. К ним относят золото, серебро, платину и палладий. Они редко встречаются в природе и имеют ювелирную ценность.

Их добыча и производство достаточно трудоемки. Все это делает их средствами накопления и обмена. К ним также относится медь, которая не является драгоценным металлом, однако из-за важного промышленного значения она стала популярным инструментом у инвесторов. Все пять металлов торгуются на международных биржах и на Мосбирже, их можно купить через брокерский счет.

Помимо этого металлы можно приобрести в виде слитков и монет, а также открыть обезличенный металлический счет, на баланс которого можно купить активы.

Постановка задачи

В самом начале работы с приложением человек может увидеть мобильную версию сайта с таблицей всех депозитных счетов банков.

Задачи, для достижения заданной цели курсового проекта, заключаются в следующем:

* выполнить анализ требований и разработать спецификации;
* Разработать код и графическое приложение;
* подготовить данные;
* Работа над ошибками;
* Проверка работы приложения.

Для разработки данной курсовой были использованы следующие инструментальные средства:

* объектно-ориентированный язык программирования Kotlin;
* сервис для построения диаграмм Draw.io;
* текстовый процессор Microsoft Word;
* программа для подготовки и просмотра презентаций Microsoft PowerPoint.
* Программа для проверки и написания кода “Android Studio”.

1.2 Обзор предшествующего решения.

Решение «Metals»

Metals — это приложение для информировании, используется для просмотра цен драгоценных металлов на сайте ЦБ РФ.

Функции решения:

1. Информация об депозитах.
2. Удобный календарь для просмотра в нужные дни.
3. Версия для слабовидящих ( подсветка ).

Изображение выглядит как текст, снимок экрана, монитор

Автоматически созданное описание

Рисунок 1. Интерфейс решения «Metals»

Стоимость: бесплатно

Достоинства: функциональное и простое в обращении приложение.

Недостатки: нет.

## 1.3 Постановка задачи

В приложении «Metals» выбранном для данного проекта пользователи не отличаются в привилегиях ролей, для каждого пользователя доступна каждая функция приложения.

Список доступных функций:

Функция для слабовидящих – данная функция делает затемненные слова более светлыми.

Функция поиска – Осуществляется нажатием на календарь, который находиться чуть выше таблицы, благодаря поиску можно увидеть нужный депозит банка в таблице.

Так же имеется Функция перевода с RU на EN – нажимая на кнопку происходит смена языка.

## 1.4 Характеристика инструментальных средств разработки

## Описание среды разработки Android studio

Android Studio — интегрированная среда разработки производства Google, с помощью которой разработчикам становятся доступны инструменты для создания приложений на платформе Android OS. Android Studio можно установить на Windows, Mac и Linux.

Среда Android Studio предназначена как для небольших команд разработчиков мобильных приложений (даже в количестве одного человека), или же крупных международных организаций с GIT или другими подобными системами управления версиями. Опытные разработчики смогут выбрать инструменты, которые больше подходят для масштабных проектов. Решения для Android разрабатываются в Android Studio с использованием Java или C++. В основе рабочего процесса Android Studio заложен концепт непрерывной интеграции, позволяющий сразу же обнаруживать имеющиеся проблемы. Продолжительная проверка кода обеспечивает возможность эффективной обратной связи с разработчиками. Такая опция позволяет быстрее опубликовать версию мобильного приложения в Google Play App Store.

Android Studio совместима с платформой Google App Engine для быстрой интеграции в облаке новых API и функций. В среде разработки вы найдете различные API, такие как Google Play, Android Pay и Health. Присутствует поддержка всех платформ Android, начиная  с Android 1.6. Есть варианты Android, которые существенно отличаются от версии Google Android. Самая популярная из них — это Amazon Fire OS. В Android Studio можно создавать APK для этой ОС. Поддержка Android Studio ограничивается онлайн-форумами.

## Описание языка программирование Kotlin

Kotlin — это язык программирования, созданный в компании JetBrains. Его разработали в 2011 году на замену Java, который в компании считали чересчур многословным. Новый язык получился на 40% компактнее предшественника, что помогло ускорить работу над основным продуктом JetBrains — средой разработки IntelliJ IDEA. При этом Kotlin полностью совместим с Java, потому что запускается на его виртуальной машине (JVM).

Благодаря совместимости с JVM, Kotlin можно запустить на любых серверах, где работает Java. Поэтому его часто используют для создания бэкенда — той части систем, которая выполняется на сервере и не видна обычному пользователю.

Java и Kotlin — функционально совместимые языки. Они действительно очень похожи: оба статически типизированы, поддерживают ООП и работают на JVM. В некоторых средах разработки их функции и классы даже можно смешивать.

Описание инструмента для построения задач и сроков GanttProject

GanttProject – это программа, которая предназначена для ведения проектов и формирования информационных баз на основе построения диаграмм Ганта. Данная утилита позволяет разделить один проект на несколько подпунктов или этапов для последующего выполнения задач определенными исполнителями с разными сроками.

## Описание инструмента для построения диаграмм draw.io

Draw.io – инструмент, который позволяет создавать различного рода диаграммы и схемы (блок-схемы, сетевые диаграммы, интеллект-карты, отношения сущностей, программные блоки, UML, макеты и т. д.). Имеется как браузерная, так и десктопная версия. Draw.io позволяет пользователям отслеживать и восстанавливать изменения, импортировать и экспортировать в PDF, PNG, XML, VSDX, HTML, а также автоматически публиковать и делиться работами.

## Описание программного продукта для создания пояснительной записки

MicrosoftWord (часто — MSWord, WinWord или просто Word) — текстовый процессор, предназначенный для создания, просмотра, редактирования и форматирования текстов статей, деловых бумаг, а также иных документов, с локальным применением простейших форм таблично-матричных алгоритмов. Выпускается корпорацией Microsoft в составе пакета Microsoft Office. Первая версия была написана Ричардом Броди (Richard Brodie) для IBM PC, использующих DOS, в 1983 году. Позднее выпускались версии для Apple Macintosh (1984), SCO UNIX и Microsoft Windows (1989). Текущей версией является Microsoft Office Word 2021 для Windows и macOS, а также веб-версия Word Online (Office Online), не требующая установки программы на компьютер.

## Описание программного продукта для создания презентации

Microsoft PowerPoint позволяет создавать яркие и наглядные презентации, состоящие из набора слайдов и включающие текст, изображения, таблицы, графики, диаграммы, блок-схемы, 3D-модели, аудио, видео. Готовые презентации подходят для воспроизведения на больших экранах, в том числе с помощью кинопроектора.

Программа предлагает множество шаблонов оформления презентаций в зависимости от тематики: бизнес, образование, маркетинг и другие. При необходимости пользователи могут создавать собственные шаблоны PowerPoint, а также загружать свежие темы с сайта Microsoft.

Приложение включает расширенные инструменты для форматирования текста, построения различных видов диаграмм и графиков, работы с изображениями, фигурами и мультимедиа. Программа также позволяет настраивать события, которые будут происходить при клике или наведении мыши на выбранные объекты.

# ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1 Анализ требований и определение спецификаций программного обеспечения

На ниже изображённых рисунках 2 и 3 показано как идёт обработка данных при выборе функции пользователем.

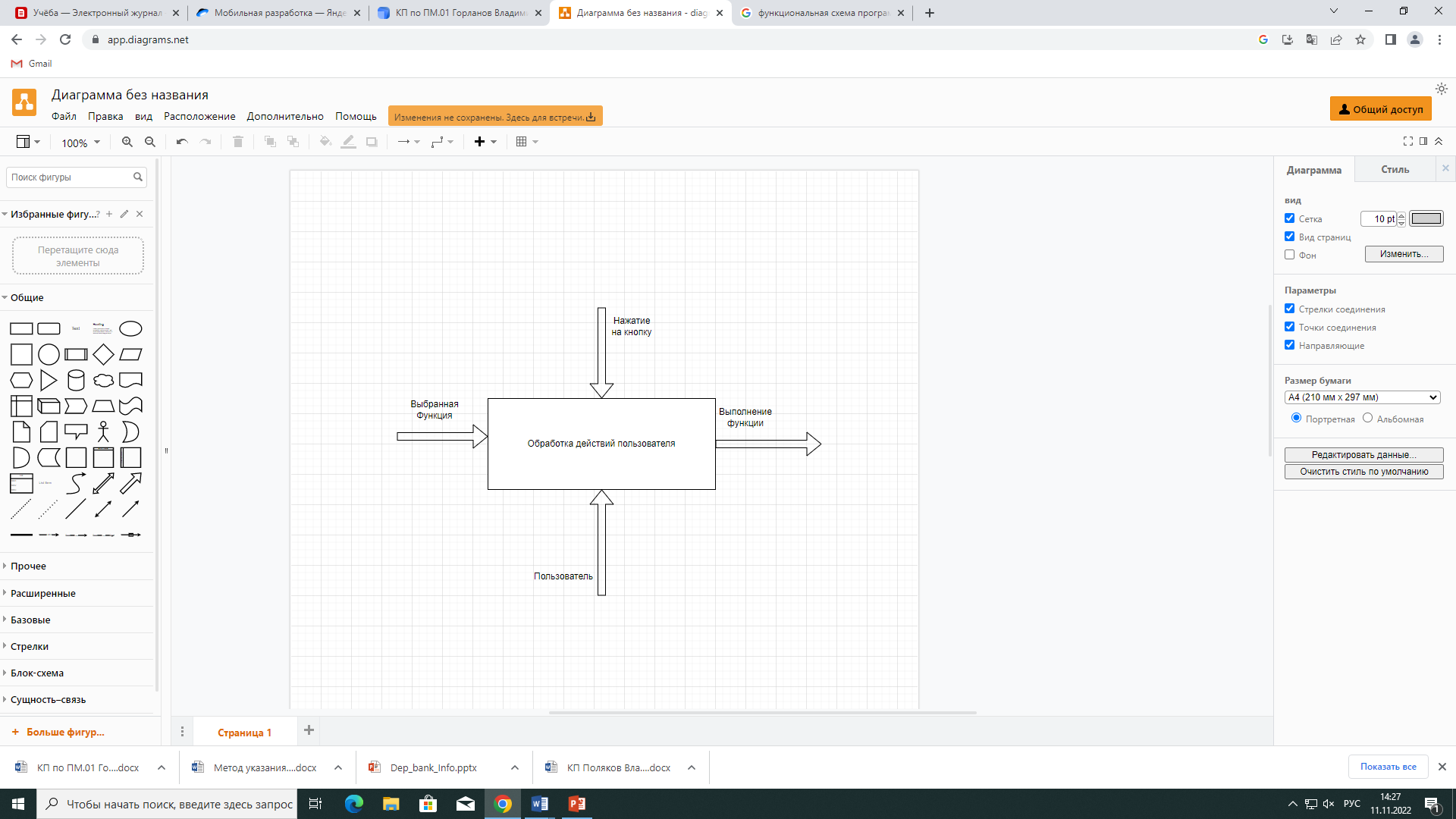


Рисунок 2. Функциональная схема программы

Рисунок 3. Функциональная схема программы

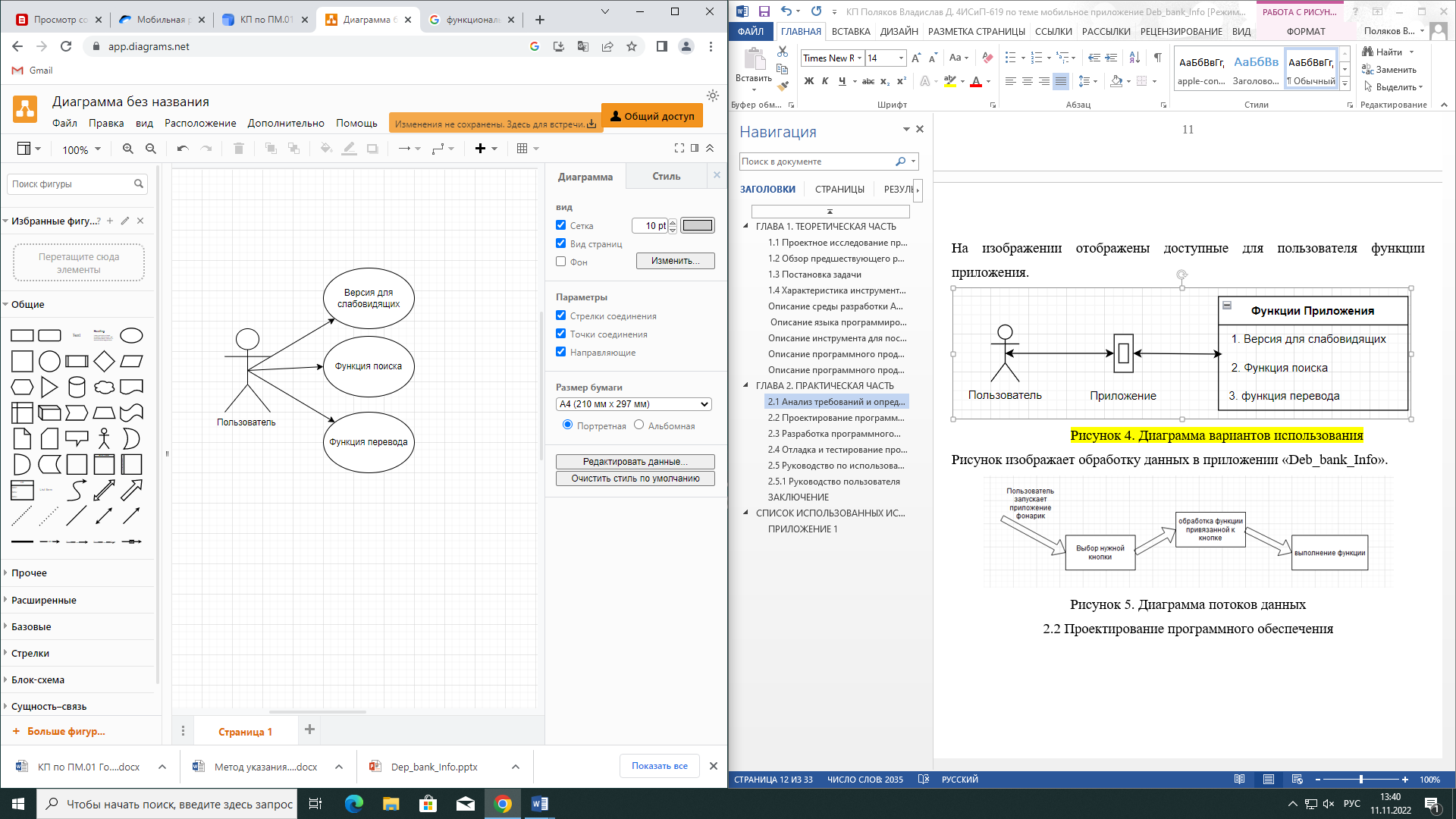
На изображении отображены доступные для пользователя функции приложения.

Рисунок 4. Диаграмма вариантов использования

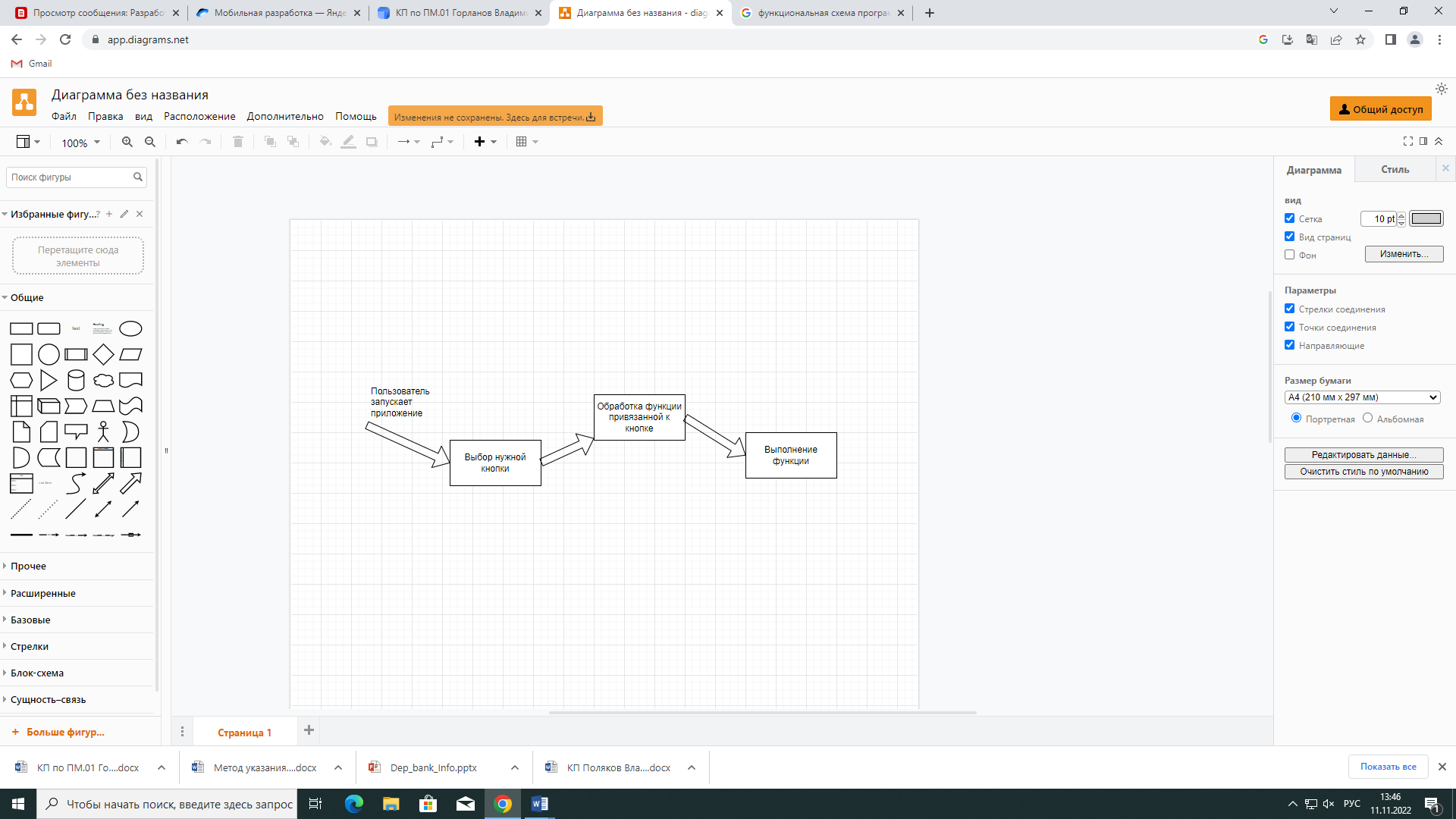
Рисунок изображает обработку данных в приложении «Metals».

Рисунок 5. Диаграмма потоков данных

## 2.2 Проектирование программного обеспечения

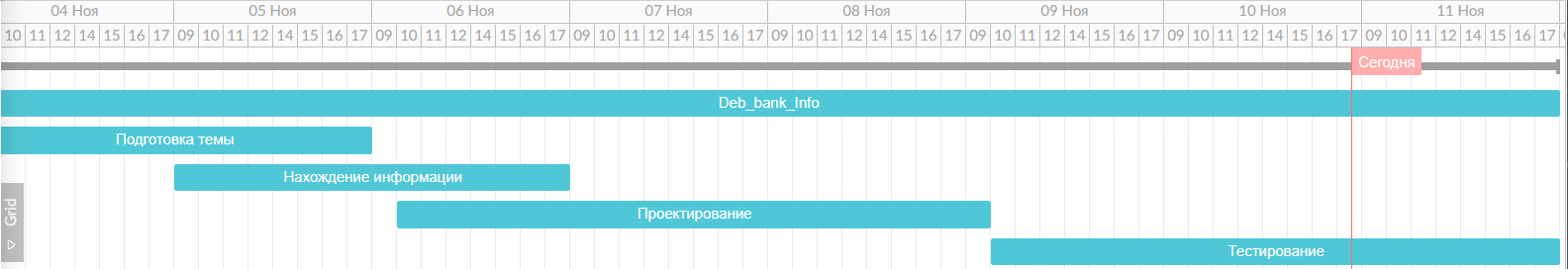
На изображении показан график работы над приложением «Metals».

Рисунок 6. Диаграмма Ганта

## 2.3 Разработка программного обеспечения

Изображение выглядит как текст, снимок экрана, монитор

Автоматически созданное описание

Рисунок 7. Итоговый вид приложения

Программный листинг MainActivity представлен ниже:

package com.example.Metals

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.view.KeyEvent

import android.view.MotionEvent

import android.view.View

import android.webkit.WebSettings

import android.webkit.WebView

import android.webkit.WebViewClient

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val mWebView = findViewById<View>(R.id.WebView) as WebView

mWebView.loadUrl("https://cbr.ru/hd\_base/metall/metall\_base\_new/")

val webSetting: WebSettings = mWebView.settings

webSetting.javaScriptEnabled = true

mWebView.webViewClient = WebViewClient()

mWebView.canGoBack()

mWebView.setOnKeyListener(View.OnKeyListener { v, keyCode, event ->

if(keyCode == KeyEvent.KEYCODE\_BACK

&& event.action == MotionEvent.ACTION\_UP

&& mWebView.canGoBack()){

mWebView.goBack()

return@OnKeyListener true

}

false

})

}

}

Программный листинг AndroidManifest.xml представлен ниже:

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.Metals">

<uses-permission android:name="android.permission.INTERNET"/>

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="Metals"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/Theme.Metals">

<activity android:name=".MainActivity"

android:exported="true">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

Программный листинг Activity\_Main.xml представлен ниже:

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<WebView

android:id="@+id/WebView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_centerHorizontal="true"/>

</RelativeLayout>

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8.1 Функция “Слабовидящие”(Нахождение).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8.2 Функция “Слабовидящие”(Итог).

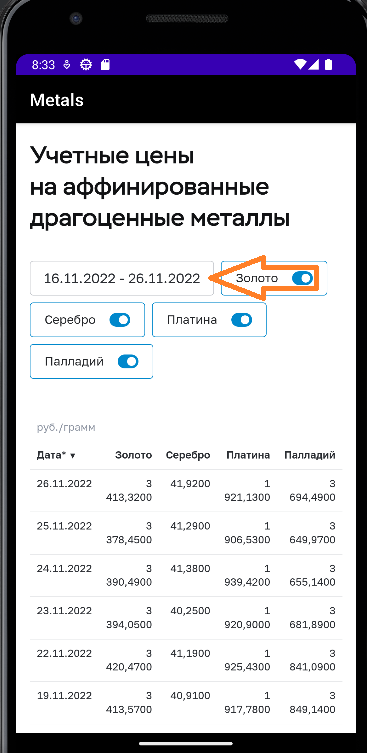


Рисунок 9.1 Функция “Поиск”(Нахождение).

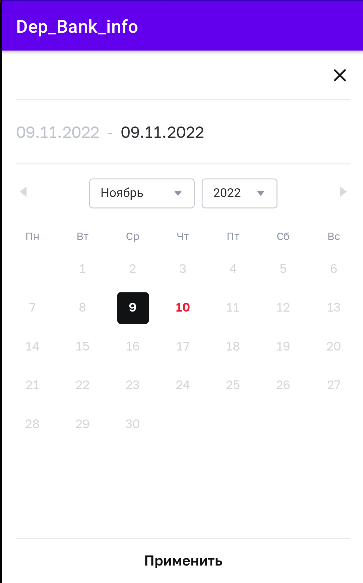


Рисунок 9.2 Функция “Поиск”(Выбираем дату и применяем).

Изображение выглядит как текст, монитор, телефон, снимок экрана

Автоматически созданное описание

Рисунок 9.3 Функция “Поиск”(Результат).

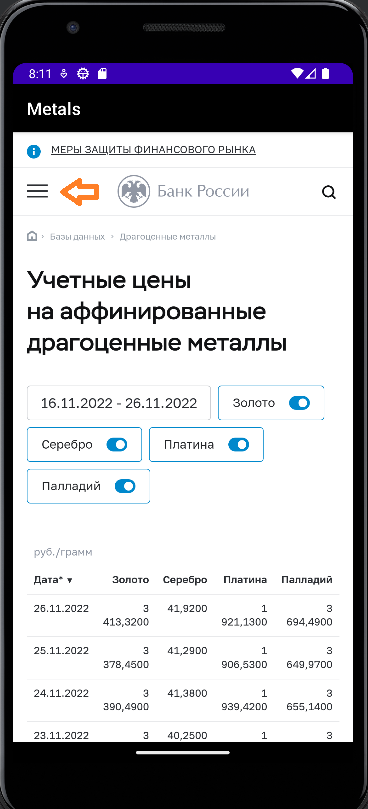


Рисунок 10.1. Функция “Перевода”(Нахождение).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10.2. Функция “Перевода”(Включение).

] Изображение выглядит как текст, снимок экрана, монитор, телефон

Автоматически созданное описание

Рисунок 10.3. Функция “Перевода”(Итог).

## 2.4 Отладка и тестирование программы

Таблица 2. Результаты отладки и тестирования программы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № теста | Входные данные | Вводимое значение | Ожидаемая  реакция программы | Фактическая реакция программы | Ошибка выявлена |
| 1 | Запуск программы «Metals» | “ ” | Создание окна MainActivity | На рисунке 7 | Нет |
| 2 | Нажатие на кнопку «доп. функции» | “ ” | Переход на окно  MainActivity | На рисунках 8.1, 8.2, 9.1, 9.2, 9.3, 10.1, 10.2, 10.3 | Нет |

В таблице номер 2 в столбце «Входные данные» описываются действия пользователя, поле «Вводимое значение» пустое так как в программе пользователь ничего не вводит, а только нажимает на кнопки. «Ожидаемая реакция программы» описывает ответ программы на действие пользователя, а в столбце «Фактическая реакция программы» - указан номер рисунка на котором представлен скриншот программы.

В проекте методом отладки является автокоррекция от среды разработки Android Studio, которая просматривает код программы и выделяет ошибки при их наличии.

Все запланированные к реализации функции успешно выполняются. Работа программы осуществляется при заданном техническом задании нефункциональных требований.

## 2.5 Руководство по использованию программы

## Руководство пользователя

Программное решение «Deb\_bank\_Info» представляет из себя информатор. В данной программе представлены функции: “Слабовидящие”, ”Поиск”, ”Переводчик”.

Функция для слабовидящих – данная функция делает затемненные слова более светлыми.

Функция поиска – Осуществляется нажатием на календарь, который находиться чуть выше таблицы, благодаря поиску можно увидеть нужный депозит банка в таблице.

Функция перевода с RUSSIA на ENGLISH – нажимая на кнопку происходит смена языка.

Для установки приложения необходимо скачать файл-установщик, далее нужно запустить установщик и подтвердить установку во всплывающем окне. По завершению установки в появившемся окне нажать на «готово».

Для запуска программы требуется нажать на иконку приложения на рабочем столе или в разделе «меню».

После запуска программы пользователь может увидеть информацию и перейти в дополнительные функции, путём нажатия на кнопку на главном экране приложения.

## ЗАКЛЮЧЕНИЕ

По результатам курсового проекта была разработано программное решение в виде приложения «Metals». Данное приложение обладает информационной сутью и дополнительными функциями: поиска (функция «поиск»), изменение языка (функция «Перевод») и повышения яркости букв (функция «Слабовидящие»). Спроектированная программа является простой в пользовании и интуитивно понятной, позволяет воспользоваться представленной информацией.

В процессе работы над курсовым проектом были освоены среда разработки мобильных приложений Android Studio и язык программирования Kotlin для чего был проанализирован соответствующий справочный материал. По итогам разработки проекта была проведена отладка программного кода с целью выявления ошибок и последующим их исправлением. После отладки осуществлялось тестирование программы на мобильном устройстве с операционной системой андроид с внесением необходимых поправок.

В дальнейшем программный продукт курсового проекта предполагается оптимизировать и модифицировать с целью размещения на платформах.

Вариант 7

Задание 3

Предметная область: Телефонный узел связи (учет абонентов).

Основные предметно-значимые сущности: Абоненты, Подразделения, Помещения.

Основные предметно-значимые атрибуты сущностей:

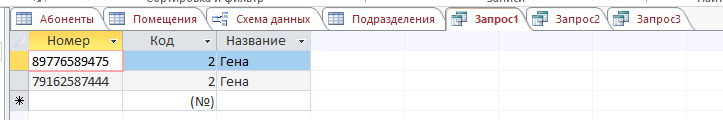
- абоненты – фамилия, имя, отчество, дата рождения, подразделение;

- помещения – название или номер помещения, вид помещения (аудитория,

кабинет и т.п.), подразделение;

- подразделения – название, вид подразделения.  
  
Основные требования к функциям системы:

- выбрать номера абонента по подразделениям;



- выбрать номера абонента по помещениям; Изображение выглядит как стол

Автоматически созданное описание

- подсчитать количество абонентов по подразделениям, помещениям.

Изображение выглядит как стол

Автоматически созданное описание

Задание 3

Вариант 4

Предметная область: Учебно-методическое

Управление (профессорско-преподавательский состав).

Основные предметно-значимые сущности: Сотрудники, Подразделения, Дисциплины.

Основные предметно-значимые атрибуты сущностей:

- сотрудники – фамилия, имя, отчество, пол, дата рождения, адрес прописки,

должность, подразделение;

- подразделения – название, вид подразделения;

- дисциплины – название.

Основные требования к функциям системы:

- выбрать дисциплины, читаемые сотрудниками или определенным сотрудником;

Изображение выглядит как стол

Автоматически созданное описание

- выбрать список сотрудников по подразделениям или определенному

подразделению;

Изображение выглядит как стол

Автоматически созданное описание

- выбрать дисциплины, читаемые сотрудниками по подразделениям или

определенному подразделению.

Изображение выглядит как текст, стол

Автоматически созданное описание

**Цель работы:** приобретение навыков анализа предметной области и построения концептуальной модели.

**Вариант №7. Нотариальная контора.**

**Описание предметной области.**

Вы работаете в нотариальной конторе. Вашей задачей является отслеживание финансовой стороны работы компании. Деятельность нотариальной конторы организована образом: фирма готова предоставить клиенту определенный комплекс услуг. Для наведения порядка вы формализовали эти услуги, составив их список с описанием каждой услуги. При обращении к вам клиента его стандартные данные (название, вид деятельности, фиксируются в базе данных. По каждому факту оказания услуги клиенту составляется документ. В документе указываются услуга, сумма сделки, комиссионные (доход конторы), описание сделки.

**Основные сущности.**

Таблица «Клиент»

|  |  |  |
| --- | --- | --- |
| Ключ | Название столбца | Тип данных |
| Первичный ключ | Код клиента | int |
| - | Название | varchar(50) |
| - | Вид деятельности | varchar(50) |
| - | Номер телефона | varchar(50) |
| - | Адрес | varcahr(50) |

Таблица «Услуги»

|  |  |  |
| --- | --- | --- |
| Ключ | Название столбца | Тип данных |
| Первичный ключ | Код услуги | int |
| - | Описание | varcahr(MAX) |
| - | Название | varchar(50) |

Таблица «Сделка»

|  |  |  |
| --- | --- | --- |
| Ключ | Название столбца | Тип данных |
| Первичный ключ | Код сделки | int |
| Внешний ключ | Код услуги | int |
| Внешний ключ | Код клиента | int |
| - | Сумма | money |
| - | Комиссионные | money |
| - | Описание | varcahr(MAX) |

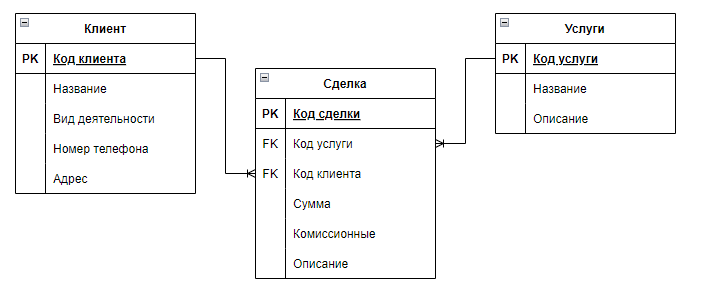


Рисунок 1. ER-диаграмма.

**Перечень возможных запросов.**

Получить список сделок с указанием контактных данных о клиенте, информации о услуги, сумме сделки и комиссионных компании.

**Цель работы:** приобретение навыков анализа предметной области и построения концептуальной модели.

**Вариант №4. Реализация готовой продукции.**

**Описание предметной области.**

Вы работаете в компании, занимающейся оптово-розничной продажей различных товаров. Вашей задачей является отслеживание финансовой стороны ее работы. Деятельность компании организована следующим образом: компания торгует товарами из определенного спектра. Каждый из этих товаров характеризуется наименованием, оптовой ценой, розничной ценой и справочной информацией. В вашу компанию обращаются покупатели. Для каждого из них вы запоминаете в базе данных стандартные данные (наименование, адрес, телефон, контактное лицо) и составляете по каждой сделке документ, запоминая наряду с покупателем количество купленного им товара и дату покупки.

**Основные сущности.**

Таблица «Клиент»

|  |  |  |
| --- | --- | --- |
| Ключ | Название столбца | Тип данных |
| Первичный ключ | Код клиента | int |
| - | Фамилия | varchar(50) |
| - | Имя | varchar(50) |
| - | Номер телефона | varchar(50) |
| - | Адрес | varcahr(50) |

Таблица «Товар»

|  |  |  |
| --- | --- | --- |
| Ключ | Название столбца | Тип данных |
| Первичный ключ | Код товара | int |
| - | Название | varchar(50) |
| - | Оптовая цена | money |
| - | Розничная цена | money |
| - | Описание | varcahr(MAX) |

Таблица «Сделка»

|  |  |  |
| --- | --- | --- |
| Ключ | Название столбца | Тип данных |
| Первичный ключ | Код сделки | int |
| Внешний ключ | Код товара | int |
| Внешний ключ | Код клиента | int |
| - | Дата продажи | datetime |
| - | Количество | int |
| - | Оптовая продажа | bit |

****

Рисунок 1. ER-диаграмма.

**Перечень возможных запросов.**

Получить список сделок с указанием контактных данных о клиенте, информации о товаре, количестве купленного товара и сумме сделки.

**Цель работы:** ознакомиться со способами создания, модификации, удаления баз.

**Ход работы:**

1. Создание базы данных в MS SQL

Изображение выглядит как текст

Автоматически созданное описание

1. Создание таблицы Customers

Изображение выглядит как стол

Автоматически созданное описание

1. Создание таблицы Orders через запрос

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

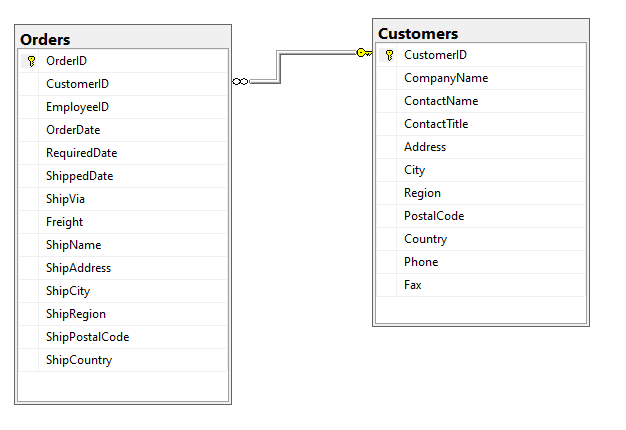
1. Создание диаграммы базы данных

Изображение выглядит как стол

Автоматически созданное описание

1. Создание связей между таблицами

Изображение выглядит как текст, снимок экрана, внутренний

Автоматически созданное описание

**Вывод:** в результате выполнения данной лабораторной работы мы ознакомились с двумя способами создания базы данных, таблиц и диаграмм, а именно – с помощью графического интерфейса, и с помощью T-SQL.

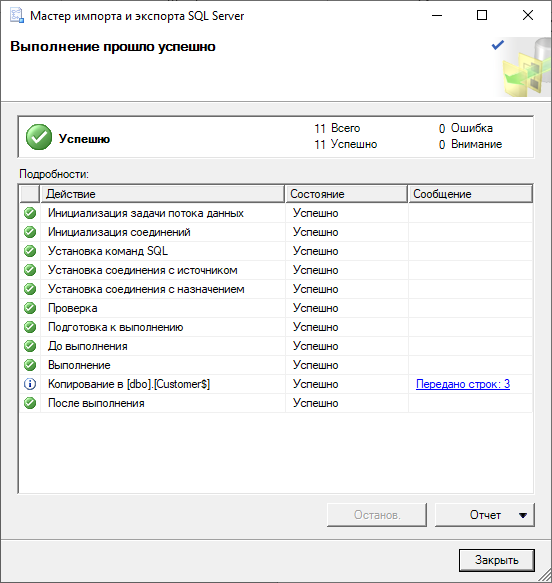
**Цель работы:** освоить методы заполнения таблиц в SQL Server.

**Ход работы:**

1. Импорт с помощью Management Tool

**Изображение выглядит как текст

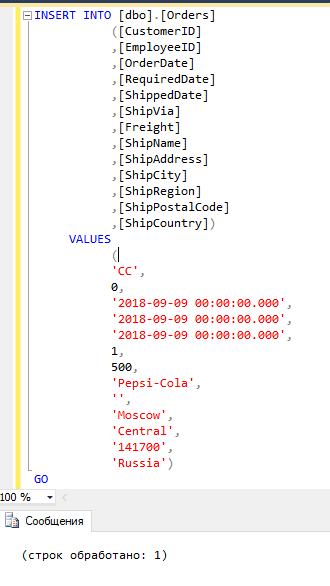
Автоматически созданное описание**

****

**Изображение выглядит как стол

Автоматически созданное описание**

1. Импорт с помощью запроса

****

1. Удаление через запрос

**Изображение выглядит как стол

Автоматически созданное описание**

1. Изменение строк

**Изображение выглядит как стол

Автоматически созданное описание**

**Вывод:** в результате выполнения данной лабораторной работы мы ознакомились со способами импорта данных в таблицы, а также со способами обновления и удаления строк.

**Цель работы:** изучить создание запросов и представлений, использование механизма объединений.

**Ход работы:**

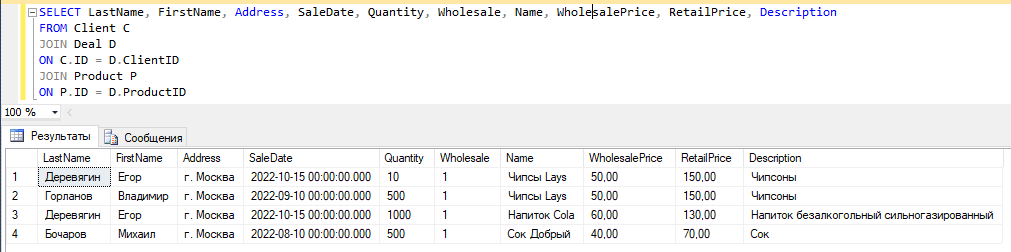
****

Рисунок 1. Вывод всей информации из БД в удобном виде

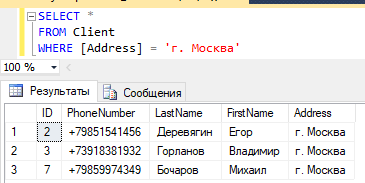


Рисунок 2. Запрос получения клиентов из определённого города

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3. Запрос получения клиентов с именем в определённом буквенном диапазоне

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4. Запрос вывод только оптовых сделок

Изображение выглядит как стол

Автоматически созданное описаниеРисунок 5. Вывод фамилии и имени в верхнем регистре

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6. Вывод клиентов, купившие определённый товар

Изображение выглядит как текст

Автоматически созданное описание  
Рисунок 7. Вывод самого дорогого товара

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8. Количество сделок за последний месяц

**Цель работы:** изучение механизма динамических запросов с использованием хранимых процедур и функций

**Ход работы:**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 1. Печать бланка сделки

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2. Функция расчёта полной стоимости сделки

**Цель работы:** изучение механизмов обеспечения целостности данных SQL

**Ход работы:**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1. Добавление ограничения CHECK

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2. Триггер на добавление записи

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3. Демонстрация работы триггера

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4. Триггер при обновлении данных

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5. Демонстрация работы триггера

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6. Триггер удаления данных

**Цель работы:** изучение механизмов обеспечения целостности данных SQL

**Ход работы:**

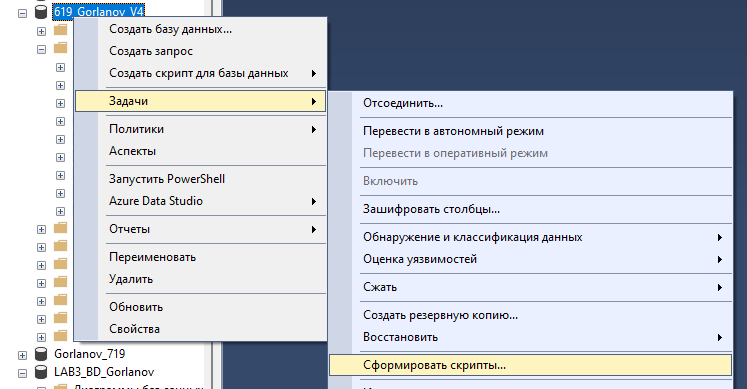


Рисунок 1. Создание скрипта базы данных

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2. Создание скрипта базы данных

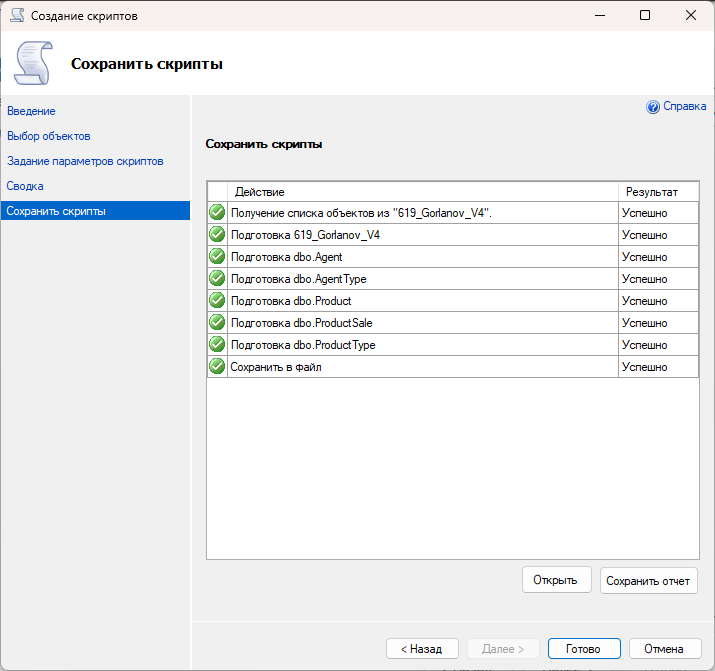


Рисунок 3. Создание скрипта базы данных

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4. Создание резервной копии

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5. Создание резервной копии

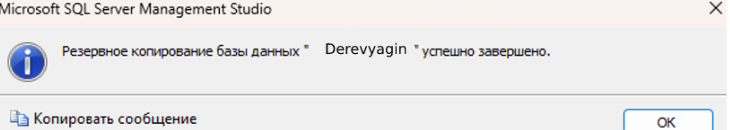
Деревягин Егор Сергеевич

Рисунок 6. Создание резервной копии