Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**КУРСОВОЙ ПРОЕКТ**

Разработка программного модуля «Информационная система для ресторана (организации общественного питания)»

МДК 01.01 «Разработка программных модулей»

**Выполнил:**

Шабалкин Данила Алексеевич \_\_\_

студент группы ИСП.18.1А\_\_\_\_\_\_\_\_\_\_\_\_\_

09.02.07 Информационные системы и программирование)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

очной формы обучения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Руководитель:**

Кузьмина Елена Евгеньевна­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка

\_\_\_\_\_\_\_\_\_/\_\_\_­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Подпись руководителя

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ликино-Дулево

2021 год

**Содержание**

[Введение 3](#_Toc70912527)

[1 Назначение разработки 4](#_Toc70912528)

[2 Требования к программному модулю 5](#_Toc70912529)

[2.1 Требования к функциональным характеристикам 5](#_Toc70912530)

[2.2 Требования к составу и параметрам технических средств 5](#_Toc70912531)

[2.3 Требования к информационной и программной совместимости 5](#_Toc70912532)

[3 Руководство программиста 8](#_Toc70912533)

[3.1 Разработка базы данных 8](#_Toc70912534)

[3.2 Разработка программного модуля 9](#_Toc70912535)

[3.3 Текст программы 34](#_Toc70912536)

[4 Руководство пользователя 42](#_Toc70912537)

[4.1 Выполнение программного модуля 42](#_Toc70912538)

[4.2 Сообщения пользователю 46](#_Toc70912539)

[5. Тестирование и отладка программного модуля 48](#_Toc70912540)

[Заключение 49](#_Toc70912541)

[Список литературы 50](#_Toc70912542)

[Приложения 51](#_Toc70912543)

[Приложение 1 51](#_Toc70912544)

# Введение

Сеть ресторанов «Cantine pas cher» существует с 2017 года и занимает лидирующие позиции в среднеценовом ресторанном сегменте.

Автоматизация ресторанов (предприятий общественного питания) — процесс внедрения программно-аппаратных комплексов автоматизации бизнес-процессов на предприятиях общественного питания.

Сочетание «автоматизация ресторанов» сформировалось в устойчивый термин в среде русскоязычных рестораторов, поставщиков оборудования и ПО. В нём имеются сразу и обобщение, и наоборот узкая трактовка смысла составляющих слов. В данной статье под термином «ресторан» подразумевается обобщенное понятие: любое заведение общепита. При этом, под предприятием общественного питания может пониматься также сеть заведений.

«Автоматизация» в данном случае, напротив, трактуется ограниченно и не затрагивает непосредственно производственные процессы (приготовление пищи), а касается только информационной составляющей бизнеса — учёт, документооборот, продажи и другие процессы, связанные с обработкой данных.

Программа, для которой пишется данная документация предназначена для автоматизации ресторанного учета, а именно – просмотра меню незарегистрированными пользователями, заказа и оплаты блюд из меню авторизованным пользователям, соответственно обеспечивать возможность регистрации и авторизации.

# 1 Назначение разработки

Создать информационную систему для автоматизации предприятия общественного питания. Программа обязана давать пользователю смотреть информацию о блюде, заказывать блюда если он авторизирован, авторизоваться или регистрироваться в системе приложения. Пользователь также может отменять заказ блюда, если он этого хочет, оплачивать блюдо, иметь возможность выйти из системы.

Список страниц приложения:

* Главная – отображает приветствие для пользователя
* Сделать заказ – страница отображает список блюд, доступных для заказа. Заказ можно сделать только если пользователь авторизован
* Вход – страница входа в аккаунт. Содержит поля ввода телефона, пароля, а также кнопки «Вход» и «Регистрация»
* Регистрация – позволяет зарегистрироваться пользователю, если у него нет аккаунта в системе.
* Заказы – отображает заказы, сделанные пользователем и позволяет оплатить или отменить сделанный заказ

Функциональная схема приложения вынесена в [приложение 1](#_Приложение_1).

# 2 Требования к программному модулю

## 2.1 Требования к функциональным характеристикам

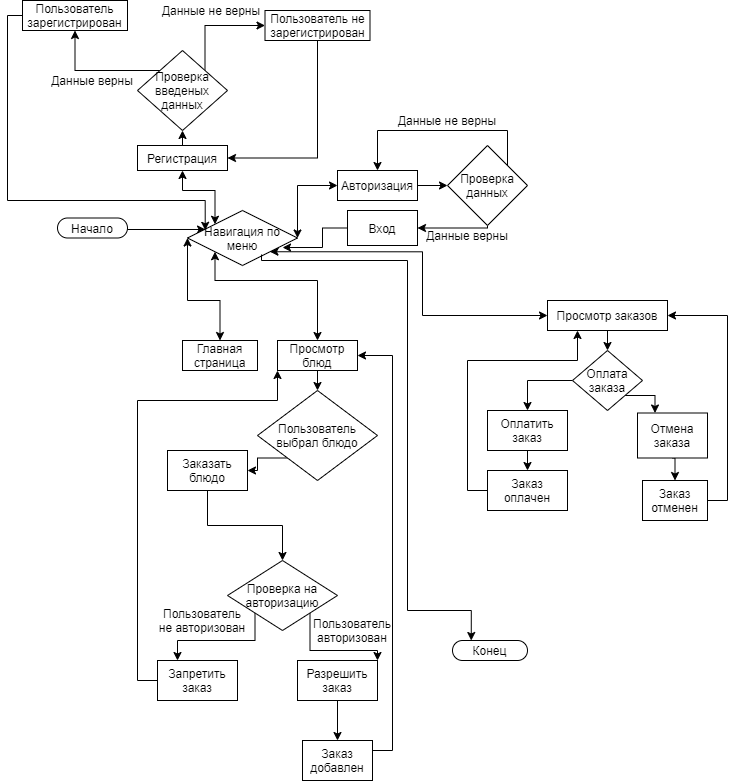


Рис. 1 «Блок-схема приложения»

## 2.2 Требования к составу и параметрам технических средств

Компьютер, со следующими системными характеристиками:

|  |  |
| --- | --- |
| Процессор | Intel Core 2 Quad q6600 |
| ОЗУ | 512 мб |
| Видеоадаптер | Любой адаптер |
| Доступ к сети | Стабильное подключение к сети |
| Разрешение экрана | От 1280х720 |
| Дисковое пространство | Не менее 200 мб |
| Устройства ввода | Мышь, клавиатура |

## 2.3 Требования к информационной и программной совместимости

* Microsoft SQL Server Management Studio – утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера.
* SQLExpress – это версия системы управления реляционными базами данных Microsoft SQL Server, которую можно бесплатно загружать, распространять и использовать. Он содержит базу данных, специально предназначенную для встроенных и небольших приложений.
* Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств.
* .NET Framework 4.5 – программная платформа, выпущенная компанией Microsoft в 2002 году. Основой платформы является общеязыковая среда исполнения Common Language Runtime, которая подходит для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду.
* Entity Framework – объектно-ориентированная технология доступа к данным, является object-relational mapping решением для .NET Framework от Microsoft. Предоставляет возможность взаимодействия с объектами как посредством LINQ в виде LINQ to Entities, так и с использованием Entity SQL.
* Xceed WPF Toolkit – набор компонентов управления WPF для создания приложений Windows нового поколения.
* Язык программирования C# – объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework.
* Microsoft Office, а именно:
  + 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.
  + Excel – программа для работы с электронными таблицами, созданная корпорацией Microsoft.
  + PowerPoint – программа подготовки презентаций и просмотра презентаций, являющаяся частью Microsoft Office.
* GitHub – крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc.
* Sublime Text – проприетарный текстовый редактор. Поддерживает плагины на языке программирования Python. Разработчик позволяет бесплатно и без ограничений ознакомиться с продуктом, однако программа уведомляет о необходимости приобретения лицензии.
* Visual Studio Code – редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений.
* Microsoft Windows 10 – операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT. После Windows 8.1 система получила номер 10, минуя 9. Серверный аналог Windows 10 — Windows Server 2016.

# 3 Руководство программиста

## 3.1 Разработка базы данных

Создание БД:

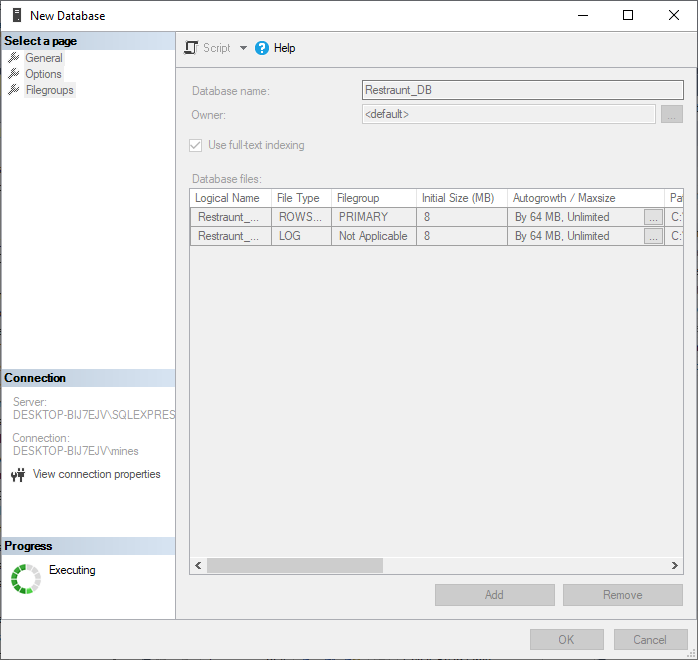


Рис. 2 «Создание базы данных»

Заполнение таблиц:

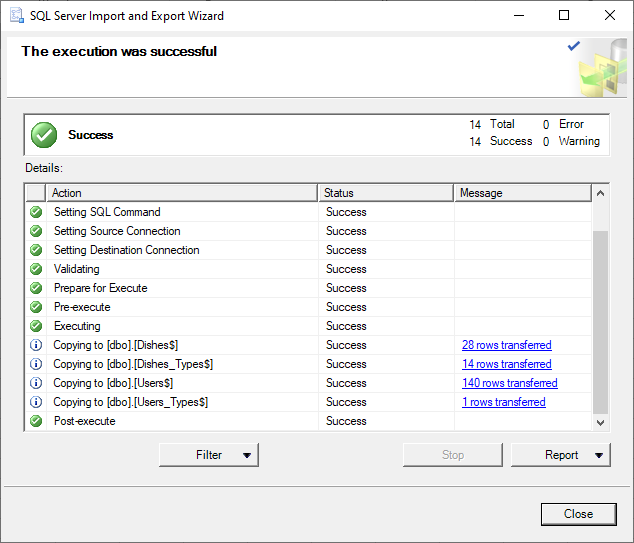


Рис. 3 «Импорт данных из Excel»

Создание таблиц:

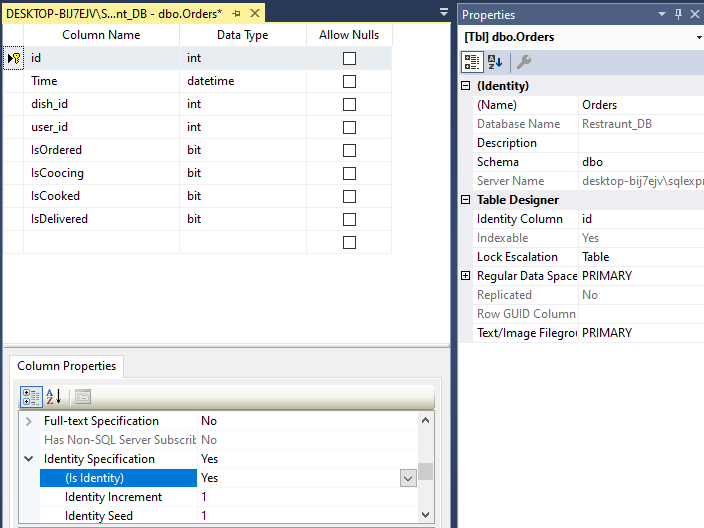


Рис. 4 «Создание таблицы с помощью средств SSMS»

## 3.2 Разработка программного модуля

Создание проекта в Visual Studio:

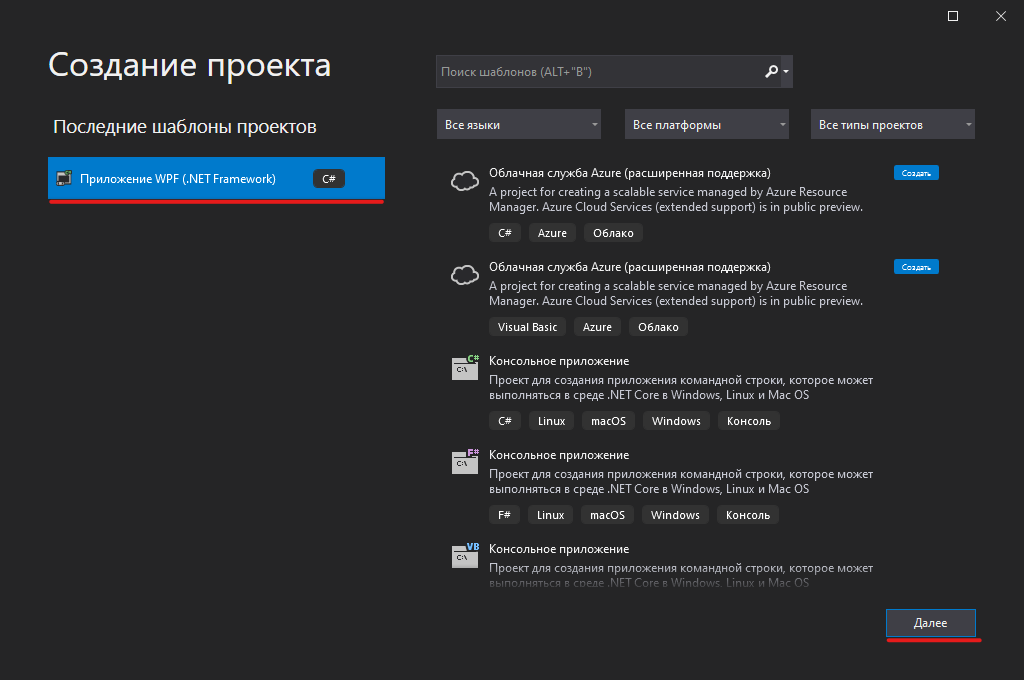


Рис. 5 «Создание проекта в VS»

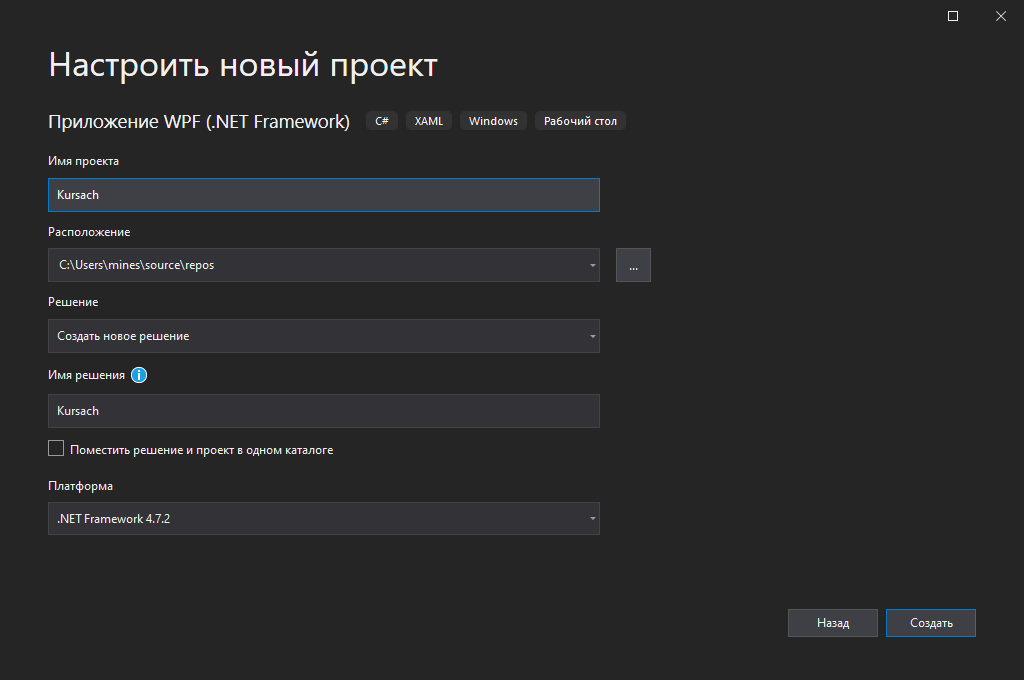


Рис. 6 «Выбор пути и имени проекта»

Использование Entity Framework – добавление ORM-модели в проект приложения:

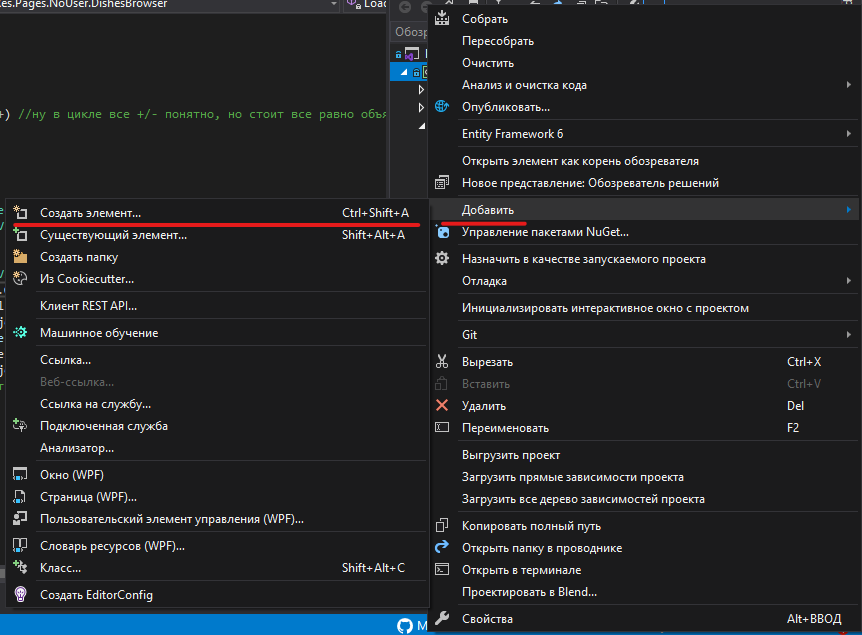


Рис. 7 «Создание нового объекта»

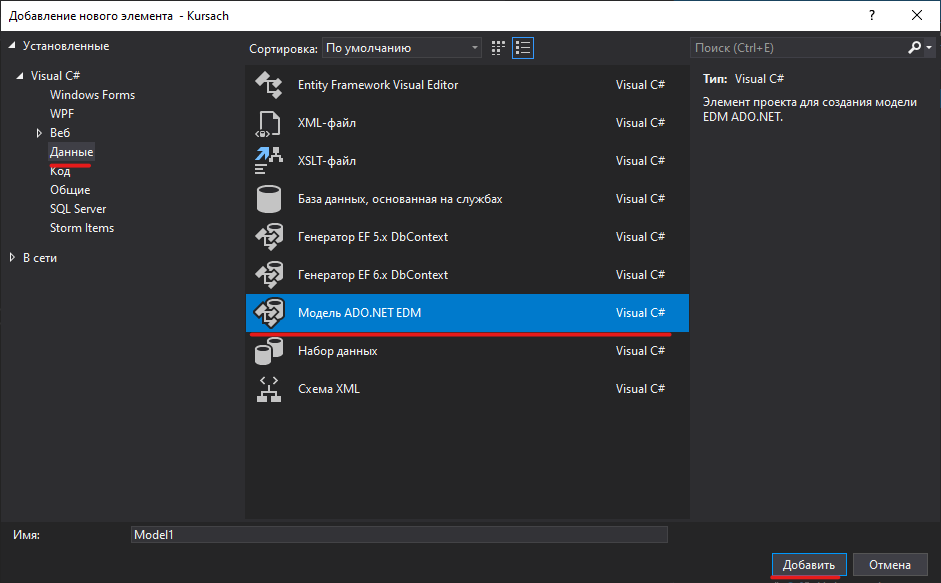


Рис. 8 «Создание модели в проекте»

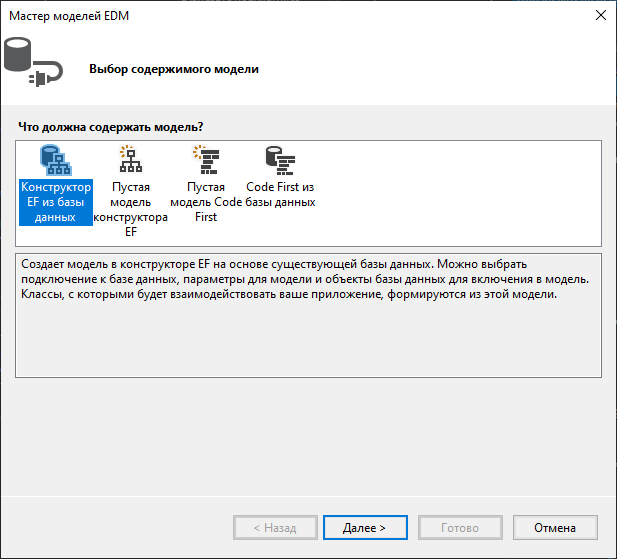


Рис. 9 «Выбор типа модели»

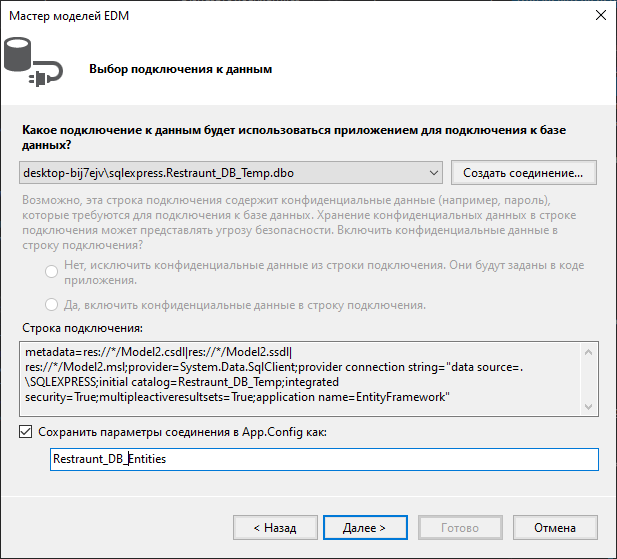


Рис. 10 «Задание имени модели»

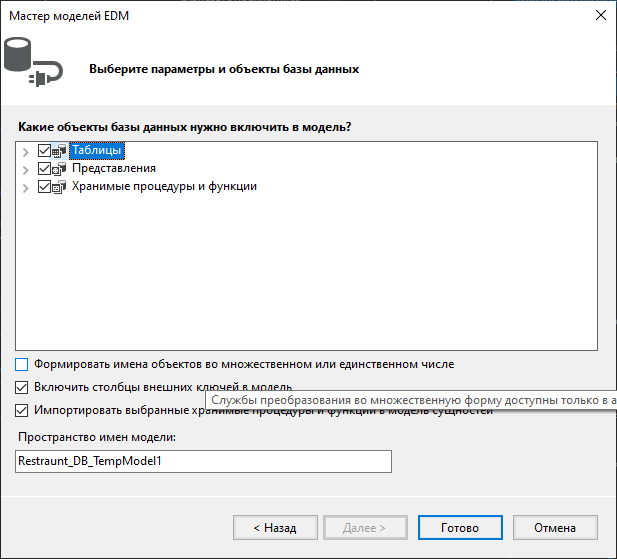


Рисунок 11 «Выбор объектов на импорт»

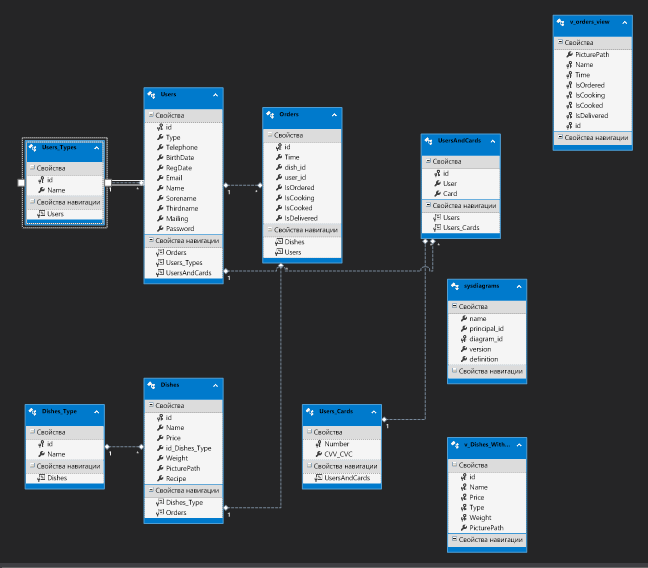


Рис. 12 «Структура модели данных»

**Создание страниц**

MainWindow.xaml

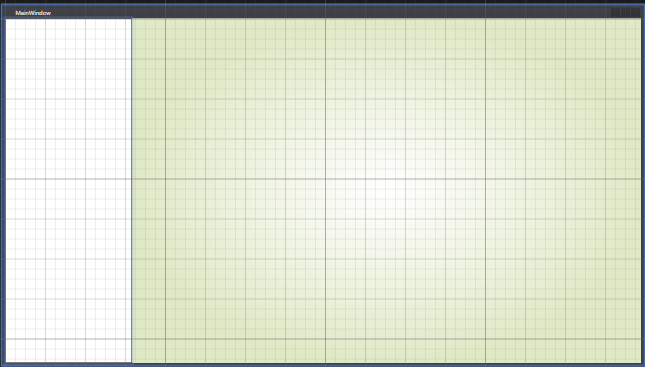


Рис. 13 «Макет окна приложения»

<Window x:Class="Kursach.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:Kursach"

mc:Ignorable="d"

Title="MainWindow" Height="720" Width="1280"

Closing="Window\_Closing"

>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="5\*"/>

<ColumnDefinition Width="20\*"/>

</Grid.ColumnDefinitions>

<Frame x:Name="fMain"

NavigationUIVisibility=

"Hidden" Grid.Column="1"

>

<Frame.Background>

<RadialGradientBrush>

<GradientStop Color="#FFE0E9C6" Offset="1"/>

<GradientStop Color="White"/>

</RadialGradientBrush>

</Frame.Background>

</Frame >

<Frame x:Name="fButtons"

BorderBrush="Black"

BorderThickness="1"

NavigationUIVisibility="Hidden"/>

</Grid>

</Window>

Button\_Guest.xaml

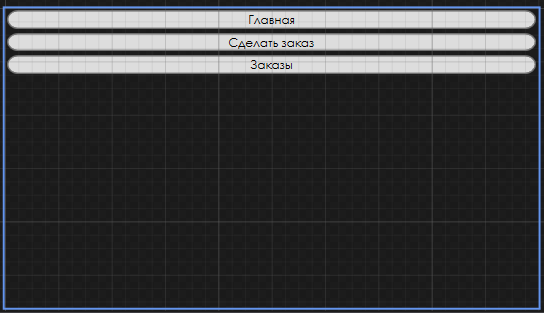


Рис. 14 «Макет кнопок меню авториз. пользователя»

<Page x:Class="Kursach.Res.Pages.Buttons.Buttons\_Guest"

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:Kursach.Res.Pages.Buttons"

mc:Ignorable="d"

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

Title="Buttons\_Guest">

<Grid>

<StackPanel>

<Button x:Name="btnMain" Content="Главная" Click="btnMain\_Click" Style="{DynamicResource MenuButtons}"/>

<Button x:Name="btnDishes" Content="Сделать заказ" Click="btnDishes\_Click" Style="{DynamicResource MenuButtons}"/>

<Button x:Name="btnOrders" Content="Заказы" Click="btnOrders\_Click" Style="{DynamicResource MenuButtons}"/>

<Label x:Name="lNotifier"/>

</StackPanel>

</Grid>

</Page>

ButtonsNoUser.xaml

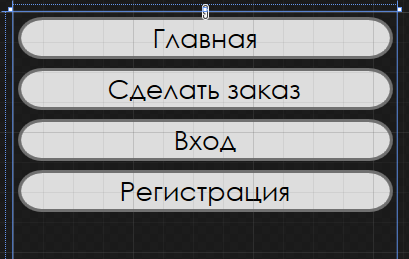


Рис. 15 «Макет кнопок неавтор. пользователя»

<Page x:Class="Kursach.Res.Pages.Buttons.ButtonsNoUser"

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:Kursach.Res.Pages.Buttons"

mc:Ignorable="d"

d:DesignHeight="720" d:DesignWidth="256"

Title="ButtonsNoUser">

<Grid>

<StackPanel x:Name="spMain">

<Button x:Name="btnMain" Content="Главная" Click="btnMain\_Click" Style="{StaticResource MenuButtons}"/>

<Button x:Name="btnDishesView" Content="Сделать заказ" Click="btnDishesView\_Click" Style="{StaticResource MenuButtons}"/>

<Button x:Name="btnLogin" Content="Вход" Click="btnLogin\_Click" Style="{StaticResource MenuButtons}"/>

<Button x:Name="btnReg" Content="Регистрация" Click="btnReg\_Click" Style="{StaticResource MenuButtons}"/>

</StackPanel>

</Grid>

</Page>

Main.xaml (Guest)

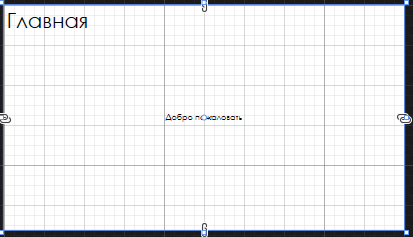


Рис. 16 «Макет гл. окна авториз. пользователя»

<Page x:Class="Kursach.Res.Pages.Guest.Main"

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:Kursach.Res.Pages.Guest"

mc:Ignorable="d"

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

Title="Main">

<Grid>

<Label Content="Главная" FontSize="42" Width="4000"/>

<Label Content="Добро пожаловать" HorizontalAlignment="Center" VerticalAlignment="Center"/>

</Grid>

</Page>

Orders.xaml

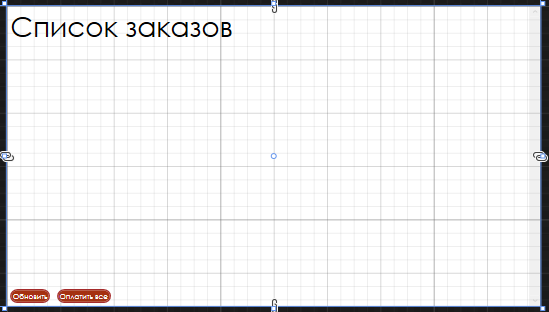


Рис. 17 «Макет списка заказов»

<Page x:Class="Kursach.Res.Pages.Guest.Orders"

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:Kursach.Res.Pages.Guest"

mc:Ignorable="d"

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

Title="Orders">

<Grid Background="White">

<ScrollViewer>

<Grid>

<WrapPanel x:Name="wpOrders">

<Label Content="Список заказов" FontSize="42" Width="4000"/>

</WrapPanel>

<StackPanel VerticalAlignment="Bottom" Orientation="Horizontal">

<Button x:Name="btnUpdate" Content="Oбновить" Style="{DynamicResource ViewsButtons}" Click="Button\_Click"/>

<Button Content="Оплатить все" Style="{DynamicResource ViewsButtons}" x:Name="btnPayAll" Click="btnPayAll\_Click"/>

</StackPanel>

</Grid>

</ScrollViewer>

</Grid>

</Page>

DishesBrowser

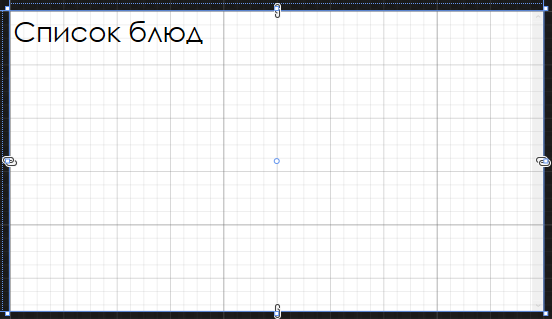


Рисунок 18 «Макет списка блюд»

<Page x:Class="Kursach.Res.Pages.NoUser.DishesBrowser"

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:Kursach.Res.Pages.NoUser"

mc:Ignorable="d"

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

Title="DishesBrowser">

<Grid Background="White">

<ScrollViewer>

<WrapPanel x:Name="wpDishes">

<Label Content="Список блюд" FontSize="42" Width="4000"/>

</WrapPanel>

</ScrollViewer>

</Grid>

</Page>

Login.xaml

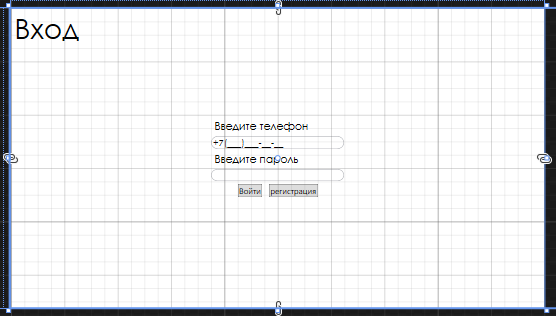


Рисунок 19 «Макет страницы входа»

<Page x:Class="Kursach.Res.Pages.NoUser.Login"

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:Kursach.Res.Pages.NoUser"

xmlns:e="http://schemas.xceed.com/wpf/xaml/toolkit"

mc:Ignorable="d"

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

Title="Login">

<Grid Background="White">

<Label Content="Вход" FontSize="42" Width="4000"/>

<StackPanel Orientation="Vertical" Grid.Row="1" Grid.Column="1" VerticalAlignment="Center">

<Label Content="Введите телефон" Margin="300,0" />

<e:MaskedTextBox x:Name="tbLogin" Width="200" Mask="+7(000)000-00-00"/>

<Label Content="Введите пароль" Margin="300,0"/>

<PasswordBox x:Name="pbPassword" Width="200"/>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" >

<Button x:Name="btnLogin" Content="Войти" Margin="5" Click="btnLogin\_Click"/>

<Button x:Name="btnRegistration" Content="регистрация" Margin="5"/>

</StackPanel>

</StackPanel>

</Grid>

</Page>

Main.xaml (NoUser)

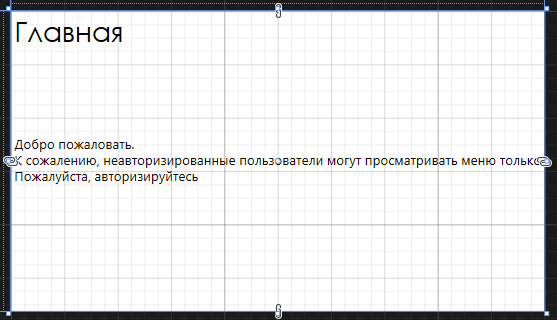


Рис. 20 «Макет гл. страницы неавтор. пользователя»

<Page x:Class="Kursach.Res.Pages.NoUser.Main"

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:Kursach.Res.InterfaceObjects"

mc:Ignorable="d"

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

Title="Main">

<Grid Background="White">

<Label Content="Главная" FontSize="42" Width="4000"/>

<Label Content="Добро пожаловать.&#13;К сожалению, неавторизированные пользователи могут просматривать меню только в бумажном виде.&#13;Пожалуйста, авторизируйтесь"

HorizontalAlignment="Center"

VerticalAlignment="Center" FontSize="20" FontFamily="Gothic sans"

Hyperlink.Click="Label\_Click"/>

</Grid>

</Page>

Registration.xaml

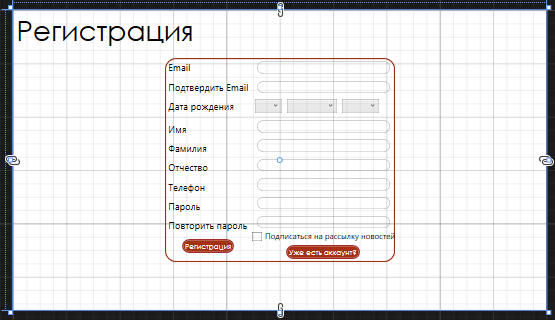


Рис. 21 «Макет страницы регистрации»

<Page x:Class="Kursach.Res.Pages.NoUser.Registration"

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:Kursach.Res.Pages.NoUser"

xmlns:e="http://schemas.xceed.com/wpf/xaml/toolkit"

mc:Ignorable="d"

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

Title="Registration">

<Grid Background="White">

<Label Content="Регистрация" FontSize="42" Width="4000"/>

<UserControl>

<Grid HorizontalAlignment="Center" VerticalAlignment="Center">

<Border>

<Border BorderBrush="#9d2d0f" BorderThickness="2" CornerRadius="15" Background="White"/>

</Border>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<StackPanel x:Name="labels" Orientation="Vertical">

<Label Content="Email" Style="{DynamicResource RegLabels}"/>

<Label Content="Подтвердить Email" Style="{DynamicResource RegLabels}"/>

<Label Content="Дата рождения" Style="{DynamicResource RegLabels}" Margin="0,0,0,5"/>

<Label Content="Имя" Style="{DynamicResource RegLabels}"/>

<Label Content="Фамилия" Style="{DynamicResource RegLabels}"/>

<Label Content="Отчество" Style="{DynamicResource RegLabels}"/>

<Label Content="Телефон" Style="{DynamicResource RegLabels}"/>

<Label Content="Пароль" Style="{DynamicResource RegLabels}"/>

<Label Content="Повторить пароль" Style="{DynamicResource RegLabels}"/>

<Button x:Name="btnRegistration" Click="btnRegistration\_Click" Content="Регистрация" HorizontalAlignment="Center" Style="{DynamicResource ViewsButtons}"/>

</StackPanel>

<StackPanel x:Name="tbs" Orientation="Vertical">

<e:MaskedTextBox x:Name="mtbEmail" Width="200" Margin="5,5,5,5"/>

<e:MaskedTextBox x:Name="mtbEmailConfirm" Width="200" Margin="5"/>

<StackPanel Orientation="Horizontal">

<ComboBox x:Name="cbDate" Width="40" Margin="5,3,3,5"/>

<ComboBox x:Name="cbMonth" SelectionChanged="cbMonth\_SelectionChanged" Width="75" Margin="5,3,3,5"/>

<ComboBox x:Name="cbYear" SelectionChanged="cbYear\_SelectionChanged" Width="55" Margin="5,3,3,5"/>

</StackPanel>

<e:MaskedTextBox x:Name="mtbName" Width="200" Margin="5"/>

<e:MaskedTextBox x:Name="mtbSurename" Width="200" Margin="5"/>

<e:MaskedTextBox x:Name="mtbThirdname" Width="200" Margin="5"/>

<e:MaskedTextBox x:Name="mtbTelephone" Width="200" Margin="5"/>

<PasswordBox x:Name="pbPassword" Width="200" Margin="5"/>

<PasswordBox x:Name="pbPasswordConfirm" Width="200" Margin="5"/>

<CheckBox x:Name="IsSubscribed" Content="Подписаться на рассылку новостей"/>

<Button x:Name="btnLogin" Click="btnLogin\_Click" Content="Уже есть аккаунт?" HorizontalAlignment="Center" Style="{DynamicResource ViewsButtons}"/>

</StackPanel>

</StackPanel>

</Grid>

</UserControl>

</Grid>

</Page>

Создание элементов управления

DishView.xaml

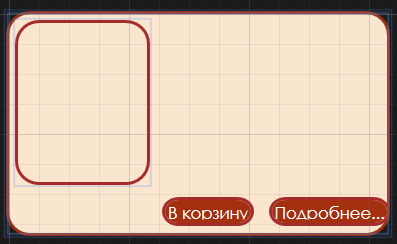


Рис. 22 «Макет элемента просмотра блюд»

<UserControl x:Class="Kursach.Res.InterfaceObjects.DishView"

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:Kursach.Res.InterfaceObjects"

xmlns:e="http://schemas.xceed.com/wpf/xaml/toolkit"

mc:Ignorable="d"

d:DesignHeight="150" d:DesignWidth="256"

Style="{DynamicResource UserControlStyle1}"

>

<UserControl.Resources>

<Style TargetType="Label">

<Setter Property="FontFamily" Value="Century Gothic"/>

</Style>

<Style x:Key="UserControlStyle1" TargetType="{x:Type UserControl}">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type UserControl}">

<Border CornerRadius="15" BorderBrush="#9d2d0f"

BorderThickness="2" Background="#fae5cf"

Padding="{TemplateBinding Padding}"

SnapsToDevicePixels="true">

<ContentPresenter HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}"

SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}"

VerticalAlignment="{TemplateBinding VerticalContentAlignment}"/>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="FocusVisual">

<Setter Property="Control.Template">

<Setter.Value>

<ControlTemplate>

<Rectangle Margin="2" SnapsToDevicePixels="true" Stroke="{DynamicResource {x:Static SystemColors.ControlTextBrushKey}}" StrokeThickness="1" StrokeDashArray="1 2"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<SolidColorBrush x:Key="Button.Static.Background" Color="#782009"/>

<SolidColorBrush x:Key="Button.Static.Border" Color="#782009"/>

<SolidColorBrush x:Key="Button.MouseOver.Background" Color="#592009"/>

<SolidColorBrush x:Key="Button.MouseOver.Border" Color="#592009"/>

<SolidColorBrush x:Key="Button.Pressed.Background" Color="#FFC4E5F6"/>

<SolidColorBrush x:Key="Button.Pressed.Border" Color="#FF2C628B"/>

<SolidColorBrush x:Key="Button.Disabled.Background" Color="#FFF4F4F4"/>

<SolidColorBrush x:Key="Button.Disabled.Border" Color="#FFADB2B5"/>

<SolidColorBrush x:Key="Button.Disabled.Foreground" Color="#FF838383"/>

<Style TargetType="{x:Type Button}">

<Setter Property="FocusVisualStyle" Value="{StaticResource FocusVisual}"/>

<Setter Property="Background" Value="{StaticResource Button.Static.Background}"/>

<Setter Property="BorderBrush" Value="#782009"/>

<Setter Property="Foreground" Value="{DynamicResource {x:Static SystemColors.ControlTextBrushKey}}"/>

<Setter Property="BorderThickness" Value="2"/>

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

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

<Setter Property="Padding" Value="1"/>

<Setter Property="FontFamily" Value="Century Gothic"/>

<Setter Property="Padding" Value="2"/>

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

<Setter Property="Background" Value="#a53010"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type Button}">

<Border x:Name="border" CornerRadius="12" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" SnapsToDevicePixels="true">

<ContentPresenter x:Name="contentPresenter" Focusable="False" HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}" Margin="{TemplateBinding Padding}" RecognizesAccessKey="True" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="{TemplateBinding VerticalContentAlignment}"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsDefaulted" Value="true">

<Setter Property="BorderBrush" TargetName="border" Value="{DynamicResource {x:Static SystemColors.HighlightBrushKey}}"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="true">

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.MouseOver.Background}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.MouseOver.Border}"/>

</Trigger>

<Trigger Property="IsPressed" Value="true">

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.Pressed.Background}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.Pressed.Border}"/>

</Trigger>

<Trigger Property="IsEnabled" Value="false">

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.Disabled.Background}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.Disabled.Border}"/>

<Setter Property="TextElement.Foreground" TargetName="contentPresenter" Value="{StaticResource Button.Disabled.Foreground}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</UserControl.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="4\*"/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="5\*"/>

<ColumnDefinition Width="8\*"/>

</Grid.ColumnDefinitions>

<UserControl Width="90" Height="110" Margin="3">

<Grid>

<Image x:Name="iPic" Width="90" Height="110" Stretch="Fill"/>

<Border>

<Border x:Name="b" CornerRadius="15" BorderBrush="Brown" BorderThickness="2" Background="Transparent"/>

</Border>

</Grid>

</UserControl>

<StackPanel Grid.Column="1">

<Label x:Name="lName" FontSize="18"/>

<Label x:Name="lType" FontSize="14"/>

<Label x:Name="lWeight" FontSize="14"/>

<Label x:Name="lNoty" FontSize="14"/>

</StackPanel>

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

<Button x:Name="btnOrder" Style="{DynamicResource ViewsButtons}" Click="btnOrder\_Click" Content="В корзину"></Button>

<Button x:Name="btnDetails" Style="{DynamicResource ViewsButtons}" Content="Подробнее..." HorizontalAlignment="Right"/>

</StackPanel>

<Label x:Name="lPrice" FontSize="18" Grid.Row="1"/>

</Grid>

</UserControl>

OrderView.xaml

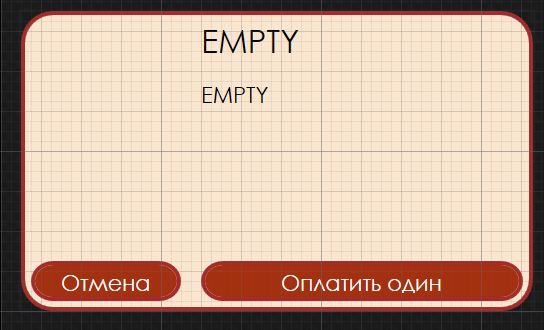


Рис. 23 «Макет элемента просмотра заказов»

<UserControl x:Class="Kursach.Res.InterfaceObjects.OrderView"

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:Kursach.Res.InterfaceObjects"

mc:Ignorable="d"

Height="150"

Width="256"

Margin="10"

>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="4\*"/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="4\*"/>

</Grid.ColumnDefinitions>

<Border Grid.ColumnSpan="2" Grid.RowSpan="2">

<Border Background="#fae5cf" BorderThickness="2" CornerRadius="15" BorderBrush="Brown"/>

</Border>

<Image x:Name="iImage" Margin="8,8,0,0"/>

<Button x:Name="btnCancel"

Click="btnCancel\_Click"

Grid.Row="1"

Style="{DynamicResource ViewsButtons}"

Content="Отмена"/>

<Button x:Name="btnPay"

Click="btnPay\_Click"

Grid.Column="1" Grid.Row="1"

Style="{DynamicResource ViewsButtons}"

Content="Оплатить один"/>

<StackPanel Orientation="Vertical" Grid.Column="1">

<Label x:Name="lName" Content="EMPTY"/>

<Label x:Name="lConditionIndicator" Content="EMPTY" FontSize="11"/>

</StackPanel>

</Grid>

</UserControl>

App.xaml

<Application

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

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

xmlns:local="clr-namespace:Kursach.Res.InterfaceObjects"

xmlns:e="http://schemas.xceed.com/wpf/xaml/toolkit"

xmlns:Themes="clr-namespace:Microsoft.Windows.Themes;assembly=PresentationFramework.Aero2" x:Class="Kursach.App"

StartupUri="MainWindow.xaml">

<Application.Resources>

<!--Menu buttons style-->

<Style x:Key="MenuButtons" TargetType="{x:Type Button}">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type Button}">

<Border x:Name="border"

CornerRadius="15"

BorderBrush="{TemplateBinding BorderBrush}"

BorderThickness="2"

Background="{TemplateBinding Background}"

SnapsToDevicePixels="true">

<ContentPresenter x:Name="contentPresenter"

Focusable="False"

HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}"

Margin="{TemplateBinding Padding}"

RecognizesAccessKey="True"

SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}"

VerticalAlignment="{TemplateBinding VerticalContentAlignment}"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="BorderBrush" TargetName="border" Value="Green"/>

<Setter Property="Background" TargetName="border" Value="#eff3e1"/>

</Trigger>

<Trigger Property="IsFocused" Value="True">

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

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

<Setter Property="Margin" Value="3"/>

<Setter Property="FontFamily" Value="Century Gothic"/>

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

</Style>

<!--#region Garbage-->

<!--Стиль для всех текстбоксов с маской-->

<Style TargetType="{x:Type e:MaskedTextBox}">

<Setter Property="FontFamily" Value="Century Gothic"/>

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

<Setter Property="Background" Value="{DynamicResource {x:Static SystemColors.WindowBrushKey}}"/>

<Setter Property="BorderBrush" Value="#FFABADB3"/>

<Setter Property="Foreground" Value="{DynamicResource {x:Static SystemColors.ControlTextBrushKey}}"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="KeyboardNavigation.TabNavigation" Value="None"/>

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

<Setter Property="FocusVisualStyle" Value="{x:Null}"/>

<Setter Property="AllowDrop" Value="True"/>

<Setter Property="ScrollViewer.PanningMode" Value="VerticalFirst"/>

<Setter Property="Stylus.IsFlicksEnabled" Value="False"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type e:MaskedTextBox}">

<Border x:Name="border" CornerRadius="10" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" SnapsToDevicePixels="True">

<ScrollViewer x:Name="PART\_ContentHost" Focusable="False" HorizontalScrollBarVisibility="Hidden" VerticalScrollBarVisibility="Hidden"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsEnabled" Value="False">

<Setter Property="Opacity" TargetName="border" Value="0.56"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="BorderBrush" TargetName="border" Value="#FF7EB4EA"/>

</Trigger>

<Trigger Property="IsKeyboardFocused" Value="True">

<Setter Property="BorderBrush" TargetName="border" Value="#FF569DE5"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

<MultiTrigger>

<MultiTrigger.Conditions>

<Condition Property="IsInactiveSelectionHighlightEnabled" Value="True"/>

<Condition Property="IsSelectionActive" Value="False"/>

</MultiTrigger.Conditions>

<Setter Property="SelectionBrush" Value="{DynamicResource {x:Static SystemColors.InactiveSelectionHighlightBrushKey}}"/>

</MultiTrigger>

</Style.Triggers>

</Style>

<!--All labels style-->

<Style TargetType="{x:Type Label}">

<Setter Property="FontFamily" Value="Century Gothic"/>

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

</Style>

<SolidColorBrush x:Key="TextBox.Static.Border" Color="#FFABAdB3"/>

<SolidColorBrush x:Key="TextBox.MouseOver.Border" Color="#FF7EB4EA"/>

<SolidColorBrush x:Key="TextBox.Focus.Border" Color="#FF569DE5"/>

<!--All Password box styles-->

<Style TargetType="{x:Type PasswordBox}">

<Setter Property="PasswordChar" Value="•"/>

<Setter Property="Background" Value="{DynamicResource {x:Static SystemColors.WindowBrushKey}}"/>

<Setter Property="BorderBrush" Value="{StaticResource TextBox.Static.Border}"/>

<Setter Property="Foreground" Value="{DynamicResource {x:Static SystemColors.ControlTextBrushKey}}"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="KeyboardNavigation.TabNavigation" Value="None"/>

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

<Setter Property="FocusVisualStyle" Value="{x:Null}"/>

<Setter Property="AllowDrop" Value="true"/>

<Setter Property="ScrollViewer.PanningMode" Value="VerticalFirst"/>

<Setter Property="Stylus.IsFlicksEnabled" Value="False"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type PasswordBox}">

<Border x:Name="border" CornerRadius="9" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" SnapsToDevicePixels="True">

<ScrollViewer x:Name="PART\_ContentHost" Focusable="false" HorizontalScrollBarVisibility="Hidden" VerticalScrollBarVisibility="Hidden"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsEnabled" Value="false">

<Setter Property="Opacity" TargetName="border" Value="0.56"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="true">

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource TextBox.MouseOver.Border}"/>

</Trigger>

<Trigger Property="IsKeyboardFocused" Value="true">

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource TextBox.Focus.Border}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

<MultiTrigger>

<MultiTrigger.Conditions>

<Condition Property="IsInactiveSelectionHighlightEnabled" Value="true"/>

<Condition Property="IsSelectionActive" Value="false"/>

</MultiTrigger.Conditions>

<Setter Property="SelectionBrush" Value="{DynamicResource {x:Static SystemColors.InactiveSelectionHighlightBrushKey}}"/>

</MultiTrigger>

</Style.Triggers>

</Style>

<Style x:Key="FocusVisual">

<Setter Property="Control.Template">

<Setter.Value>

<ControlTemplate>

<Rectangle StrokeDashArray="1 2" StrokeThickness="1" Stroke="{DynamicResource {x:Static SystemColors.ControlTextBrushKey}}" SnapsToDevicePixels="true" Margin="2"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<LinearGradientBrush x:Key="ComboBox.Static.Background" EndPoint="0,1" StartPoint="0,0">

<GradientStop Color="#FFF0F0F0" Offset="0.0"/>

<GradientStop Color="#FFE5E5E5" Offset="1.0"/>

</LinearGradientBrush>

<SolidColorBrush x:Key="ComboBox.Static.Border" Color="#FFACACAC"/>

<SolidColorBrush x:Key="ComboBox.Static.Editable.Background" Color="#FFFFFFFF"/>

<SolidColorBrush x:Key="ComboBox.Static.Editable.Border" Color="#FFABADB3"/>

<SolidColorBrush x:Key="ComboBox.Static.Editable.Button.Background" Color="Transparent"/>

<SolidColorBrush x:Key="ComboBox.Static.Editable.Button.Border" Color="Transparent"/>

<SolidColorBrush x:Key="ComboBox.MouseOver.Glyph" Color="#FF000000"/>

<LinearGradientBrush x:Key="ComboBox.MouseOver.Background" EndPoint="0,1" StartPoint="0,0">

<GradientStop Color="#FFECF4FC" Offset="0.0"/>

<GradientStop Color="#FFDCECFC" Offset="1.0"/>

</LinearGradientBrush>

<SolidColorBrush x:Key="ComboBox.MouseOver.Border" Color="#FF7EB4EA"/>

<SolidColorBrush x:Key="ComboBox.MouseOver.Editable.Background" Color="#FFFFFFFF"/>

<SolidColorBrush x:Key="ComboBox.MouseOver.Editable.Border" Color="#FF7EB4EA"/>

<LinearGradientBrush x:Key="ComboBox.MouseOver.Editable.Button.Background" EndPoint="0,1" StartPoint="0,0">

<GradientStop Color="#FFEBF4FC" Offset="0.0"/>

<GradientStop Color="#FFDCECFC" Offset="1.0"/>

</LinearGradientBrush>

<SolidColorBrush x:Key="ComboBox.MouseOver.Editable.Button.Border" Color="#FF7EB4EA"/>

<SolidColorBrush x:Key="ComboBox.Pressed.Glyph" Color="#FF000000"/>

<LinearGradientBrush x:Key="ComboBox.Pressed.Background" EndPoint="0,1" StartPoint="0,0">

<GradientStop Color="#FFDAECFC" Offset="0.0"/>

<GradientStop Color="#FFC4E0FC" Offset="1.0"/>

</LinearGradientBrush>

<SolidColorBrush x:Key="ComboBox.Pressed.Border" Color="#FF569DE5"/>

<SolidColorBrush x:Key="ComboBox.Pressed.Editable.Background" Color="#FFFFFFFF"/>

<SolidColorBrush x:Key="ComboBox.Pressed.Editable.Border" Color="#FF569DE5"/>

<LinearGradientBrush x:Key="ComboBox.Pressed.Editable.Button.Background" EndPoint="0,1" StartPoint="0,0">

<GradientStop Color="#FFDAEBFC" Offset="0.0"/>

<GradientStop Color="#FFC4E0FC" Offset="1.0"/>

</LinearGradientBrush>

<SolidColorBrush x:Key="ComboBox.Pressed.Editable.Button.Border" Color="#FF569DE5"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Glyph" Color="#FFBFBFBF"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Background" Color="#FFF0F0F0"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Border" Color="#FFD9D9D9"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Editable.Background" Color="#FFFFFFFF"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Editable.Border" Color="#FFBFBFBF"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Editable.Button.Background" Color="Transparent"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Editable.Button.Border" Color="Transparent"/>

<SolidColorBrush x:Key="ComboBox.Static.Glyph" Color="#FF606060"/>

<Style x:Key="ComboBoxToggleButton" TargetType="{x:Type ToggleButton}">

<Setter Property="OverridesDefaultStyle" Value="true"/>

<Setter Property="IsTabStop" Value="false"/>

<Setter Property="Focusable" Value="false"/>

<Setter Property="ClickMode" Value="Press"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type ToggleButton}">

<Border x:Name="templateRoot" SnapsToDevicePixels="true" Background="{StaticResource ComboBox.Static.Background}" BorderThickness="{TemplateBinding BorderThickness}" BorderBrush="{StaticResource ComboBox.Static.Border}">

<Border x:Name="splitBorder" Width="{DynamicResource {x:Static SystemParameters.VerticalScrollBarWidthKey}}" SnapsToDevicePixels="true" Margin="0" HorizontalAlignment="Right" BorderThickness="1" BorderBrush="Transparent">

<Path x:Name="arrow" VerticalAlignment="Center" Margin="0" HorizontalAlignment="Center" Fill="{StaticResource ComboBox.Static.Glyph}" Data="F1 M 0,0 L 2.667,2.66665 L 5.3334,0 L 5.3334,-1.78168 L 2.6667,0.88501 L0,-1.78168 L0,0 Z"/>

</Border>

</Border>

<ControlTemplate.Triggers>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="true"/>

<Condition Binding="{Binding IsMouseOver, RelativeSource={RelativeSource Self}}" Value="false"/>

<Condition Binding="{Binding IsPressed, RelativeSource={RelativeSource Self}}" Value="false"/>

<Condition Binding="{Binding IsEnabled, RelativeSource={RelativeSource Self}}" Value="true"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.Static.Editable.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.Static.Editable.Border}"/>

<Setter Property="Background" TargetName="splitBorder" Value="{StaticResource ComboBox.Static.Editable.Button.Background}"/>

<Setter Property="BorderBrush" TargetName="splitBorder" Value="{StaticResource ComboBox.Static.Editable.Button.Border}"/>

</MultiDataTrigger>

<Trigger Property="IsMouseOver" Value="true">

<Setter Property="Fill" TargetName="arrow" Value="{StaticResource ComboBox.MouseOver.Glyph}"/>

</Trigger>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsMouseOver, RelativeSource={RelativeSource Self}}" Value="true"/>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="false"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.MouseOver.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.MouseOver.Border}"/>

</MultiDataTrigger>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsMouseOver, RelativeSource={RelativeSource Self}}" Value="true"/>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="true"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.MouseOver.Editable.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.MouseOver.Editable.Border}"/>

<Setter Property="Background" TargetName="splitBorder" Value="{StaticResource ComboBox.MouseOver.Editable.Button.Background}"/>

<Setter Property="BorderBrush" TargetName="splitBorder" Value="{StaticResource ComboBox.MouseOver.Editable.Button.Border}"/>

</MultiDataTrigger>

<Trigger Property="IsPressed" Value="true">

<Setter Property="Fill" TargetName="arrow" Value="{StaticResource ComboBox.Pressed.Glyph}"/>

</Trigger>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsPressed, RelativeSource={RelativeSource Self}}" Value="true"/>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="false"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.Pressed.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.Pressed.Border}"/>

</MultiDataTrigger>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsPressed, RelativeSource={RelativeSource Self}}" Value="true"/>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="true"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.Pressed.Editable.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.Pressed.Editable.Border}"/>

<Setter Property="Background" TargetName="splitBorder" Value="{StaticResource ComboBox.Pressed.Editable.Button.Background}"/>

<Setter Property="BorderBrush" TargetName="splitBorder" Value="{StaticResource ComboBox.Pressed.Editable.Button.Border}"/>

</MultiDataTrigger>

<Trigger Property="IsEnabled" Value="false">

<Setter Property="Fill" TargetName="arrow" Value="{StaticResource ComboBox.Disabled.Glyph}"/>

</Trigger>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsEnabled, RelativeSource={RelativeSource Self}}" Value="false"/>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="false"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.Disabled.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.Disabled.Border}"/>

</MultiDataTrigger>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsEnabled, RelativeSource={RelativeSource Self}}" Value="false"/>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="true"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.Disabled.Editable.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.Disabled.Editable.Border}"/>

<Setter Property="Background" TargetName="splitBorder" Value="{StaticResource ComboBox.Disabled.Editable.Button.Background}"/>

<Setter Property="BorderBrush" TargetName="splitBorder" Value="{StaticResource ComboBox.Disabled.Editable.Button.Border}"/>

</MultiDataTrigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<ControlTemplate x:Key="ComboBoxTemplate" TargetType="{x:Type ComboBox}">

<Grid x:Name="templateRoot" SnapsToDevicePixels="true">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition MinWidth="{DynamicResource {x:Static SystemParameters.VerticalScrollBarWidthKey}}" Width="0"/>

</Grid.ColumnDefinitions>

<Popup x:Name="PART\_Popup" AllowsTransparency="true" Grid.ColumnSpan="2" IsOpen="{Binding IsDropDownOpen, Mode=TwoWay, RelativeSource={RelativeSource TemplatedParent}}" Margin="1" PopupAnimation="{DynamicResource {x:Static SystemParameters.ComboBoxPopupAnimationKey}}" Placement="Bottom">

<Themes:SystemDropShadowChrome x:Name="shadow" Color="Transparent" MaxHeight="{TemplateBinding MaxDropDownHeight}" MinWidth="{Binding ActualWidth, ElementName=templateRoot}">

<Border x:Name="dropDownBorder" BorderBrush="{DynamicResource {x:Static SystemColors.WindowFrameBrushKey}}" BorderThickness="1" Background="{DynamicResource {x:Static SystemColors.WindowBrushKey}}">

<ScrollViewer x:Name="DropDownScrollViewer">

<Grid x:Name="grid" RenderOptions.ClearTypeHint="Enabled">

<Canvas x:Name="canvas" HorizontalAlignment="Left" Height="0" VerticalAlignment="Top" Width="0">

<Rectangle x:Name="opaqueRect" Fill="{Binding Background, ElementName=dropDownBorder}" Height="{Binding ActualHeight, ElementName=dropDownBorder}" Width="{Binding ActualWidth, ElementName=dropDownBorder}"/>

</Canvas>

<ItemsPresenter x:Name="ItemsPresenter" KeyboardNavigation.DirectionalNavigation="Contained" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}"/>

</Grid>

</ScrollViewer>

</Border>

</Themes:SystemDropShadowChrome>

</Popup>

<ToggleButton x:Name="toggleButton" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" Grid.ColumnSpan="2" IsChecked="{Binding IsDropDownOpen, Mode=TwoWay, RelativeSource={RelativeSource TemplatedParent}}" Style="{StaticResource ComboBoxToggleButton}"/>

<ContentPresenter x:Name="contentPresenter" ContentTemplate="{TemplateBinding SelectionBoxItemTemplate}" ContentTemplateSelector="{TemplateBinding ItemTemplateSelector}" Content="{TemplateBinding SelectionBoxItem}" ContentStringFormat="{TemplateBinding SelectionBoxItemStringFormat}" HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}" IsHitTestVisible="false" Margin="{TemplateBinding Padding}" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="{TemplateBinding VerticalContentAlignment}"/>

</Grid>

<ControlTemplate.Triggers>

<Trigger Property="HasDropShadow" SourceName="PART\_Popup" Value="true">

<Setter Property="Margin" TargetName="shadow" Value="0,0,5,5"/>

<Setter Property="Color" TargetName="shadow" Value="#71000000"/>

</Trigger>

<Trigger Property="HasItems" Value="false">

<Setter Property="Height" TargetName="dropDownBorder" Value="95"/>

</Trigger>

<MultiTrigger>

<MultiTrigger.Conditions>

<Condition Property="IsGrouping" Value="true"/>

<Condition Property="VirtualizingPanel.IsVirtualizingWhenGrouping" Value="false"/>

</MultiTrigger.Conditions>

<Setter Property="ScrollViewer.CanContentScroll" Value="false"/>

</MultiTrigger>

<Trigger Property="ScrollViewer.CanContentScroll" SourceName="DropDownScrollViewer" Value="false">

<Setter Property="Canvas.Top" TargetName="opaqueRect" Value="{Binding VerticalOffset, ElementName=DropDownScrollViewer}"/>

<Setter Property="Canvas.Left" TargetName="opaqueRect" Value="{Binding HorizontalOffset, ElementName=DropDownScrollViewer}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

<SolidColorBrush x:Key="TextBox.Static.Background" Color="#FFFFFFFF"/>

<Style x:Key="ComboBoxEditableTextBox" TargetType="{x:Type TextBox}">

<Setter Property="OverridesDefaultStyle" Value="true"/>

<Setter Property="AllowDrop" Value="true"/>

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

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

<Setter Property="FocusVisualStyle" Value="{x:Null}"/>

<Setter Property="ScrollViewer.PanningMode" Value="VerticalFirst"/>

<Setter Property="Stylus.IsFlicksEnabled" Value="False"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type TextBox}">

<ScrollViewer x:Name="PART\_ContentHost" Background="Transparent" Focusable="false" HorizontalScrollBarVisibility="Hidden" VerticalScrollBarVisibility="Hidden"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<ControlTemplate x:Key="ComboBoxEditableTemplate" TargetType="{x:Type ComboBox}">

<Grid x:Name="templateRoot" SnapsToDevicePixels="true">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition MinWidth="{DynamicResource {x:Static SystemParameters.VerticalScrollBarWidthKey}}" Width="0"/>

</Grid.ColumnDefinitions>

<Popup x:Name="PART\_Popup" AllowsTransparency="true" Grid.ColumnSpan="2" IsOpen="{Binding IsDropDownOpen, RelativeSource={RelativeSource TemplatedParent}}" PopupAnimation="{DynamicResource {x:Static SystemParameters.ComboBoxPopupAnimationKey}}" Placement="Bottom">

<Themes:SystemDropShadowChrome x:Name="shadow" Color="Transparent" MaxHeight="{TemplateBinding MaxDropDownHeight}" MinWidth="{Binding ActualWidth, ElementName=templateRoot}">

<Border x:Name="dropDownBorder" BorderBrush="{DynamicResource {x:Static SystemColors.WindowFrameBrushKey}}" BorderThickness="1" Background="{DynamicResource {x:Static SystemColors.WindowBrushKey}}">

<ScrollViewer x:Name="DropDownScrollViewer">

<Grid x:Name="grid" RenderOptions.ClearTypeHint="Enabled">

<Canvas x:Name="canvas" HorizontalAlignment="Left" Height="0" VerticalAlignment="Top" Width="0">

<Rectangle x:Name="opaqueRect" Fill="{Binding Background, ElementName=dropDownBorder}" Height="{Binding ActualHeight, ElementName=dropDownBorder}" Width="{Binding ActualWidth, ElementName=dropDownBorder}"/>

</Canvas>

<ItemsPresenter x:Name="ItemsPresenter" KeyboardNavigation.DirectionalNavigation="Contained" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}"/>

</Grid>

</ScrollViewer>

</Border>

</Themes:SystemDropShadowChrome>

</Popup>

<ToggleButton x:Name="toggleButton" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" Grid.ColumnSpan="2" IsChecked="{Binding IsDropDownOpen, Mode=TwoWay, RelativeSource={RelativeSource TemplatedParent}}" Style="{StaticResource ComboBoxToggleButton}"/>

<Border x:Name="border" Background="{StaticResource TextBox.Static.Background}" Margin="{TemplateBinding BorderThickness}">

<TextBox x:Name="PART\_EditableTextBox" HorizontalContentAlignment="{TemplateBinding HorizontalContentAlignment}" IsReadOnly="{Binding IsReadOnly, RelativeSource={RelativeSource TemplatedParent}}" Margin="{TemplateBinding Padding}" Style="{StaticResource ComboBoxEditableTextBox}" VerticalContentAlignment="{TemplateBinding VerticalContentAlignment}"/>

</Border>

</Grid>

<ControlTemplate.Triggers>

<Trigger Property="IsEnabled" Value="false">

<Setter Property="Opacity" TargetName="border" Value="0.56"/>

</Trigger>

<Trigger Property="IsKeyboardFocusWithin" Value="true">

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

</Trigger>

<Trigger Property="HasDropShadow" SourceName="PART\_Popup" Value="true">

<Setter Property="Margin" TargetName="shadow" Value="0,0,5,5"/>

<Setter Property="Color" TargetName="shadow" Value="#71000000"/>

</Trigger>

<Trigger Property="HasItems" Value="false">

<Setter Property="Height" TargetName="dropDownBorder" Value="95"/>

</Trigger>

<MultiTrigger>

<MultiTrigger.Conditions>

<Condition Property="IsGrouping" Value="true"/>

<Condition Property="VirtualizingPanel.IsVirtualizingWhenGrouping" Value="false"/>

</MultiTrigger.Conditions>

<Setter Property="ScrollViewer.CanContentScroll" Value="false"/>

</MultiTrigger>

<Trigger Property="ScrollViewer.CanContentScroll" SourceName="DropDownScrollViewer" Value="false">

<Setter Property="Canvas.Top" TargetName="opaqueRect" Value="{Binding VerticalOffset, ElementName=DropDownScrollViewer}"/>

<Setter Property="Canvas.Left" TargetName="opaqueRect" Value="{Binding HorizontalOffset, ElementName=DropDownScrollViewer}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

<Style TargetType="{x:Type ComboBox}">

<Setter Property="FocusVisualStyle" Value="{StaticResource FocusVisual}"/>

<Setter Property="Background" Value="{StaticResource ComboBox.Static.Background}"/>

<Setter Property="BorderBrush" Value="{StaticResource ComboBox.Static.Border}"/>

<Setter Property="Foreground" Value="{DynamicResource {x:Static SystemColors.WindowTextBrushKey}}"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="ScrollViewer.HorizontalScrollBarVisibility" Value="Auto"/>

<Setter Property="ScrollViewer.VerticalScrollBarVisibility" Value="Auto"/>

<Setter Property="Padding" Value="6,3,5,3"/>

<Setter Property="ScrollViewer.CanContentScroll" Value="true"/>

<Setter Property="ScrollViewer.PanningMode" Value="Both"/>

<Setter Property="Stylus.IsFlicksEnabled" Value="False"/>

<Setter Property="Template" Value="{StaticResource ComboBoxTemplate}"/>

<Style.Triggers>

<Trigger Property="IsEditable" Value="true">

<Setter Property="IsTabStop" Value="false"/>

<Setter Property="Padding" Value="2"/>

<Setter Property="Template" Value="{StaticResource ComboBoxEditableTemplate}"/>

</Trigger>

</Style.Triggers>

</Style>

<!--#endregion-->

<!--Изначально задумывался, как стиль для кнопок в элементах просмотра блюд и заказов, но стал таковым для всех кнопок-->

<Style TargetType="{x:Type Button}" x:Key="ViewsButtons">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type Button}">

<Border x:Name ="border"

CornerRadius="10"

BorderBrush="Brown"

BorderThickness="2"

Background="#a53010"

SnapsToDevicePixels="True">

<ContentPresenter x:Name="contentPresenter"

Focusable="False"

HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}"

Margin="{TemplateBinding Padding}"

RecognizesAccessKey="True"

SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}"

VerticalAlignment="{TemplateBinding VerticalContentAlignment}"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="BorderBrush" TargetName="border" Value="Green"/>

<Setter Property="Background" TargetName="border" Value="#eff3e1"/>

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

</Trigger>

<Trigger Property="IsEnabled" Value="False">

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

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

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

<Setter Property="Margin" Value="5"/>

<Setter Property="FontFamily" Value="Century Gothic"/>

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

<Setter Property="FontStyle" Value="Normal"/>

<Setter Property="FontSize" Value="11"/>

<Setter Property="Padding" Value="2"/>

</Style>

<Style x:Key="RegLabels" TargetType="Label">

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

</Style>

</Application.Resources>

</Application>

## 3.3 Текст программы

Логика страниц:

MainWindow.xaml.cs

namespace Kursach{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window{

public MainWindow(){

InitializeComponent();

Res.Classes.ObjectsVisibility.FrameVision.f = fMain;

Res.Classes.ObjectsVisibility.btnsFrameVision.f = fButtons;

FrameVision.f.Navigate(new Res.Pages.NoUser.Main());

btnsFrameVision.f.Navigate(new Res.Pages.Buttons.ButtonsNoUser());

Res.Pages.Guest.OrdersPageHolder.p = new Res.Pages.Guest.Orders();}

private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e){

Res.Classes.ObjectsVisibility.EntityVision.e.Dispose(); } } }

Buttons\_Guest.xaml.cs

namespace Kursach.Res.Pages.Buttons{

/// <summary>

/// Логика взаимодействия для Buttons\_Guest.xaml

/// </summary>

public partial class Buttons\_Guest : Page {

public static Label lNoty;

public static Button \_btnMain;

public static Button \_btnOrders;

public static Button \_btnDishes;

public static Button \_btnOld;

public Buttons\_Guest(){

InitializeComponent();

\_btnMain = btnMain;

\_btnDishes = btnDishes;

\_btnOrders = btnOrders;

\_btnOld = btnMain;

lNoty = lNotifier;

Classes.StaticClasses.ButtonsBehaviour.SetButtonsColorDefault(\_btnMain, \_btnOld); }

private void btnMain\_Click(object sender, RoutedEventArgs e){

Classes.ObjectsVisibility.FrameVision.f.Navigate(new Guest.Main());}

private void btnDishes\_Click(object sender, RoutedEventArgs e){

Classes.ObjectsVisibility.FrameVision.f.Navigate(new NoUser.DishesBrowser());}

private void btnOrders\_Click(object sender, RoutedEventArgs e){

Classes.ObjectsVisibility.FrameVision.f.Navigate(Guest.OrdersPageHolder.p);}}}

ButtonsNoUser.xaml.cs

namespace Kursach.Res.Pages.Buttons{

/// <summary>

/// Логика взаимодействия для ButtonsNoUser.xaml

/// </summary>

public partial class ButtonsNoUser : Page{

/// <summary>

/// Статические поля для взаимодействия с кнопками извне

/// </summary>

public static Button \_btnMain;

public static Button \_btnLogin;

public static Button \_btnDishes;

public static Button \_btnReg;

private static Button oldButton;

public ButtonsNoUser(){

InitializeComponent();

\_btnMain = btnMain;

\_btnDishes = btnDishesView;

\_btnLogin = btnLogin;

\_btnReg = btnReg;

oldButton = btnMain;

Classes.StaticClasses.ButtonsBehaviour.SetButtonsColorDefault();

Classes.StaticClasses.ButtonsBehaviour.SetButtonsColorDefault(\_btnMain, btnDishesView);}

private void btnDishesView\_Click(object sender, RoutedEventArgs e){

SelectPage(btnDishesView, oldButton, new NoUser.DishesBrowser());}

private void btnMain\_Click(object sender, RoutedEventArgs e){

SelectPage(btnMain, oldButton, new NoUser.Main());}

private void btnLogin\_Click(object sender, RoutedEventArgs e){

SelectPage(btnLogin, oldButton, new NoUser.Login());}

private void btnReg\_Click(object sender, RoutedEventArgs e){

SelectPage(btnReg, oldButton, new NoUser.Registration());}

/// <summary>

/// Задает страницу, соответствующую нажатой кнопки, изменяя цвет этой кнопки

/// </summary>

/// <param name="active">Активная кнопка</param>

/// <param name="newActive">Активная страница</param>

private static void SelectPage(Button active, Button old, Page newActive){

Classes.ObjectsVisibility.FrameVision.f.Navigate(newActive);

Classes.StaticClasses.ButtonsBehaviour.SetButtonsColorDefault(active, old);

oldButton = active;}}}

Main.xaml.cs (guest)

namespace Kursach.Res.Pages.Guest{

/// <summary>

/// Логика взаимодействия для Main.xaml

/// </summary>

public partial class Main : Page{

public Main(){

InitializeComponent();}}}

Orders.xaml.cs

namespace Kursach.Res.Pages.Guest{

/// <summary>

/// Логика взаимодействия для Orders.xaml

/// </summary>

public partial class Orders : Page{

public static WrapPanel wpOrdersVisible;

DataGrid dgOrders = new DataGrid();

public Orders(){

InitializeComponent();

LoadOrdersToTable();

LoadOrders();}

private void LoadOrdersToTable(){

try{

EntityVision.e.v\_orders\_view.Load();

dgOrders.ItemsSource = EntityVision.e.v\_orders\_view.Local.ToBindingList();

wpOrdersVisible = wpOrders;}

catch { }}

private void LoadOrders(){

for (int i = 0; i < dgOrders.Items.Count + 1; i++){

try{

OrderView ovTemp = new OrderView();

BitmapImage bm = new BitmapImage();

bm.BeginInit();

bm.UriSource = new Uri(System.IO.Directory.GetCurrentDirectory() + EntityVision.e.v\_orders\_view.Local[i].PicturePath);

bm.EndInit();

ovTemp.iImage.Source = bm;

ovTemp.lConditionIndicator.Content = "Не оплачен";

ovTemp.lName.Content = EntityVision.e.v\_orders\_view.Local[i].Name;

ovTemp.id = EntityVision.e.v\_orders\_view.Local[i].id;

wpOrders.Children.Add(ovTemp);}

catch { }}}

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

wpOrders.Children.Clear();

LoadOrdersToTable();

LoadOrders();}

private void btnPayAll\_Click(object sender, RoutedEventArgs e){}}}

OrdersPageHolder.cs

namespace Kursach.Res.Pages.Guest{

class OrdersPageHolder{

public static Page p = new Page();}}

DishesBrowser.xaml.cs

namespace Kursach.Res.Pages.NoUser{

/// <summary>

/// Логика взаимодействия для DishesBrowser.xaml

/// </summary>

public partial class DishesBrowser : Page{

DataGrid nota = new DataGrid(); //По приколу создаем "виртуальную" таблицу чтобы пихнуть туда блюда из базы

List<InterfaceObjects.DishView> lViews = new List<InterfaceObjects.DishView>(); //Делаем коллекцию с кастомным элементом управления

public DishesBrowser(){

InitializeComponent();

LoadDishesDataFromDB();

ThreadLoadedData();

LoadDishesToViews();

WriteDishesCount();}

private void WriteDishesCount(){

var l = new Label();

l.Content = $"Записей загружено: {lViews.Count}";

wpDishes.Children.Add(l);}

private void ThreadLoadedData(){

Thread thread1 = new Thread(LoadDishesToViews);

thread1.Start();

Thread.Sleep(0);}

private void LoadDishesDataFromDB(){

try{

Classes.ObjectsVisibility.EntityVision.e.v\_Dishes\_With\_Types\_View.Load();//Загружаем таблицу с блюдами из базы данных

nota.ItemsSource = Classes.ObjectsVisibility.EntityVision.e.v\_Dishes\_With\_Types\_View.Local.ToBindingList();

catch { }}

private void LoadDishesToViews(){

for (int i = 0; i < nota.Items.Count - 1; i++) //ну в цикле все +/- понятно, но стоит все равно объяснить{

try{

BitmapImage bm = new BitmapImage();

bm.BeginInit();

bm.UriSource = new Uri(System.IO.Directory.GetCurrentDirectory() +

Classes.ObjectsVisibility.EntityVision.e.v\_Dishes\_With\_Types\_View.Local[i].PicturePath);

bm.EndInit();

lViews.Add(new InterfaceObjects.DishView()); //Добавляем новый элемент в коллекцию

decimal t\_price = Math.Round(Classes.ObjectsVisibility.EntityVision.e.v\_Dishes\_With\_Types\_View.Local[i].Price, 2); //Округляем циферки в цене

lViews[i].id = Classes.ObjectsVisibility.EntityVision.e.v\_Dishes\_With\_Types\_View.Local[i].id; //Задаем значение поля id

lViews[i].lName.Content = Classes.ObjectsVisibility.EntityVision.e.v\_Dishes\_With\_Types\_View.Local[i].Name.ToString(); //Задаем имя...

lViews[i].lPrice.Content = $"{t\_price} p."; //...цену...

lViews[i].lWeight.Content = $"{Classes.ObjectsVisibility.EntityVision.e.v\_Dishes\_With\_Types\_View.Local[i].Weight} г."; //...вес...

lViews[i].lType.Content = Classes.ObjectsVisibility.EntityVision.e.v\_Dishes\_With\_Types\_View.Local[i].Type; //...тип...

lViews[i].iPic.Source = bm; //...картинку

wpDishes.Children.Add(lViews[i]);}

catch { }}}}}

Login.xaml.cs

namespace Kursach.Res.Pages.NoUser{

/// <summary>

/// Логика взаимодействия для Login.xaml

/// </summary>

public partial class Login : Page{

public Login(){

InitializeComponent();

tbLogin.Focus();}

private void btnLogin\_Click(object sender, RoutedEventArgs e){

LogMeIn();}

/// <summary>

/// Метод входа пользователя

/// </summary>

private void LogMeIn(){

try{

var usr = Classes.ObjectsVisibility.EntityVision.e.Users.FirstOrDefault(x => x.Telephone == tbLogin.Text && x.Password == pbPassword.Password); // Создает временный объект пользователя с введенными логином и паролем

if (usr == null){

MessageBox.Show("Такого пользователя нет"); // Если данные введены некорректно или не введены вовсе, выводится сообщение о том, что пользователя не существует}

else{

switch (usr.Type) // Если такой пользователь есть, то он проверяет его роль{

case (1): // Если это роль пользователя, то загружаются соответствующие страницы

Classes.ObjectsVisibility.btnsFrameVision.f.Navigate(new Buttons.Buttons\_Guest());

Classes.ObjectsVisibility.FrameVision.f.Navigate(new Guest.Main());

break;

default: // Если роли пользователя не найдено, то выводится сообщение о том, что ее не существует

MessageBox.Show("Такой роли пользователя не существует");

break;}

// Следующие команды записывают данные о пользователе в соответствующие поля в статическом классе для дальнейшего использования

Classes.ObjectsVisibility.CurrentUserInfo.id = usr.id;

Classes.ObjectsVisibility.CurrentUserInfo.type = usr.Type;}}

catch { }}}}

Main.xaml.cs (NoUser)

namespace Kursach.Res.Pages.NoUser{

/// <summary>

/// Логика взаимодействия для Main.xaml

/// </summary>

public partial class Main : Page{

public Main(){

InitializeComponent();}}}

Registration.xaml.cs

namespace Kursach.Res.Pages.NoUser{

/// <summary>

/// Логика взаимодействия для Registration.xaml

/// </summary>

public partial class Registration : Page{

private Dictionary<string, int[]> Dates = new Dictionary<string, int[]>();

private Dictionary<string, int[]> DatesLoap = new Dictionary<string, int[]>();

private List<string> Months = new List<string>{

"Январь","Февраль","Март","Апрель","Май","Июнь","Июль","Август","Сентябрь","Октябрь", "Ноябрь","Декабрь"};

public Registration(){

InitializeComponent();

FillMonths();

FillMonthsLoap();

FillcbY();

FillcbM();

FillcbD();}

private void FillMonths(){

Dates.Add("Январь", new int[] { 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

Dates.Add("Февраль", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28});

Dates.Add("Март", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

Dates.Add("Апрель", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30});

Dates.Add("Май", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

Dates.Add("Июнь", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30});

Dates.Add("Июль", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

Dates.Add("Август", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

Dates.Add("Сентябрь", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30});

Dates.Add("Октябрь", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

Dates.Add("Ноябрь", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30});

Dates.Add("Декабрь", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

private void FillMonthsLoap(){

DatesLoap.Add("Январь", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

DatesLoap.Add("Февраль", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29});

DatesLoap.Add("Март", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

DatesLoap.Add("Апрель", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30});

DatesLoap.Add("Май", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

DatesLoap.Add("Июнь", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30});

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

DatesLoap.Add("Август", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

DatesLoap.Add("Сентябрь", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30});

DatesLoap.Add("Октябрь", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

DatesLoap.Add("Ноябрь", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30});

DatesLoap.Add("Декабрь", new int[] {

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31});

private void FillcbY(){

for (int i = DateTime.Now.Year - 18; i > 1900; i--){

cbYear.Items.Add(i);}

cbYear.SelectedIndex = 0;}

private void FillcbM(){

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

cbMonth.Items.Add(Months[i]);}

cbMonth.SelectedIndex = 0;}

private void FillcbD(){

if ((int)cbYear.SelectedItem % 4 == 0){

cbDate.ItemsSource = DatesLoap[cbMonth.SelectedItem.ToString()];}

else{

cbDate.ItemsSource = Dates[cbMonth.SelectedItem.ToString()];}

cbDate.SelectedIndex = 0;}

private void cbMonth\_SelectionChanged(object sender, SelectionChangedEventArgs e){

if (cbMonth.SelectedItem != null){

FillcbD();}}

private void cbYear\_SelectionChanged(object sender, SelectionChangedEventArgs e){

if (cbMonth.SelectedItem != null){

FillcbD();}}

private void btnRegistration\_Click(object sender, RoutedEventArgs e){

Registrate();}

private void btnLogin\_Click(object sender, RoutedEventArgs e){

FrameVision.f.Navigate(new Login());}

private void Registrate(){

if(pbPassword.Password != pbPasswordConfirm.Password){

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

else if(EntityVision.e.Users.Count(x => x.Telephone == mtbTelephone.Text) > 0){

MessageBox.Show("Такой телефон уже зарегестрирован.");}

if (mtbEmail.Text != mtbEmailConfirm.Text){

MessageBox.Show("Почты не совпадают.");}

else{

try{

Users usr = new Users(){

Type = 1,

Telephone = mtbTelephone.Text,

BirthDate = PickBirthDate(),

RegDate = DateTime.Now,

Email = mtbEmail.Text,

Name = mtbName.Text,

Sorename = mtbSurename.Text,

Thirdname = mtbThirdname.Text,

Mailing = IsSubscribed.IsEnabled,

Password = pbPassword.Password};

EntityVision.e.Users.Add(usr);

EntityVision.e.SaveChanges();

MessageBox.Show("Пользователь создан");

FrameVision.f.Navigate(new Login());}

catch (Exception a){

MessageBox.Show(a.ToString());}}}

private DateTime PickBirthDate(){

DateTime date = new DateTime(Convert.ToInt32(cbYear.SelectedItem), cbMonth.SelectedIndex + 1, Convert.ToInt32(cbDate.SelectedItem));

return date;}}}

Статические классы:

btnsFrameVision.cs

namespace Kursach.Res.Classes.ObjectsVisibility{

class btnsFrameVision{

public static Frame f = new Frame();}}

CurrentUserInfo.cs

namespace Kursach.Res.Classes.ObjectsVisibility{

class CurrentUserInfo{

public static int id;

public static int type;}}

EntityVision.cs

namespace Kursach.Res.Classes.ObjectsVisibility{

class EntityVision{

public static Restraunt\_DB\_Entities e = new Restraunt\_DB\_Entities();}}

FrameVision.cs

namespace Kursach.Res.Classes.ObjectsVisibility{

class FrameVision{

public static Frame f = new Frame();}}

ButtonsBehaviour.cs

namespace Kursach.Res.Classes.StaticClasses{

public static class ButtonsBehaviour{

public static void SetButtonsColorDefault(Button activeButton, Button oldButton){

oldButton.Background = new SolidColorBrush(Color.FromRgb(249, 207, 195));

activeButton.Background = new SolidColorBrush(Color.FromRgb(255, 255, 255));}

public static void SetButtonsColorDefault(){

switch (Classes.ObjectsVisibility.CurrentUserInfo.type){

case 1:

Buttons\_Guest.\_btnMain.Background = new SolidColorBrush(Color.FromRgb(249, 207, 195));

Buttons\_Guest.\_btnMain.Background = new SolidColorBrush(Color.FromRgb(249, 207, 195));

break;

default:

ButtonsNoUser.\_btnMain.Background = new SolidColorBrush(Color.FromRgb(249, 207, 195));

ButtonsNoUser.\_btnLogin.Background = new SolidColorBrush(Color.FromRgb(249, 207, 195));

ButtonsNoUser.\_btnDishes.Background = new SolidColorBrush(Color.FromRgb(249, 207, 195));

ButtonsNoUser.\_btnReg.Background = new SolidColorBrush(Color.FromRgb(249, 207, 195));

break;}}}}

Логика элементов управления:

DishView.xaml.cs

namespace Kursach.Res.InterfaceObjects{

/// <summary>

/// Логика взаимодействия для DishView.xaml

/// </summary>

public partial class DishView : UserControl{

public int id;

public DishView(){

InitializeComponent();

this.Margin = new Thickness(10);}

private void btnOrder\_Click(object sender, RoutedEventArgs e){

if (Classes.ObjectsVisibility.CurrentUserInfo.type == 1){

var order = new Orders{

dish\_id = id,

user\_id = Classes.ObjectsVisibility.CurrentUserInfo.id,

Time = DateTime.Now,

IsOrdered = false,

IsCooking = false,

IsCooked = false,

IsDelivered = false};

Classes.ObjectsVisibility.EntityVision.e.Orders.Add(order);

Classes.ObjectsVisibility.EntityVision.e.SaveChanges();

Notificate("Заказ добавлен");}

else{

MessageBox.Show("Только авторизованные пользователи могут заказывать блюда.", "Ошибка");}}

private void Notificate(string Message){

lNoty.Content = Message;}}}

OrderView.xaml.cs

namespace Kursach.Res.InterfaceObjects{

/// <summary>

/// Логика взаимодействия для OrderView.xaml

/// </summary>

public partial class OrderView : UserControl{

public int id;

public OrderView(){

InitializeComponent();}

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

EntityVision.e.Orders.Remove(EntityVision.e.Orders.Find(id));

EntityVision.e.SaveChanges();

Pages.Guest.Orders.wpOrdersVisible.Children.Remove(this);}

public void btnPay\_Click(object sender, RoutedEventArgs e){

lConditionIndicator.Content = "Оплачено, готовится";

btnCancel.IsEnabled = false;

btnPay.IsEnabled = false;}}}

# 4 Руководство пользователя

## 4.1 Выполнение программного модуля

Для начала, пользователь должен запустить саму программу.

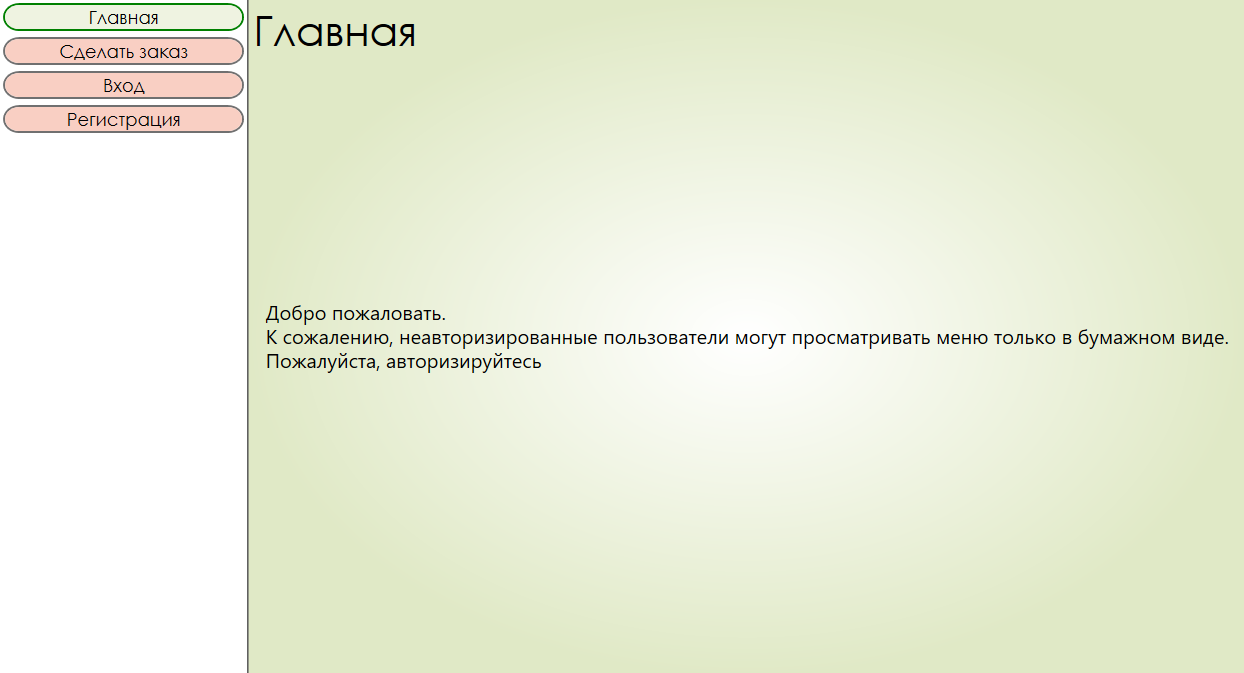


Рис. 24 «Запуск программы»

Затем, перед ним появится окно приложения, в котором он увидит меню навигации по приложению слева, а справа основное окно, в котором, в зависимости от выбранного пункта меню, будет отображаться та или иная страница (по умолчанию – главная).

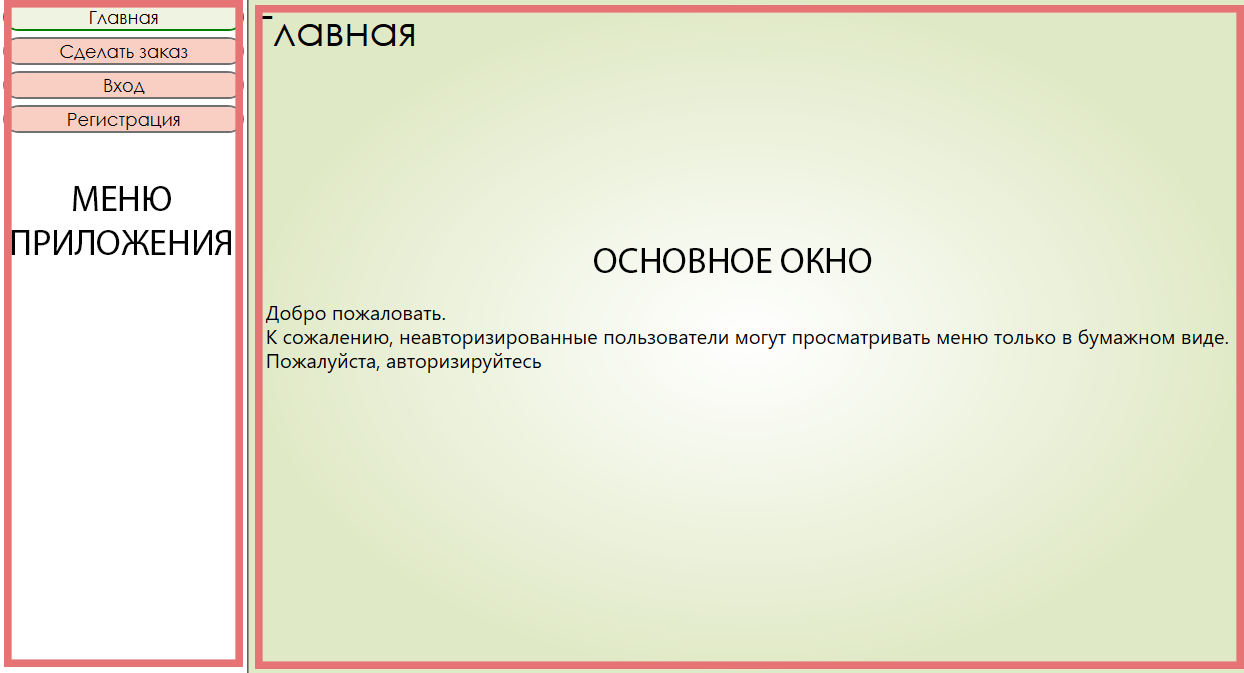


Рис. 25 «Расположение элементов»

Если пользователь нажимает на кнопку «Сделать заказ», то открывается страница с блюдами, доступными к заказу.



Рис. 26 «Просмотр блюд»

Если неавторизованный пользователь попытается сделать заказ, то у него это не получится, так как только авторизованные пользователи могут заказать блюдо в программе.

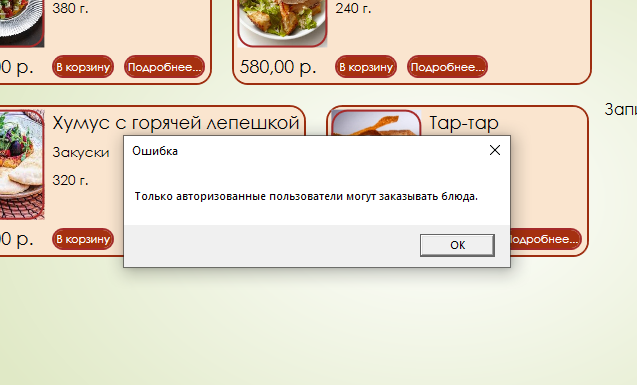


Рис. 27 «Ошибка – пользователь пытался сделать заказ будучи неавторизованным»

На странице входа пользователю нужно будет ввести телефон и пароль, если у него есть аккаунт.

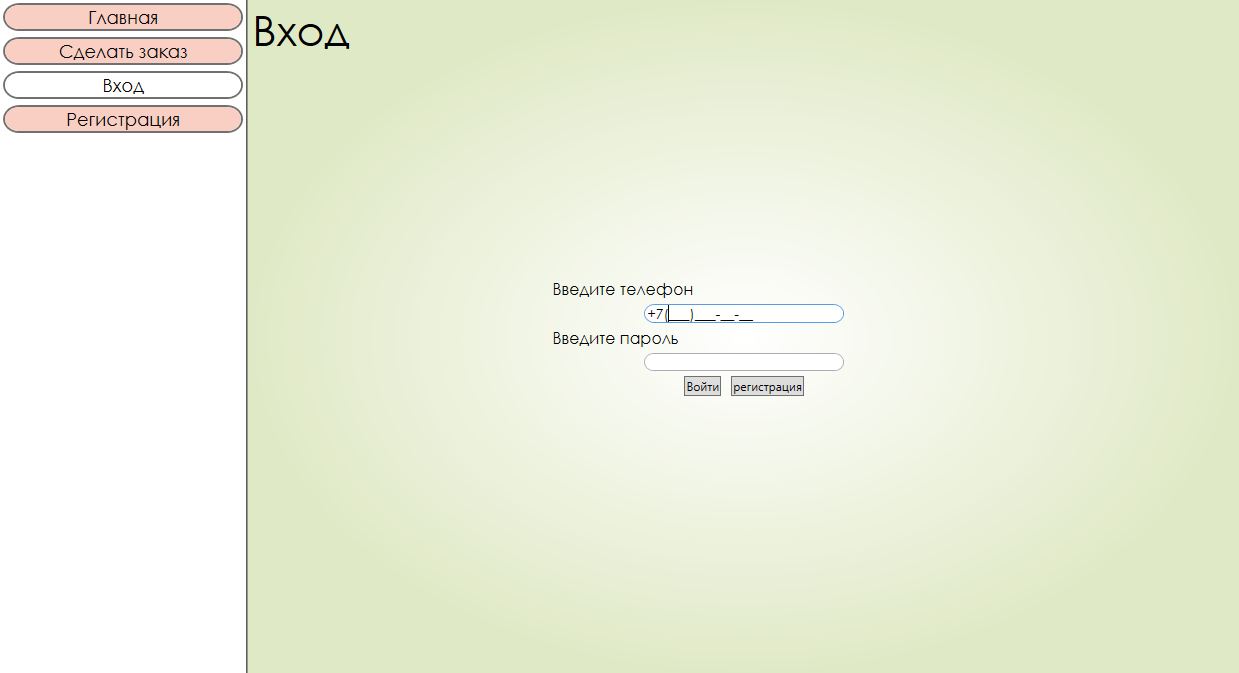


Рис. 28 «Страница входа»

На странице регистрации пользователю будет необходимо ввести его личные данные, а также выбрать, будет ли он получать рассылку по почте.

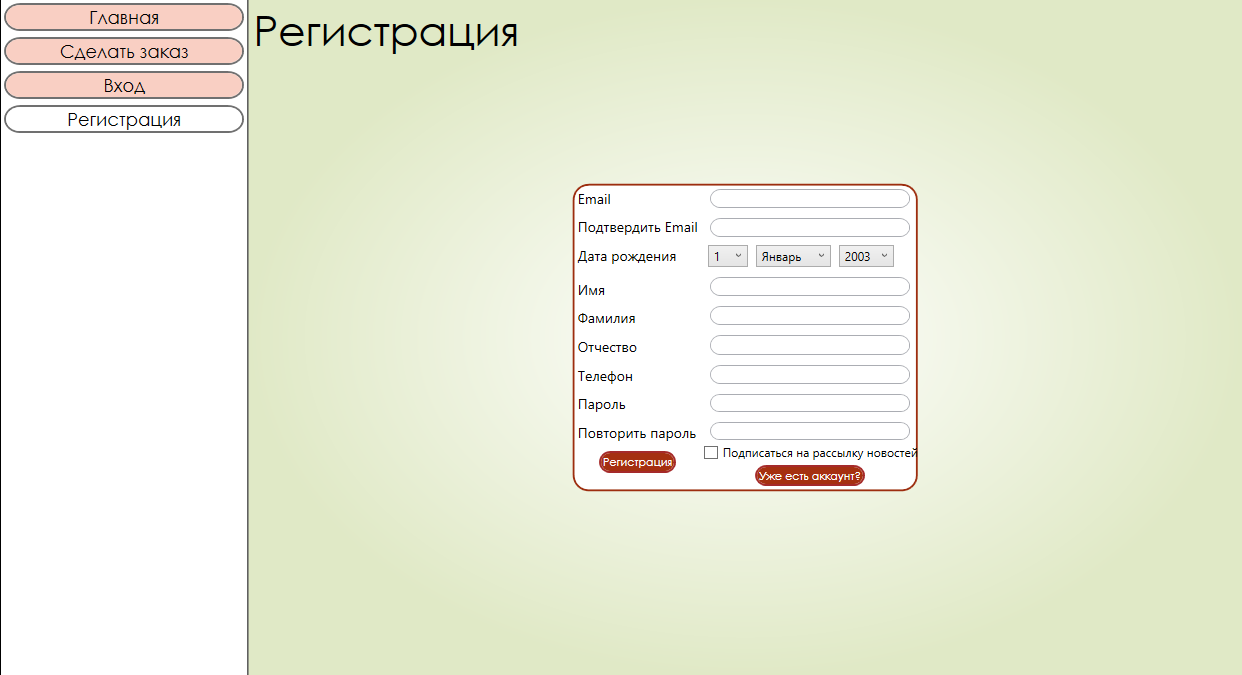


Рис. 29 «Страница регистрации»

Войдя, пользователь увидит главную страницу.

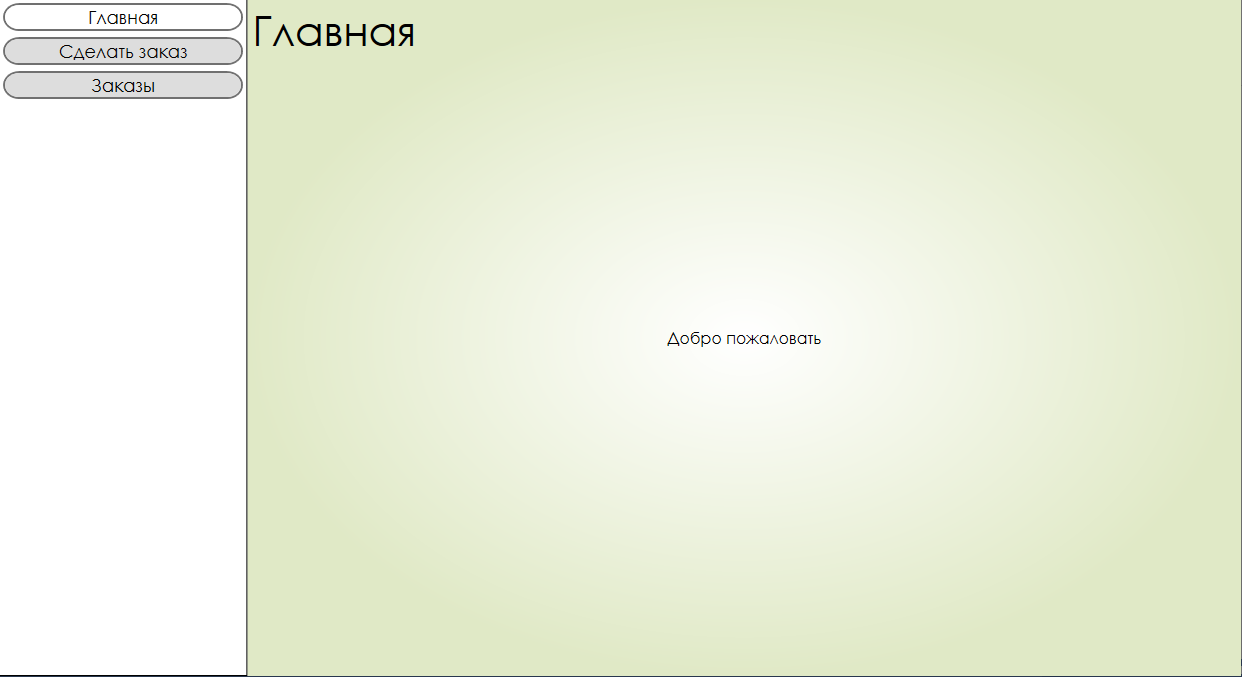


Рис. 30 «Главная страница авторизованного пользователя»

На странице выбора блюд пользователь может заказать себе блюдо на выбор.



Рис. 31 «Заказ блюда прошел успешно»

Зайдя на страницу заказов, пользователь увидит свои заказы.



Рис. 32 «Список заказов»

Заказ можно отменить.



Рис. 33 «Заказ отменен

Также, заказ можно оплатить и отправить в готовку.

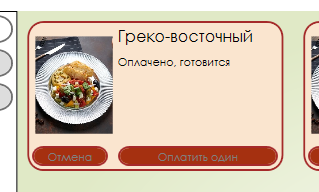


Рис. 34 «Заказ оплачен и началась его готовка»

## 4.2 Сообщения пользователю

Попытка заказать блюдо, не войдя в аккаунт:

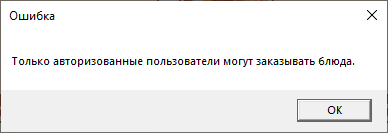


Рис. 35 «Ошибка 1»

Попытка войти, введя неверные пользовательские данные:

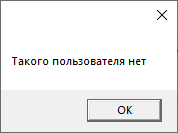
****

Рис. 36 «Ошибка 2»

# 5. Тестирование и отладка программного модуля

В ходе тестирования были выявлены следующие ошибки:

Таблица 1 «Баги приложения»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Заголовок** | **Шаги воспроизведения** | **Ожидаемый результат** | **Фактический результат** |
| 1 | Отсутствие информации о блюде | Меню – Сделать заказ/Любое блюдо/ Подробнее | Появится окно, содержащее информацию о блюде (состав, рецепт) | Отсутствие результата |
| 2 | Появление несуществующих записей | (Авт.) Меню – Заказы/ Отменить заказ/ Обновить | После отмены заказ уберется из списка заказов и не появляются после обновления | После обновления отмененный заказ снова появляется |
| 3 | Обнуление заказа после оплаты | (Авт.) Меню – Заказы/  Оплатить один/ Обновить | После оплаты заказа все кнопки виджета блюда блокируется, а сам заказ переходит в состояние оплаченного. После обновления все должно остаться как есть | После обновления блюдо снова становится неоплаченным |
| 4 | Баг в меню навигации | (Авт.) Меню | Неактивные кнопки должны быть окрашены в ярко-коричневый цвет, а активные в белый | Неактивные кнопки серого цвета, плюс кнопка «Главная» белого |
| 5 | Полная оплата заказа | (Авт.) Меню – заказы/ Оплатить все | Все заказы будут оплачены как описано в п.3 | Отсутствие результата |
| 6 | Регистрация пустого пользователя | Меню – Регистрация/  Оставить поля пустыми/  Регистрация | Должно появиться окно, предупреждающее, что нельзя оставлять поле пустым | Регистрируется пустой пользователь |

Авт. – авторизированный пользователь

# Заключение

В ходе написания курсовой работы был разработан программный модуль для автоматизации организации общественного питания. Были изучены такие технологии как Visual C#, Entity Framework, SQL Server, XCeed WPF ToolKit, Visual Studio, Microsoft SQL Server Management Studio на базе SQLExpress. Во время разработки программы были изучены методы проектирования программных модулей и баз данных, а также непосредственное их создание.

Созданный программный модуль соответствует поставленным задачам. Были реализованы следующие функции:

* Просмотр информации о блюде
* Заказ блюда, если пользователь авторизован
* Авторизация и регистрация пользователей в системе
* Отмена и оплата блюда по желанию пользователя.

Были созданы следующие страницы

* Главная – отображает приветствие для пользователя
* Сделать заказ – страница отображает список блюд, доступных для заказа. Заказ можно сделать только если пользователь авторизован
* Вход – страница входа в аккаунт. Содержит поля ввода телефона, пароля, а также кнопки «Вход» и «Регистрация»
* Регистрация – позволяет зарегистрироваться пользователю, если у него нет аккаунта в системе.
* Заказы – отображает заказы, сделанные пользователем и позволяет оплатить или отменить сделанный заказ

Конечный продукт соответствует функциональной схеме и блок-схеме приложения.

# Список литературы

1. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем: учебник для студентов СПО /Г.Н. Федорова. 2-е изд., стер. - М.: Издательский центр «Академия»,2018. - 384с.
2. Федорова Г.Н. Осуществление интеграции программных модулей: учебник для студентов СПО / Г.Н. Федорова. 2-е изд., стер. - М.: Издательский центр «Академия»,2018. - 285 с.
3. Царёв, Р.Ю. Алгоритмы и структуры данных (CDIO): учебник /Р.Ю. Царёв, А.В. Прокопенко; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск: СФУ, 2016. - 204 с. : ил. - Библиогр. в кн. - ISBN 978-5-7638-3388-1; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=497016>
4. Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство Юрайт, 2020. — 235 с.
5. C# 7.0. Справочник. Полное описание языка.: Пер. с англ. – СпБ.: ООО “Альфакнига”, 2018. – 1024 с. : ил. – Парал. тит. англ.
6. Microsoft Visual C#. Подробное руководство. 8-е изд. – СПб.: Питер, 2017. – 848 с.: ил. – (Серия «Библиотека программиста»).
7. Программирование на C# для начинающих. Основные сведения / Алексей Васильев. – Москва : Эксмо, 2018. – 592 с. – (Российский компьютерный бестселлер).
8. Язык программирования C# 7 и платформы .NET и .NET Core, 8-е изд. : Пер. с англ. – СПб. : ООО “Диалектика”, 2018 – 1328 с. : ил. – Парал. тит. англ.
9. Оппель, Эндрю Дж. SQL. Полное руководство / Оппель Эндрю Дж.. - М.: Диалектика / Вильямс, 2016. - 902 c.

# Приложения

## Приложение 1

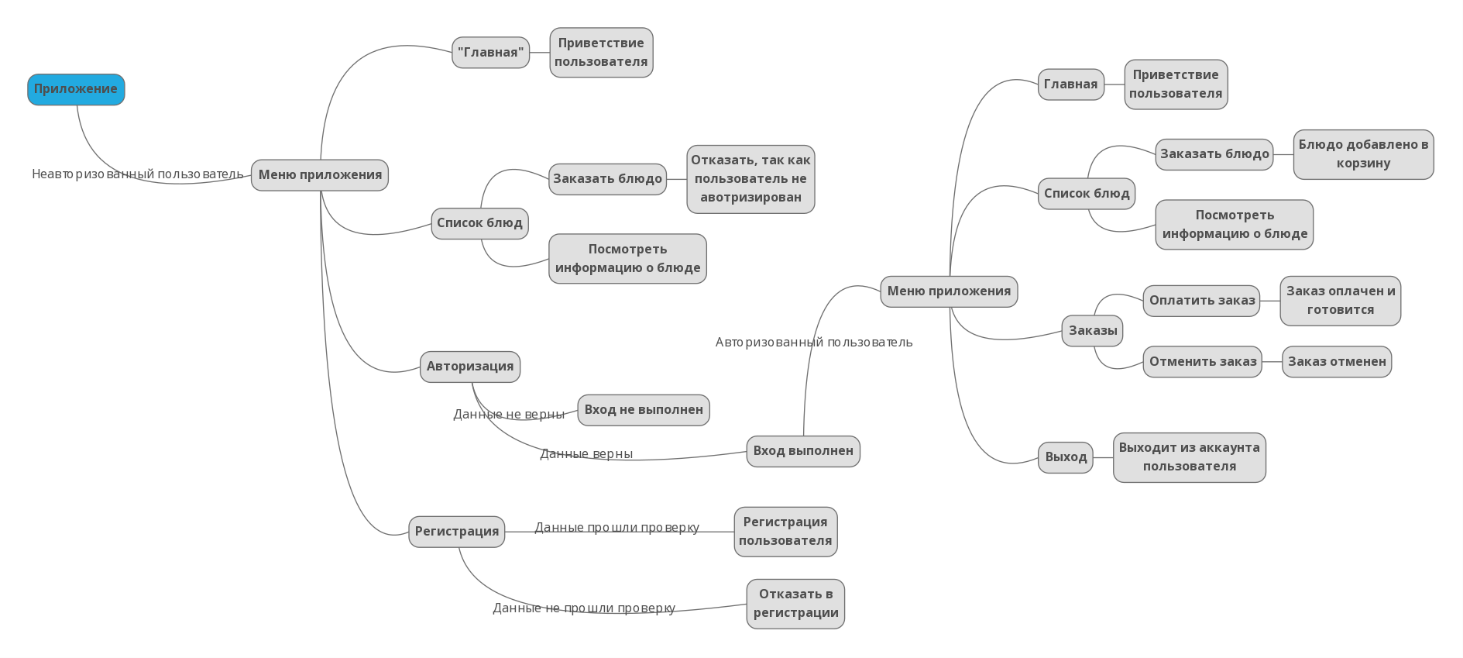


Рис. 37 «Функциональная схема приложения»