федеральное государственное бюджетное образовательное учреждение высшего образования

«**Вологодский государственный университет**»

университетский колледж

Информационные системы и программирование

(наименование специальности)

**КУРСОВОЙ ПРОЕКТ**

Дисциплина: «Разработка программных модулей»

Наименование темы: «Разработка информационной системы интернет-магазина»

Код работы КП 09.02.07.13.00.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Код специальности, код УК, регистрационный номер по журналу, год

Руководитель \_\_\_\_\_Дегтярёв Михаил Евгеньевич\_\_\_\_\_\_\_\_\_\_

(уч. степень, звание, должность. Ф.И.О)

Выполнил (а) обучающийся \_\_\_\_\_Юматов Вячеслав Дмитриевич\_\_\_\_\_\_\_\_\_

(Ф.И.О)

Группа, курс \_\_\_\_\_ИСП 32-20, 3 курс\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата сдачи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка по защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись преподавателя)

Вологда

2023 г.

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

СОДЕРЖАНИЕ

Введение 4

1. Аналитический обзор 6

* 1. Описание требований 6
  2. Описание программного обеспечения 6
  3. База данных 8

2. Разработка программного обеспечения 11

1. Разработка пользовательского интерфейса 11
2. Разработка программного функционала 18

3. Тестирование 36

4. Руководство пользователя 40

# Заключение 47

# Список использованных источников 48

# Приложение 1 49

# Приложение 2 53

ВВЕДЕНИЕ

В современном мире интернет-магазины стали очень популярными и востребованными. Они позволяют покупателям с лёгкостью и удобством выбирать и заказывать товары, не покидая дом. Однако, для эффективной работы и привлечения покупателей, необходима слаженная и корректная работа информационной системы. Кроме этого, информационная система интернет-магазина должна обладать интуитивно понятным дизайном, который позволяет удобно выбирать товары, становится одним из факторов, которые обеспечивают долгосрочный успех интернет-магазина. Автоматизация работы всех функций информационной системы также является важным компонентом, который помогает экономить время, ресурсы и обеспечивает быстрый доступ к нужной информации. К тому же, безотказная работа системы и готовность к быстрому реагированию на любые изменения в работе магазина являются еще одними из важных факторов. Это позволяет предотвращать возникновение ошибок, связанных с техническими проблемами, также быстро решать все проблемы и возникающие недоразумения. Благодаря шифрованию данных, пользователи информационных систем интернет-магазина могут не беспокоиться за недоступность аккаунта по причине чужого внедрения в личные данные и получения несанкционированного доступа к аккаунту. Постоянное совершенствование работы информационной системы позволит в будущем расширить ассортимент товаров, привлечь новых покупателей и улучшить удобство использования интернет-магазина. Итак, информационная система является важной составляющей любого успешного интернет-магазина. Ее эффективная работа обеспечивает простоту, удобство и безопасность для покупателей, а также ускоряет и упрощает процессы для владельцев магазина. Правильное использование современных технологий и постоянное совершенствование позволяют интернет-магазину добиваться успеха и развиваться в будущем. Описанная в проекте игра является достаточно популярной среди пользователей Интернета. Интернет-магазины и их информационные системы крайне часто разрабатываются на нынешний день, что делает общую тему реализации информационной системы интернет-магазина актуальной.

Целью курсового проекта является «Разработка информационной системы интернет-магазина».

Задачами курсового проекта являются:

1. Выбор среды разработки информационной системы интернет-магазина.
2. Установка средств для разработки информационной системы интернет-магазина.
3. Реализация самого процесса заполнения информационной системы интернет-магазина необходимой информацией по тематике курсового проекта.
4. Тестирование созданной информационной системы интернет-магазина.

1 АНАЛИТИЧЕСКИЙ ОБЗОР

1.1 Описание требований

Информационная система интернет-магазина должна выполнять данные функции: корректный переход между окнами приложения (оконная навигация), авторизация пользователя, регистрация пользователя, изменение логина в окне профиля, добавление картинки в окне профиля, изменение добавленной картинки в окне профиля, корректный выход из профиля, корректное отображение роли пользователя в профиле, ролевая система типа “пользователь-администратор”, вывод имеющихся в базе данных продуктов в окно продуктов, корректное перемещение пользователем или администратором продуктов из окна продуктов в корзину, оформление продуктового чека, формирующегося из корзины продуктов, удаление продукта администратором, добавление продукта администратором, редактирование продукта администратором. Данные функции представлены в соответствии с приложением 1.

1.2 Описание программного обеспечения

Для разработки информационной системы интернет-магазинов используется специализированное программное обеспечение компьютера, с помощью которого есть возможность достигнуть полноценного рационального результата. Таких программ в сети Интернет достаточно немало, но для данного проекта были выбраны лишь несколько программ. В проекте были использованы: SQL Server Management Studio (SSMS) для осуществления запросов к базе данных и хранения данных в базе данных, Microsoft Visual Studio (версии 17.4.4), являющаяся интегрированной средой разработки от компании Microsoft. Также, был выбран .Net Framework (версии 4.7.2), являющийся программным обеспечением, имеющим набор инструментов и библиотек для создания, тестирования и развертывания программного обеспечения. Графическим интерфейсом в данном курсовом проекте послужил WPF (Windows Presentation Foundation). Язык программирования, на котором был написан код программы, является C# — объектно-ориентированный язык программирования, разработанный компанией Microsoft. Также, в данном курсовом проекте был выбран Entity Framework — это набор технологий в ADO.NET, которые поддерживают разработку программных приложений, ориентированных на данные.

SQL Server Management Studio (SSMS) — это интегрированная среда разработки, которая позволяет управлять объектами базы данных в SQL Server. Она предоставляет возможность создавать, изменять и удалять объекты баз данных, такие как таблицы, процедуры хранимые, функции, триггеры, индексы и многое другое. Вы можете использовать SSMS для выполнения запросов к базам данных, настройки наших конфигураций и мониторинга производительности.

Microsoft Visual Studio — это интегрированная среда разработки (IDE) от компании Microsoft, которая предназначена для создания приложений для Windows, Android, iOS, веб-сайтов и облака. Она включает в себя ряд инструментов и технологий, которые позволяют разработчикам создавать приложения быстрее и с большей эффективностью.

.NET Framework — это программная платформа для разработки и запуска приложений на компьютерах под управлением операционных систем Windows. Она была создана компанией Microsoft и включает в себя несколько технологий и библиотек, позволяющих программистам быстро и эффективно разрабатывать приложения.

Windows Presentation Foundation (WPF) – это одна из самых популярных технологий Microsoft для разработки приложений со сложным и красивым пользовательским интерфейсом. Она была анонсирована в 2006 году и стала частью .NET Framework версии 3.0.

C# — это язык программирования, который разработала корпорация Microsoft в 2000 году. Он рассматривается как один из основных языков для создания приложений для операционных систем Windows и .NET Framework. Он основан на синтаксисе языка C, но имеет существенные отличия, сделанные разработчиками для обеспечения безопасности и более простого использования.

Entity Framework (EF) является объектно-ориентированным инструментом доступа к данным, который разработан Microsoft для работы с данными в приложениях .NET. Он предоставляет удобный способ работать с данными, скрывая детали работы с базой данных от разработчика и позволяя работать с данными в виде объектов, что упрощает процесс разработки. Entity Framework позволяет создавать модели данных на основе схемы базы данных и использовать LINQ для выполнения запросов к базе данных. Entity Framework автоматически генерирует SQL-запросы, используемые для получения данных из базы данных на основе LINQ-запросов. Он позволяет разработчику работать с данными в виде объектов и не думать о подробностях работы с базой данных.

1.3 База данных

При реализации данного курсового проекта была разработана база данных, необходимая для реализации основной программной части информационной системы интернет-магазина. В ней хранятся все требуемые данные в столбцах таблиц.

ER-диаграмма таблиц и связей базы данных представлена в соответствии с рисунком 1.

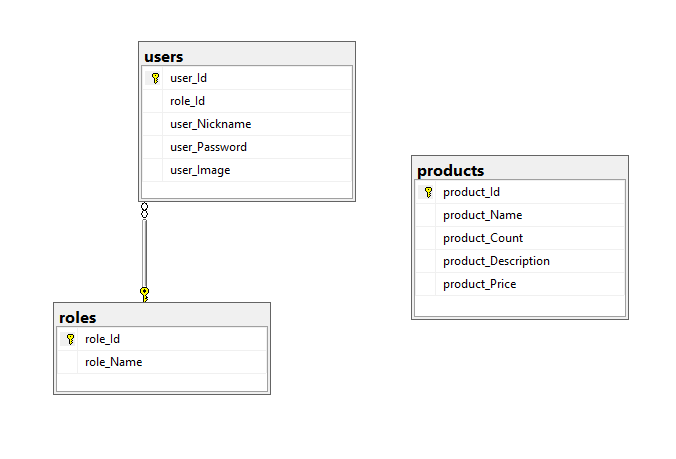


Рисунок 1. ER-диаграмма таблиц и связей базы данных

Таблица “users” содержит в себе данные столбцы: “user\_Id”, “role\_Id”, “user\_Nickname”, “user\_Password”, “user\_Image”. Данная таблица отвечает за основной функционал информационной системы – хранение данных о пользователях. Она содержит все требующиеся данные о пользователях информационной системы: идентификатор пользователя, идентификатор роли пользователя, логин пользователя, пароль пользователя, картинка на аватар в профильном окне для пользователя. Таблица “users” представлена в соответствии с рисунком 2.

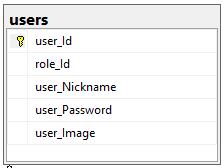


Рисунок 2. Внешний вид таблицы “users”

Таблица “roles” содержит в себе данные столбцы: “role\_Id”, “role\_Name”. Данная таблица является дополнительной таблицей к таблице “users”, она выполняет функцию уточнения ролей для пользователей и администраторов информационной системы интернет-магазина. Таблица “roles” представлена в соответствии с рисунком 3.

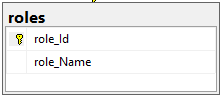


Рисунок 3. Внешний вид таблицы “roles”

Таблица “products” содержит в себе данные столбцы: “product\_Id”, “product\_Name”, “product\_Count”, “product\_Description”, “product\_Price”. Данная таблица отвечает за главный функционал информационной системы – продукты и корзина продуктов, которую набирает пользовать в процессе покупки продуктов. Она хранит все данные о продуктах: идентификатор продукта, имя продукта, количество продукта, описание продукта и цена продукта. Таблица “products” представлена в соответствии с рисунком 4.

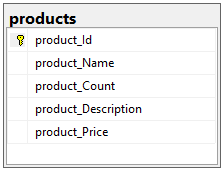


Рисунок 4. Внешний вид таблицы “products”

2 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Разработка пользовательского интерфейса

В данном курсовом проекте был разработан XAML код, являющийся языком разметки в данной информационной системе. XAML код в WPF осуществляет описание графического интерфейса, связь между элементами и логику их поведения. Окна, разработанные на XAML коде: “App.xaml”, “MainPage.xaml”, “Products.xaml”, “Profile.xaml”, “Authorization.xaml”, “Registration.xaml”, “ShoppingCart.xaml”, “changeNick.xaml”, “addingProduct.xaml”, “redactingProduct.xaml”, “removingProduct.xaml”.

Окно “App.xaml” хранит в себе основные стили элементов остальных XAML окон. XAML код окна “MainPage.xaml” представлен ниже:

*<?xml version="1.0" encoding="utf-8"?><Window x:Class="internetShopProject.MainPage"xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"Title="Main Page" Height="450" Width="800" ResizeMode="CanMinimize"><Grid><Grid.RowDefinitions><RowDefinition Height="60"/><RowDefinition Height="\*"/><RowDefinition Height="50"/></Grid.RowDefinitions><StackPanel Orientation="Horizontal" Grid.Row="0" Background="#5d1d99"><TextBlock FontFamily="Montserrat" FontSize="25" Foreground="White" Margin="20, 13, 0, 10" Text="Main Page"/><Button Style="{StaticResource mainCreateAccBtn}" Content="Registration" Click="createAccClicked"/><Button Style="{StaticResource mainCreateAccBtn}" Content="Authorization" Click="authorizeIntoAccClicked"/></StackPanel><StackPanel Orientation="Horizontal" Grid.Row="1" Background="#461673"><Label FontFamily="Montserrat" Grid.Row="1" FontSize="25" Foreground="White" Margin="15, 15, 0, 0" VerticalAlignment="Top" Height="50" Content="Welcome to the internet shop system!"/></StackPanel><StackPanel Grid.Row="2" Background="#5d1d99"><Button Name="cartBtnOnMain" Style="{StaticResource mainCartBtn}" Content="Shopping Cart" Click="cartBtnClicked"/><Button Name="prdBtnOnMain" Style="{StaticResource mainProductsBtn}" Content="Products" Click="productsBtnClicked"/></StackPanel></Grid></Window>*

XAML код окна “Products.xaml” представлен ниже:

*<Window x:Class="internetShopProject.Products" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Title="Products" Height="450" Width="800" ResizeMode="CanMinimize"> <Grid> <Grid.RowDefinitions> <RowDefinition Height="60"/> <RowDefinition Height="\*"/> <RowDefinition Height="50"/> </Grid.RowDefinitions> <StackPanel Orientation="Horizontal" Grid.Row="0" Background="#5d1d99"> <TextBlock Style="{StaticResource productTextBlock}" Text="Products"/> <Button Name="addProductButton" Style="{StaticResource productButton}" Content="Add Product" Click="addPrdClicked"/> <Button Name="removeProductButton" Style="{StaticResource productButton}" Content="Remove Product" Click="removePrdClicked"/> <Button Name="changeProductButton" Style="{StaticResource productButton}" Content="Change Product" Click="changePrdClicked"/> </StackPanel> <StackPanel Orientation="Horizontal" Grid.Row="1" Background="#461673"> <ListBox Name="listViewForProducts" ItemsSource="{Binding productsCollection}" Width="800" Background="Transparent" BorderThickness="0" PreviewMouseDoubleClick="productDoubleClicked"> <ListBox.ItemTemplate> <DataTemplate> <StackPanel> <TextBlock Name="productNameTextBlock" Foreground="White" FontSize="16"> <Run Text="Product: "/> <Run Text="{Binding product\_Name}"/> </TextBlock> <TextBlock Name="productDescTextBlock" Foreground="White" FontSize="16"> <Run Text="Description: "/> <Run Text="{Binding product\_Description}"/> </TextBlock> <TextBlock Name="productCountTextBlock" Foreground="White" FontSize="16"> <Run Text="Count: "/> <Run Text="{Binding product\_Count}"/> </TextBlock> <TextBlock Name="productPriceTextBlock" Foreground="White" FontSize="16"> <Run Text="Price: "/> <Run Text="{Binding product\_Price}"/> </TextBlock> </StackPanel> </DataTemplate> </ListBox.ItemTemplate> </ListBox> </StackPanel> <StackPanel Grid.Row="2" Background="#5d1d99"> <Button Style="{StaticResource refreshProductButton}" HorizontalAlignment="Right" Content="Refresh" Click="refreshProductClicked"/> </StackPanel> </Grid></Window>*

XAML код окна “Profile.xaml” представлен ниже:

*<?xml version="1.0" encoding="utf-8"?><Window x:Class="internetShopProject.Profile" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Title="Profile" Height="300" Width="450" ResizeMode="CanMinimize"> <Grid> <Grid.RowDefinitions> <RowDefinition Height="60"/> <RowDefinition Height="\*"/> <RowDefinition Height="50"/> </Grid.RowDefinitions> <StackPanel Grid.Row="0" Background="#5d1d99"> <TextBlock Name="roleTextBlock" Style="{StaticResource profileTextBlock}" Grid.Row="0" Text="Your Role"/> <Button Style="{StaticResource refreshDataBtn}" Content="Refresh" Click="refreshClicked"/> <Button Style="{StaticResource signOutBtn}" Content="Sign Out" Click="signOutClicked"/> </StackPanel> <StackPanel Orientation="Horizontal" Grid.Row="1" Background="#461673"> <TextBlock Name="welcomingText" Style="{StaticResource nicknameTextBlockInProfile}" Text="Welcome, user!"/> </StackPanel> <StackPanel Orientation="Horizontal" Grid.Row="1" HorizontalAlignment="Right" Margin="0, 0, 5, 0" Width="120"> <Image x:Name="mainProfileImage" Height="80" Width="80" Margin="20, 0, 0, 50" Source="mainAssets\questionMark.png"/> </StackPanel> <StackPanel Orientation="Horizontal" Grid.Row="2" Background="#5d1d99"> <Button Name="changeNicknameBtn" Style="{StaticResource bottomBtnOnProfile}" Content="Change Nickname" Click="changeNicknameBtnClicked"/> <Button Name="addImgBtn" Style="{StaticResource bottomBtnOnProfile}" Content="Add Image" Click="addImageBtnClicked"/> <Button Name="changeImgBtn" Style="{StaticResource bottomBtnOnProfile}" Content="Change Image" Click="changeImageBtnClicked"/> </StackPanel> </Grid></Window>*

XAML код окна “Authorization.xaml” представлен ниже:

*<?xml version="1.0" encoding="utf-8"?><Window x:Class="internetShopProject.Authorization" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Title="Authorization" Height="400" Width="250" ResizeMode="CanMinimize"> <Grid> <Grid.RowDefinitions> <RowDefinition Height="60"/> <RowDefinition Height="\*"/> </Grid.RowDefinitions> <StackPanel Grid.Row="0" Background="#5d1d99"/> <StackPanel Orientation="Horizontal" Grid.Row="1" Background="#461673"> <Button Style="{StaticResource authorizeBtn}" Content="Authorize" Click="authorizeBtnClicked"/> </StackPanel> <TextBlock Name="textForLoginBox2" Margin="0, 27, 0, 70" Text="Login:" Grid.Column="0" Grid.Row="1" Style="{StaticResource textForLoginBox2}"/> <TextBlock Name="textForPasswordBox2" Margin="5, 90, 0, 70" Text="Password:" Grid.Column="0" Grid.Row="1" Style="{StaticResource textForPasswordBox2}"/> <TextBox Name="loginBox2" Grid.Row="1" Style="{StaticResource loginBox2}"/> <PasswordBox Name="passwordBox2" Grid.Row="1" PasswordChar="\*" Style="{StaticResource passwordBox2}"/> </Grid></Window>*

XAML код окна “Registration.xaml” представлен ниже:

*<?xml version="1.0" encoding="utf-8"?><Window x:Class="internetShopProject.Registration" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Title="Registration" Height="400" Width="250" ResizeMode="NoResize"> <Grid> <Grid.RowDefinitions> <RowDefinition Height="60"/> <RowDefinition Height="\*"/> </Grid.RowDefinitions> <StackPanel Grid.Row="0" Background="#5d1d99"/> <StackPanel Style="{StaticResource middlePanel}" Grid.Row="1"> <Button Style="{StaticResource createAccBtn}" Content="Create Account" Click="createAccountBtnClicked"/> </StackPanel> <TextBlock Name="textForLoginBox" Grid.Column="0" Grid.Row="1" Text="Login:" Style="{StaticResource textForLoginBox}"/> <TextBlock Name="textForPasswordBox" Grid.Column="0" Grid.Row="1" Text="Password:" Style="{StaticResource textForPasswordBox}"/> <TextBlock Name="textForRepeatPasswordBox" Grid.Column="0" Grid.Row="1" Text="Repeat Password:" Style="{StaticResource textForRepeatPasswordBox}"/> <TextBox Name="loginBox" Grid.Row="1" Style="{StaticResource loginBox}"/> <PasswordBox x:Name="passwordBox" Grid.Row="1" PasswordChar="\*" Style="{StaticResource passwordStyleBox}"/> <PasswordBox x:Name="repeatPasswordBox" Grid.Row="1" PasswordChar="\*" Style="{StaticResource repeatPasswordStyleBox}"/> </Grid></Window>*

XAML код окна “ShoppingCart.xaml” представлен ниже:

*<Window x:Class="internetShopProject.ShoppingCart" 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:internetShopProject" mc:Ignorable="d" Title="Shopping Cart" Height="450" Width="500" ResizeMode="CanMinimize" Loaded="shoppingCartWindowLoaded"> <Grid> <Grid.RowDefinitions> <RowDefinition Height="60"/> <RowDefinition Height="\*"/> <RowDefinition Height="50"/> </Grid.RowDefinitions> <StackPanel Orientation="Horizontal" Grid.Row="0" Background="#5d1d99"> <TextBlock Style="{StaticResource shoppingCartTextBlock}" Text="Shopping Cart"/> <Button Style="{StaticResource shoppingCartButton}" Content="Order" Click="orderButtonClicked"/> <Button Style="{StaticResource shoppingCartButton}" Content="Clear" Click="clearButtonClicked"/> </StackPanel> <StackPanel Orientation="Horizontal" Grid.Row="1" Background="#461673"> <ListBox Name="cartListViewForProducts" ItemsSource="{Binding cartProductCollection}" Width="500" Background="Transparent" BorderThickness="0" PreviewMouseDoubleClick="cartDoubleClicked"> <ListBox.ItemTemplate> <DataTemplate> <StackPanel> <TextBlock Name="cartProductNameTextBlock" Foreground="White" FontSize="16"> <Run Text="Product: "/> <Run Text="{Binding product\_Name}"/> </TextBlock> <TextBlock Name="cartProductDescTextBlock" Foreground="White" FontSize="16"> <Run Text="Description: "/> <Run Text="{Binding product\_Description}"/> </TextBlock> <TextBlock Name="cartProductCountTextBlock" Foreground="White" FontSize="16"> <Run Text="Count: "/> <Run Text="{Binding product\_Count}"/> </TextBlock> <TextBlock Name="cartProductPriceTextBlock" Foreground="White" FontSize="16"> <Run Text="Price: "/> <Run Text="{Binding product\_Price}"/> </TextBlock> </StackPanel> </DataTemplate> </ListBox.ItemTemplate> </ListBox> </StackPanel> <StackPanel Grid.Row="2" Background="#5d1d99"/> </Grid></Window>*

XAML код окна “changeNick.xaml” представлен ниже:

*<Window x:Class="internetShopProject.changeNick" 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:internetShopProject" mc:Ignorable="d" Title="Nick Changing" Height="200" Width="300" ResizeMode="CanMinimize"> <Grid Background="#461673"> <StackPanel Orientation="Vertical"> <TextBlock Text="New Nickname:" FontSize="20" Foreground="White" HorizontalAlignment="Center" Margin="0, 20, 0, 0"/> <TextBox Name="newNickTextBox" Background="BlueViolet" Width="150" Height="30" FontSize="16" BorderThickness="2" BorderBrush="#5d1d99" FontWeight="Bold" Foreground="White" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Margin="0, 5, 0, 0"/> <Button Background="BlueViolet" Width="150" Height="30" BorderThickness="0" HorizontalAlignment="Center" FontSize="14" Foreground="White" Content="Apply Changes" Margin="0, 25, 0, 0" Click="newNickBtn\_Clicked"/> </StackPanel> </Grid></Window>*

XAML код окна “addingProduct.xaml” представлен ниже:

*<Window x:Class="internetShopProject.addingProduct" 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:internetShopProject" mc:Ignorable="d" Title="Product Adding" Height="420" Width="250" ResizeMode="CanMinimize"> <Grid Background="#461673"> <StackPanel Orientation="Vertical"> <TextBlock Text="New Product:" FontSize="20" Foreground="White" HorizontalAlignment="Center" Margin="0, 20, 0, 0"/> <TextBox Name="newProductTextBox" Background="BlueViolet" Width="150" Height="30" FontSize="16" BorderThickness="2" BorderBrush="#5d1d99" FontWeight="Bold" Foreground="White" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Margin="0, 5, 0, 0" PreviewTextInput="wrongInputPrevented"/> <TextBlock Text="Description:" FontSize="20" Foreground="White" HorizontalAlignment="Center" Margin="0, 10, 0, 0"/> <TextBox Name="newDescriptionTextBox" Background="BlueViolet" Width="150" Height="30" FontSize="16" BorderThickness="2" BorderBrush="#5d1d99" FontWeight="Bold" Foreground="White" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Margin="0, 5, 0, 0"/> <TextBlock Text="Count:" FontSize="20" Foreground="White" HorizontalAlignment="Center" Margin="0, 10, 0, 0"/> <TextBox Name="newCountTextBox" Background="BlueViolet" Width="150" Height="30" FontSize="16" BorderThickness="2" BorderBrush="#5d1d99" FontWeight="Bold" Foreground="White" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Margin="0, 5, 0, 0" PreviewTextInput="textInputPrevented"/> <TextBlock Text="Price:" FontSize="20" Foreground="White" HorizontalAlignment="Center" Margin="0, 10, 0, 0"/> <TextBox Name="newPriceTextBox" Background="BlueViolet" Width="150" Height="30" FontSize="16" BorderThickness="2" BorderBrush="#5d1d99" FontWeight="Bold" Foreground="White" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Margin="0, 5, 0, 0" PreviewTextInput="textInputPrevented"/> <Button Background="BlueViolet" Width="150" Height="30" BorderThickness="0" HorizontalAlignment="Center" FontSize="14" Foreground="White" Content="Create Product" Margin="0, 25, 0, 0" Click="createPrdClicked"/> </StackPanel> </Grid></Window>*

XAML код окна “redactingProduct.xaml” представлен ниже:

*<Window x:Class="internetShopProject.redactingProduct" 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:internetShopProject" mc:Ignorable="d" Title="Product Redaction" Height="480" Width="250" ResizeMode="CanMinimize"> <Grid Background="#461673"> <StackPanel Orientation="Vertical"> <TextBlock Text="Old Product Name:" FontSize="20" Foreground="White" HorizontalAlignment="Center" Margin="0, 20, 0, 0"/> <TextBox Name="oldProductNameTextBox" Background="BlueViolet" Width="150" Height="30" FontSize="16" BorderThickness="2" BorderBrush="#5d1d99" FontWeight="Bold" Foreground="White" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Margin="0, 5, 0, 0" PreviewTextInput="wrongInputPrevented"/> <TextBlock Text="New Product Name:" FontSize="20" Foreground="White" HorizontalAlignment="Center" Margin="0, 10, 0, 0"/> <TextBox Name="newRedactProductNameTextBox" Background="BlueViolet" Width="150" Height="30" FontSize="16" BorderThickness="2" BorderBrush="#5d1d99" FontWeight="Bold" Foreground="White" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Margin="0, 5, 0, 0" PreviewTextInput="wrongInputPrevented"/> <TextBlock Text="New Product Description:" FontSize="20" Foreground="White" HorizontalAlignment="Center" Margin="0, 10, 0, 0"/> <TextBox Name="newRedactProductDescriptionTextBox" Background="BlueViolet" Width="150" Height="30" FontSize="16" BorderThickness="2" BorderBrush="#5d1d99" FontWeight="Bold" Foreground="White" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Margin="0, 5, 0, 0"/> <TextBlock Text="New Product Count:" FontSize="20" Foreground="White" HorizontalAlignment="Center" Margin="0, 10, 0, 0"/> <TextBox Name="newRedactProductCountTextBox" Background="BlueViolet" Width="150" Height="30" FontSize="16" BorderThickness="2" BorderBrush="#5d1d99" FontWeight="Bold" Foreground="White" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Margin="0, 5, 0, 0" PreviewTextInput="textInputPrevented"/> <TextBlock Text="New Product Price:" FontSize="20" Foreground="White" HorizontalAlignment="Center" Margin="0, 10, 0, 0"/> <TextBox Name="newRedactProductPriceTextBox" Background="BlueViolet" Width="150" Height="30" FontSize="16" BorderThickness="2" BorderBrush="#5d1d99" FontWeight="Bold" Foreground="White" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Margin="0, 5, 0, 0" PreviewTextInput="textInputPrevented"/> <Button Background="BlueViolet" Width="150" Height="30" BorderThickness="0" HorizontalAlignment="Center" FontSize="14" Foreground="White" Content="Redact Product" Margin="0, 25, 0, 0" Click="redactBtnClicked"/> </StackPanel> </Grid></Window>*

XAML код окна “removingProduct.xaml” представлен ниже:

*<Window x:Class="internetShopProject.removingProduct" 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:internetShopProject" mc:Ignorable="d" Title="Product Removing" Height="200" Width="250" ResizeMode="CanMinimize"> <Grid Background="#461673"> <StackPanel Orientation="Vertical"> <TextBlock Text="Product Name:" FontSize="20" Foreground="White" HorizontalAlignment="Center" Margin="0, 20, 0, 0"/> <TextBox Name="mainProductNameTextBox" Background="BlueViolet" Width="150" Height="30" FontSize="16" BorderThickness="2" BorderBrush="#5d1d99" FontWeight="Bold" Foreground="White" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Margin="0, 5, 0, 0"/> <Button Background="BlueViolet" Width="150" Height="30" BorderThickness="0" HorizontalAlignment="Center" FontSize="14" Foreground="White" Content="Delete Product" Margin="0, 25, 0, 0" Click="deleteProductClicked"/> </StackPanel> </Grid></Window>*

2.2 Разработка программного функционала

В данном курсовом проекте были разработаны “xaml.cs” классы кода, содержащие логику и поведение, связанные с соответствующими файлами разметки XAML. Разработанные “xaml.cs” классы кода: “App.xaml.cs”, “MainPage.xaml.cs”, “Products.xaml.cs”, “Profile.xaml.cs”, “Authorization.xaml.cs”, “Registration.xaml.cs”, “ShoppingCart.xaml.cs”, “changeNick.xaml.cs”, “addingProduct.xaml.cs”, “redactingProduct.xaml.cs”, “removingProduct.xaml.cs”. “xaml.cs” код окна “App.xaml.cs” представлен ниже:

using System;using System.Windows;using System.Data;using System.Xml;using System.Configuration;using System.Xml.Linq;using System.Collections.Generic;namespace internetShopProject{ public partial class App : Application { // Создание "контекста" для обращения к элементам базы данных (таблицам и столбцам в них) public static mainEntities.internetShop\_DBEntities3 Context { get; } = new mainEntities.internetShop\_DBEntities3(); }}

“xaml.cs” код окна “MainPage.xaml.cs” представлен ниже:

using internetShopProject;using System;using System.Collections.Generic;using System.Linq;using System.Text;using System.Windows;using System.Windows.Controls;using System.Windows.Data;using System.Windows.Documents;using System.Windows.Input;using System.Windows.Media;namespace internetShopProject{ public partial class MainPage : Window { public MainPage() { InitializeComponent(); // Иницилизация значений глобального списка программы App.Current.Properties["productsEnabled"] = 0; App.Current.Properties["userRole"] = 0; App.Current.Properties["nicknameInSession"] = "None"; } void createAccClicked(object sender, RoutedEventArgs e) { // Создание нового регистрационного окна Registration regWindow = new Registration(); regWindow.Show(); } void authorizeIntoAccClicked(object sender, RoutedEventArgs e) { // Проверка на существующую сессию авторизации на данный момент в системе int authorizedInSystem = (int)App.Current.Properties["productsEnabled"]; if (authorizedInSystem > 0) { MessageBox.Show("You are already authorized in system!"); return; } else { // Создание нового авторизационного окна Authorization authWindow = new Authorization(); authWindow.Show(); } } protected override void OnClosed(EventArgs e) { // При закрытии программы, уничтожается общий процесс программы и все окна в процессе base.OnClosed(e); Application.Current.Shutdown(); } void cartBtnClicked(object sender, RoutedEventArgs e) { // Получение доступа к кнопке "Cart" посредством проверки значения глобального списка программы int productsEnabledValue = (int)App.Current.Properties["productsEnabled"]; // Получение доступа к окну "Products" для последующих манипуляций Window productsWindowIsOpened = App.Current.Windows.OfType<Products>().FirstOrDefault(); if (productsEnabledValue == 0) { MessageBox.Show("Please authorize in the system to view shopping cart!"); } else if (productsWindowIsOpened == null) { MessageBox.Show("Please open the \"Products\" window to work with a \"Cart\"!"); } else { // Проверка на количество открытых окон корзины int cartWindowCounting = 0; foreach (Window window in Application.Current.Windows) { if (window.GetType() == typeof(ShoppingCart)) { cartWindowCounting++; } } if (cartWindowCounting == 1) { cartWindowCounting = 0; return; } else { // Создание нового окна корзины ShoppingCart cartWindow = new ShoppingCart(); cartWindow.Show(); } } } void productsBtnClicked(object sender, RoutedEventArgs e) { // Получение доступа к кнопке "Products" посредством проверки значения глобального списка программы int productsEnabledValue = (int)App.Current.Properties["productsEnabled"]; if (productsEnabledValue == 0) { MessageBox.Show("Please authorize in the system to view products!"); } else { // Проверка на количество открытых окон продуктов int productsWindowCounting = 0; foreach (Window window in Application.Current.Windows) { if (window.GetType() == typeof(Products)) { productsWindowCounting++; } } if (productsWindowCounting == 1) { productsWindowCounting = 0; return; } else { // Создание нового окна продукции Products productsWindow = new Products(); productsWindow.Show(); } } } }}

“xaml.cs” код окна “Products.xaml.cs” представлен ниже:

using internetShopProject.mainEntities;using System;using System.Collections.Generic;using System.Collections.ObjectModel;using System.ComponentModel;using System.Data.Entity;using System.Linq;using System.Reflection;using System.Runtime.Remoting.Contexts;using System.Text;using System.Threading.Tasks;using System.Windows;using System.Windows.Controls;using System.Windows.Data;using System.Windows.Documents;using System.Windows.Input;using System.Windows.Media;using System.Windows.Media.Imaging;using System.Windows.Shapes;using System.Xml.Linq;namespace internetShopProject{ public partial class Products : Window { public Products() { InitializeComponent(); // Иницилизация коллекции для продуктов корзины ObservableCollection<product> cartProductList = new ObservableCollection<product>(); App.Current.Properties["listOfProductsForCart"] = cartProductList; // Проверка роли пользователя на соответствие int userRoleValue = (int)App.Current.Properties["userRole"]; if (userRoleValue == 1) { addProductButton.Visibility = Visibility.Collapsed; removeProductButton.Visibility = Visibility.Collapsed; changeProductButton.Visibility = Visibility.Collapsed; } // Иницилизация коллекции для продуктов ObservableCollection<product> productsCollection = new ObservableCollection<product>(); var productsList = App.Context.products.ToList(); App.Current.Properties["mainProductsCollection"] = productsCollection; // Занесение полученных данных из базы данных в коллекцию foreach (var product in productsList) { productsCollection.Add(new product { product\_Name = product.product\_Name, product\_Count = product.product\_Count, product\_Description = product.product\_Description, product\_Price = product.product\_Price }); } listViewForProducts.ItemsSource = productsCollection; } void productDoubleClicked(object sender, RoutedEventArgs e) { Window detectingCart = App.Current.Windows.OfType<ShoppingCart>().FirstOrDefault(); if (detectingCart != null) { // Получение выбранного элемента ListBox listBox = sender as ListBox; object selectedItem = listBox.SelectedItem; product selectedProduct = selectedItem as product; // Уменьшение количества имеющихся продуктов в приложении if (selectedProduct.product\_Count != 0) { // Формирование коллекций данных для корзины и продуктов пользователя ObservableCollection<product> cartProductCollectionForCheckingProducts = App.Current.Properties["listOfProductsForCart"] as ObservableCollection<product>; ObservableCollection<product> prdCollectionForCheckingProducts = App.Current.Properties["mainProductsCollection"] as ObservableCollection<product>; // Цикл для манипуляции с данными удаления и добавления значений foreach (product itemInCart in cartProductCollectionForCheckingProducts) { if (itemInCart.product\_Name == selectedProduct.product\_Name) { selectedProduct.product\_Count -= 1; itemInCart.product\_Count += 1; ICollectionView cartProductRefreshView = CollectionViewSource.GetDefaultView(cartProductCollectionForCheckingProducts); cartProductRefreshView.Refresh(); ICollectionView prdRefreshView = CollectionViewSource.GetDefaultView(prdCollectionForCheckingProducts); prdRefreshView.Refresh(); return; } } } else { // Если продукт закончился, конец выполнения дальнейшего скрипта MessageBox.Show("This product is over!"); return; } // Иницилизация коллекции для её последующего обновления ObservableCollection<product> productToRefreshCollection = App.Current.Properties["mainProductsCollection"] as ObservableCollection<product>; ICollectionView viewForRefresh = CollectionViewSource.GetDefaultView(productToRefreshCollection); viewForRefresh.Refresh(); // Создание словаря для последующих манипуляций с данными Dictionary<string, string> productInfoDictionary = new Dictionary<string, string> { { "productName", selectedProduct.product\_Name }, { "productCount", selectedProduct.product\_Count.ToString() }, { "productDescription", selectedProduct.product\_Description }, { "productPrice", selectedProduct.product\_Price.ToString() } }; // Формирование коллекции данных для корзины пользователя ObservableCollection<product> cartProductList = App.Current.Properties["listOfProductsForCart"] as ObservableCollection<product>; cartProductList.Add(new product { product\_Name = productInfoDictionary["productName"], product\_Count = 1, product\_Description = productInfoDictionary["productDescription"], product\_Price = selectedProduct.product\_Price }); selectedProduct.product\_Count -= 1; // Вставка коллекции данных в корзину пользователя Window cartWindow = App.Current.Windows.OfType<ShoppingCart>().FirstOrDefault(); if (cartWindow != null) { ListBox listBoxOfCartWindow = cartWindow.FindName("cartListViewForProducts") as ListBox; listBoxOfCartWindow.ItemsSource = cartProductList; } else { MessageBox.Show("Open the \"Shopping Cart\" window to buy products!"); } } } void refreshProductClicked(object sender, RoutedEventArgs e) { // Иницилизация коллекции для её последующего обновления ObservableCollection<product> productToRefreshCollection = App.Current.Properties["mainProductsCollection"] as ObservableCollection<product>; // Проход по коллекции и проверка на закончившиеся продукты в коллекции (если закончились, запись удаляется из коллекции) foreach (product productInProducts in productToRefreshCollection) { if (productInProducts.product\_Count == 0) { productToRefreshCollection.Remove(productInProducts); break; } } // Обновление view коллекции продуктов ICollectionView viewForRefresh = CollectionViewSource.GetDefaultView(productToRefreshCollection); viewForRefresh.Refresh(); } void removePrdClicked(object sender, RoutedEventArgs e) { double top = Top; double left = Left; // Создание нового окна удаления продукта removingProduct deleteProductWindow = new removingProduct(); deleteProductWindow.Top = top; deleteProductWindow.Left = left - 250; deleteProductWindow.Show(); } void addPrdClicked(object sender, RoutedEventArgs e) { double top = Top; double left = Left; // Создание нового окна добавления продукта addingProduct addProductWindow = new addingProduct(); addProductWindow.Top = top; addProductWindow.Left = left - 250; addProductWindow.Show(); } void changePrdClicked(object sender, RoutedEventArgs e) { double top = Top; double left = Left; // Создание нового окна изменения продукта redactingProduct changeProductWindow = new redactingProduct(); changeProductWindow.Top = top; changeProductWindow.Left = left - 250; changeProductWindow.Show(); } }}

“xaml.cs” код окна “Profile.xaml.cs” представлен ниже:

using System;using System.Collections.Generic;using System.Text;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.Data.SqlClient;using System.Linq;using Microsoft.Win32;using static System.Net.Mime.MediaTypeNames;using System.Windows.Media.Imaging;using System.IO;using internetShopProject.mainEntities;using iText.IO.Image;namespace internetShopProject{ public partial class Profile : Window { public Profile() { InitializeComponent(); // Вывод ника в форму профиля при входе в профильное окно string nickInSession = (string)App.Current.Properties["nicknameInSession"]; TextBlock welcomingText = (TextBlock)FindName("welcomingText"); string resultMessage = "Welcome, " + nickInSession + "!"; welcomingText.Text = resultMessage; // Запрос на выборку роли пользователя с последующим выводом на форму профиля var role = App.Context.users.Where(u => u.user\_Nickname == nickInSession).Select(r => r.role\_Id).FirstOrDefault(); if (role != 0) { if (role == 1) { TextBlock welcomingRole = (TextBlock)FindName("roleTextBlock"); string resultMessageOfRole = "Role: User"; welcomingRole.Text = resultMessageOfRole; App.Current.Properties["userRole"] = 1; } else { TextBlock welcomingRole = (TextBlock)FindName("roleTextBlock"); string resultMessageOfRole = "Role: Admin"; welcomingRole.Text = resultMessageOfRole; App.Current.Properties["userRole"] = 2; } } else { MessageBox.Show("User role wasn't found!"); } // Запрос на проверку имеющейся картинки в базе данных (присутствует, либо отсутствует) var gottenImg = App.Context.users.Where(u => u.user\_Nickname == nickInSession).Select(i => i.user\_Image).FirstOrDefault(); if (gottenImg != null) { // Если картинка имеется addImgBtn.Visibility = Visibility.Collapsed; // Считываем двоичные данные картинки из базы данных MemoryStream ms = new MemoryStream(gottenImg); BitmapImage image = new BitmapImage(); image.BeginInit(); image.StreamSource = ms; image.CacheOption = BitmapCacheOption.OnLoad; image.EndInit(); mainProfileImage.Source = image; } else { // Если картинка отсутствует changeImgBtn.Visibility = Visibility.Collapsed; } } void changeNicknameBtnClicked(object sender, RoutedEventArgs e) { double top = Top; double left = Left; // Создание нового окна изменения никнейма changeNick chNickWindow = new changeNick(); chNickWindow.Top = top + 300; chNickWindow.Left = left; chNickWindow.Show(); } void addImageBtnClicked(object sender, RoutedEventArgs e) { // Функция добавления картинки в профиль (соответственно, при отсутствии картинки в профиле) string nickInSession = (string)App.Current.Properties["nicknameInSession"]; // Открытие файлового диалога для выбора желаемой картинки, перенос на директорию "Рабочего стола" OpenFileDialog openFileDialog = new OpenFileDialog(); openFileDialog.Filter = "Image Files (\*.bmp, \*.jpg, \*.png)|\*.bmp;\*.jpg;\*.png"; openFileDialog.InitialDirectory = @"C:\Desktop\"; Nullable<bool> result = openFileDialog.ShowDialog(); // Условный оператор успешного выбора картинки if (result == true) { // Получение пути к картинке и её иницилизация в программу string selectedFileName = openFileDialog.FileName; // Записываем двоичные данные картинки для базы данных BitmapImage image = new BitmapImage(); image.BeginInit(); image.UriSource = new Uri(selectedFileName); image.EndInit(); mainProfileImage.Source = image; // Добавление выбранной картинки в базу данных byte[] imageData = File.ReadAllBytes(selectedFileName); var foundNickname = App.Context.users.FirstOrDefault(n => n.user\_Nickname == nickInSession); if (foundNickname != null) { foundNickname.user\_Image = imageData; App.Context.SaveChanges(); } // Смена видимости с кнопки добавления картинки на кнопку изменения картинки addImgBtn.Visibility = Visibility.Collapsed; changeImgBtn.Visibility = Visibility.Visible; MessageBox.Show("An image was successfully added to your profile!"); } } void changeImageBtnClicked(object sender, RoutedEventArgs e) { // Функция изменения картинки в профиле (соответственно, при присутствии картинки в профиле) string nickInSession = (string)App.Current.Properties["nicknameInSession"]; // Открытие файлового диалога для выбора желаемой картинки, перенос на директорию "Рабочего стола" OpenFileDialog openFileDialog = new OpenFileDialog(); openFileDialog.Filter = "Image Files (\*.bmp, \*.jpg, \*.png)|\*.bmp;\*.jpg;\*.png"; openFileDialog.InitialDirectory = @"C:\Desktop\"; Nullable<bool> result = openFileDialog.ShowDialog(); // Условный оператор успешного выбора картинки if (result == true) { // Получение пути к картинке и её иницилизация в программу string selectedFileName = openFileDialog.FileName; // Записываем двоичные данные картинки для базы данных BitmapImage image = new BitmapImage(); image.BeginInit(); image.UriSource = new Uri(selectedFileName); image.EndInit(); mainProfileImage.Source = image; // Изменение существующей картинки в базе данных на новую byte[] imageData = File.ReadAllBytes(selectedFileName); var foundNickname = App.Context.users.FirstOrDefault(n => n.user\_Nickname == nickInSession); if (foundNickname != null) { foundNickname.user\_Image = imageData; App.Context.SaveChanges(); } MessageBox.Show("An image was successfully changed in your profile!"); } } void refreshClicked(object sender, RoutedEventArgs e) { // Обновление данных (ника) пользователя при нажатии на кнопку "Refresh" string nickInSession = (string)App.Current.Properties["nicknameInSession"]; TextBlock welcomingText = (TextBlock)FindName("welcomingText"); string resultMessage = "Welcome, " + nickInSession + "!"; welcomingText.Text = resultMessage; var foundRole = App.Context.users.Where(u => u.user\_Nickname == nickInSession).Select(ur => ur.role\_Id).FirstOrDefault(); if (foundRole == 1) { TextBlock welcomingRole = (TextBlock)FindName("roleTextBlock"); string resultMessageOfRole = "Role: User"; welcomingRole.Text = resultMessageOfRole; App.Current.Properties["userRole"] = 1; } else { TextBlock welcomingRole = (TextBlock)FindName("roleTextBlock"); string resultMessageOfRole = "Role: Admin"; welcomingRole.Text = resultMessageOfRole; App.Current.Properties["userRole"] = 2; } } void signOutClicked(object sender, RoutedEventArgs e) { // Установление запрета на доступ к кнопке и вкладке "Products" для гостя App.Current.Properties["productsEnabled"] = 0; double top = Top; double left = Left; // Цикл очистки окон при санкционированном закрытии профиля с помощью кнопки "Sign Out" var windowsToClose = new List<Window>(); windowsToClose.AddRange(App.Current.Windows.OfType<Products>()); windowsToClose.AddRange(App.Current.Windows.OfType<changeNick>()); windowsToClose.AddRange(App.Current.Windows.OfType<ShoppingCart>()); windowsToClose.AddRange(App.Current.Windows.OfType<addingProduct>()); windowsToClose.AddRange(App.Current.Windows.OfType<removingProduct>()); windowsToClose.AddRange(App.Current.Windows.OfType<redactingProduct>()); foreach (var window in windowsToClose) { window.Close(); } windowsToClose = null; Close(); } protected override void OnClosed(EventArgs e) { // Установление запрета на доступ к кнопке и вкладке "Products" для гостя App.Current.Properties["productsEnabled"] = 0; double top = Top; double left = Left; // Цикл очистки окон при принудительном закрытии профиля var windowsToClose = new List<Window>(); windowsToClose.AddRange(App.Current.Windows.OfType<Products>()); windowsToClose.AddRange(App.Current.Windows.OfType<changeNick>()); windowsToClose.AddRange(App.Current.Windows.OfType<ShoppingCart>()); windowsToClose.AddRange(App.Current.Windows.OfType<addingProduct>()); windowsToClose.AddRange(App.Current.Windows.OfType<removingProduct>()); windowsToClose.AddRange(App.Current.Windows.OfType<redactingProduct>()); foreach (var window in windowsToClose) { window.Close(); } windowsToClose = null; base.OnClosed(e); Close(); } }}

“xaml.cs” код окна “Authorization.xaml.cs” представлен ниже:

using System;using System.Collections.Generic;using System.Linq;using System.Text;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.Xml;using System.Data.SqlClient;using System.Security.Cryptography;namespace internetShopProject{ public partial class Authorization : Window { public Authorization() { InitializeComponent(); } void authorizeBtnClicked(object sender, RoutedEventArgs e) { // Запрос на проверку существующего ника при авторизации var nicknameSearch = App.Context.users.Where(u => u.user\_Nickname == loginBox2.Text).Select(u => u.user\_Nickname).FirstOrDefault(); if (nicknameSearch != null) { // Процедура хэширования пароля в MD5 для сравнения паролей MD5 md5Hash = MD5.Create(); byte[] data = md5Hash.ComputeHash(Encoding.UTF8.GetBytes(passwordBox2.Password)); StringBuilder sb = new StringBuilder(); for (int i = 0; i < data.Length; i++) { sb.Append(data[i].ToString("x2")); } string hashedPassword = sb.ToString(); // Конкретный запрос на проверку соответствия введёных пользователем данных при авторизации (логин и пароль) var passwordSearch = App.Context.users.Where(u => u.user\_Nickname == loginBox2.Text && u.user\_Password == hashedPassword).Select(u => u.user\_Password).FirstOrDefault(); if (passwordSearch != null) { // Разрешение пользования кнопкой "Products" и соответствующим окном продукции после авторизации // Занесение логина в сессию никнейма в программу App.Current.Properties["productsEnabled"] = 1; App.Current.Properties["nicknameInSession"] = loginBox2.Text; MessageBox.Show("Valid data!"); this.Hide(); // Создание нового профильного окна Profile profileWindow = new Profile(); profileWindow.Show(); } else { MessageBox.Show("Invalid password!"); } } else { MessageBox.Show("Invalid login!"); } } }}

“xaml.cs” код окна “Registration.xaml.cs” представлен ниже:

using System;using System.Collections.Generic;using System.Data.SqlClient;using System.Linq;using System.Runtime.Remoting.Contexts;using System.Security.Cryptography;using System.Text;using System.Windows;using System.Windows.Controls;using System.Windows.Data;using System.Windows.Documents;using System.Windows.Input;using System.Windows.Media;namespace internetShopProject{ public partial class Registration : Window { public Registration() { InitializeComponent(); } void createAccountBtnClicked(object sender, RoutedEventArgs e) { // Метод, подсчитывающий соответствующее количество каждого необходимого условия в пароле string text = passwordBox.Password; int upperLetters = 0; int lowerLetters = 0; int digitCount = 0; int specialLetters = 0; foreach (char c in text) { if (Char.IsUpper(c)) { upperLetters++; } if (Char.IsLower(c)) { lowerLetters++; } if (Char.IsDigit(c)) { digitCount++; } if (!Char.IsLetterOrDigit(c)) { specialLetters++; } } // Запрос на проверку существующего ника при регистрации var nicknameSearch = App.Context.users.Where(u => u.user\_Nickname == loginBox.Text).Select(u => u.user\_Nickname).FirstOrDefault(); if (nicknameSearch != null) { MessageBox.Show("This nickname already was taken!"); } else { // Конкретные проверки заполненных данных при регистрации (логин, пароль, повторный пароль) if (loginBox.Text == "" || passwordBox.Password == "" || repeatPasswordBox.Password == "") { MessageBox.Show("Fill the empty boxes!"); } else if (passwordBox.Password != repeatPasswordBox.Password) { MessageBox.Show("Passwords must be the same!"); } else if (loginBox.Text.Length > 20 || passwordBox.Password.Length > 20) { MessageBox.Show("Login and password must be maximum 20 symbols!"); } else if (loginBox.Text.Length < 8 || passwordBox.Password.Length < 8) { MessageBox.Show("Login and password must be minimum 8 symbols!"); } else if (loginBox.Text.Contains(" ") || passwordBox.Password.Contains(" ")) { MessageBox.Show("Login and password must be wrote without spaces!"); } else if (upperLetters < 1) { MessageBox.Show("There are must be atleast one \"Upper Letter\" in password!"); } else if (lowerLetters < 1) { MessageBox.Show("There are must be atleast one \"Lower Letter\" in password!"); } else if (digitCount < 1) { MessageBox.Show("There are must be atleast one \"Digit\" in password!"); } else if (specialLetters < 1) { MessageBox.Show("There are must be atleast one \"Special Letter\" in password!"); } else { // Процедура хэширования пароля в MD5 MD5 md5Hash = MD5.Create(); byte[] data = md5Hash.ComputeHash(Encoding.UTF8.GetBytes(passwordBox.Password)); StringBuilder sb = new StringBuilder(); for (int i = 0; i < data.Length; i++) { sb.Append(data[i].ToString("x2")); } string hashedPassword = sb.ToString(); // Процедура добавления пользователя в базу данных с ролью пользователя var userAddingProcedure = new mainEntities.user { user\_Nickname = loginBox.Text, user\_Password = hashedPassword, role\_Id = 1 }; // Сохранение осуществлённых изменений для базы данных App.Context.users.Add(userAddingProcedure); App.Context.SaveChanges(); MessageBox.Show("Account was successfully registered!"); this.Hide(); // Создание нового авторизационного окна Authorization authWindow = new Authorization(); authWindow.Show(); } } // Обнуление подсчитанных в имеющемся методе значений upperLetters = 0; lowerLetters = 0; digitCount = 0; specialLetters = 0; } }}

“xaml.cs” код окна “ShoppingCart.xaml.cs” представлен ниже:

using internetShopProject.mainEntities;using System;using System.Collections.Generic;using System.Collections.ObjectModel;using System.ComponentModel;using System.Linq;using System.Text;using System.Threading.Tasks;using System.Windows;using System.Windows.Controls;using System.Windows.Data;using System.Windows.Input;using System.Windows.Media;using System.Windows.Media.Imaging;using System.Windows.Shapes;using iText.Kernel.Pdf;using iText.Layout;using iText.Layout.Element;using System.IO;namespace internetShopProject{ public partial class ShoppingCart : Window { public ShoppingCart() { InitializeComponent(); } void shoppingCartWindowLoaded(object sender, RoutedEventArgs e) { // Занесение в коллекцию соответствующих продуктов и обновление вида ObservableCollection<product> shoppingCartProductList = App.Current.Properties["listOfProductsForCart"] as ObservableCollection<product>; cartListViewForProducts.ItemsSource = shoppingCartProductList; ICollectionView viewForRefresh = CollectionViewSource.GetDefaultView(shoppingCartProductList); viewForRefresh.Refresh(); } void clearButtonClicked(object sender, RoutedEventArgs e) { // Принудительная очистка коллекции от товаров ObservableCollection<product> prdOfShopCartCollection = App.Current.Properties["listOfProductsForCart"] as ObservableCollection<product>; ObservableCollection<product> prdCollection = App.Current.Properties["mainProductsCollection"] as ObservableCollection<product>; // Проход по коллекциям и обновление данных в основной коллекции foreach (product productInProducts in prdCollection) { foreach (product productInCart in prdOfShopCartCollection) { if (productInProducts.product\_Name == productInCart.product\_Name) { productInProducts.product\_Count += productInCart.product\_Count; prdOfShopCartCollection.Remove(productInCart); ICollectionView cartProductRefreshView = CollectionViewSource.GetDefaultView(prdCollection); cartProductRefreshView.Refresh(); break; } } } } void orderButtonClicked(object sender, RoutedEventArgs e) { ObservableCollection<product> cartProductList = App.Current.Properties["listOfProductsForCart"] as ObservableCollection<product>; if (cartProductList.Count == 0) { MessageBox.Show("You have no products to order in a \"Shopping Cart\""); } else { // Процедура формирования PDF-отчёта (чека) о покупке имеющихся в корзине продуктов пользователем // Создание PDF-файла PdfWriter pdfWriter = new PdfWriter("..\\..\\mainDocuments\\productCheck.pdf"); PdfDocument pdfDocument = new PdfDocument(pdfWriter); // Создание документа Document document = new Document(pdfDocument); // Добавление контента внутрь созданного документа string nickInSession = (string)App.Current.Properties["nicknameInSession"]; DateTime currentDate = DateTime.Now; // Заголовок контента чека Paragraph mainParagraph = new Paragraph("INTERNET SHOP SYSTEM PRODUCT CHECK" + "\n\n"); mainParagraph.SetBold(); // Текст в заголовке контента чека mainParagraph.Add(new Text("COUNT TYPES OF PRODUCTS BOUGHT: " + cartProductList.Count + "\n")); mainParagraph.Add(new Text("DATE OF A PURCHASE: " + currentDate + "\n")); mainParagraph.Add(new Text("BUYER NICKNAME: " + nickInSession)); document.Add(mainParagraph); // Данные о продуктах (заголовок) Paragraph secondaryParagraph = new Paragraph("PRODUCT DATA"); secondaryParagraph.SetBold(); document.Add(secondaryParagraph); Paragraph thirdParagraph = new Paragraph(); List<Paragraph> temporaryListForProducts = new List<Paragraph>(); // Вставка продуктов из корзины в созданный документ foreach (product cartProduct in cartProductList) { var mainProductText = new Paragraph("Product: " + cartProduct.product\_Name + "\nDescription: " + cartProduct.product\_Description + "\nCount: " + cartProduct.product\_Count.ToString() + "\nPrice for one: " + cartProduct.product\_Price.ToString() + "\n"); temporaryListForProducts.Add(mainProductText); } for (int i = 0; i < temporaryListForProducts.Count; i++) { document.Add(temporaryListForProducts[i]); } // Итоговая сумма всей покупки Paragraph fourthParagraph = new Paragraph(); List<int> temporaryListCounts = new List<int>(); foreach (product cartProduct in cartProductList) { int countedValueForOneProduct = (int)(cartProduct.product\_Price \* cartProduct.product\_Count); temporaryListCounts.Add(countedValueForOneProduct); } Paragraph resultedSumma = new Paragraph("RESULT PRICE: " + temporaryListCounts.Sum().ToString()); resultedSumma.SetBold(); document.Add(resultedSumma); // Закрытие документа document.Close(); ObservableCollection<product> prdOfShopCartCollection = App.Current.Properties["listOfProductsForCart"] as ObservableCollection<product>; ObservableCollection<product> prdCollection = App.Current.Properties["mainProductsCollection"] as ObservableCollection<product>; // Проход по коллекциям и очистка данных из приложения + базы данных (купленного количества) foreach (product productInProducts in prdCollection) { foreach (product productInCart in prdOfShopCartCollection) { if (productInProducts.product\_Name == productInCart.product\_Name) { // Процедура изменения количества продукции в базе данных после формирования отчёта (запрос) var foundProduct = App.Context.products.FirstOrDefault(p => p.product\_Name == productInCart.product\_Name); if (foundProduct != null) { foundProduct.product\_Count -= productInCart.product\_Count; App.Context.SaveChanges(); } prdOfShopCartCollection.Remove(productInCart); break; } } } MessageBox.Show("Product check was successfully created!"); } } void cartDoubleClicked(object sender, RoutedEventArgs e) { Window detectingProduct = App.Current.Windows.OfType<Products>().FirstOrDefault(); if (detectingProduct != null) { // Получение выбранного элемента ListBox listBox = sender as ListBox; object selectedItem = listBox.SelectedItem; product selectedProduct = selectedItem as product; // Уменьшение количества имеющихся продуктов из корзины if (selectedProduct.product\_Count > 1 || selectedProduct.product\_Count == 1) { // Формирование коллекций данных для корзины и продуктов пользователя ObservableCollection<product> cartProductCollectionForCheckingProducts = App.Current.Properties["listOfProductsForCart"] as ObservableCollection<product>; ObservableCollection<product> prdCollectionForCheckingProducts = App.Current.Properties["mainProductsCollection"] as ObservableCollection<product>; // Процедура удаления записи из коллекции корзины, если осталась 1 штука продукта if (selectedProduct.product\_Count == 1) { cartProductCollectionForCheckingProducts.Remove(selectedProduct); foreach (product itemInProducts in prdCollectionForCheckingProducts) { if (itemInProducts.product\_Name == selectedProduct.product\_Name) { itemInProducts.product\_Count += 1; ICollectionView prdRefreshView = CollectionViewSource.GetDefaultView(prdCollectionForCheckingProducts); prdRefreshView.Refresh(); return; } } } // Цикл для манипуляции с данными удаления и добавления значений foreach (product itemInProducts in prdCollectionForCheckingProducts) { if (itemInProducts.product\_Name == selectedProduct.product\_Name) { itemInProducts.product\_Count += 1; selectedProduct.product\_Count -= 1; ICollectionView cartProductRefreshView = CollectionViewSource.GetDefaultView(cartProductCollectionForCheckingProducts); cartProductRefreshView.Refresh(); ICollectionView prdRefreshView = CollectionViewSource.GetDefaultView(prdCollectionForCheckingProducts); prdRefreshView.Refresh(); } } } } } }}

“xaml.cs” код окна “changeNick.xaml.cs” представлен ниже:

using System;using System.Collections.Generic;using System.Linq;using System.Text;using System.Threading.Tasks;using System.Windows;using System.Windows.Controls;using System.Windows.Data;using System.Windows.Documents;using System.Windows.Input;using System.Windows.Media;using System.Windows.Media.Imaging;using System.Windows.Shapes;using System.Data.SqlClient;using System.Runtime.Remoting.Messaging;using System.Data.Entity.Migrations;namespace internetShopProject{ public partial class changeNick : Window { public changeNick() { InitializeComponent(); } void newNickBtn\_Clicked(object sender, RoutedEventArgs e) { string nickInSession = (string)App.Current.Properties["nicknameInSession"]; // Запрос на выборку поиска ника пользователя для его последующего изменения var nicknameSearch = App.Context.users.Where(u => u.user\_Nickname == newNickTextBox.Text).Select(u => u.user\_Nickname).FirstOrDefault(); if (nicknameSearch != null) { MessageBox.Show("This nickname already was taken!"); } else { // Метод, подсчитывающий соответствующее количество каждого необходимого условия в новом заполненном нике string newNick = newNickTextBox.Text; int upperLetters = 0; int lowerLetters = 0; int digitCount = 0; int specialLetters = 0; foreach (char c in newNick) { if (Char.IsUpper(c)) { upperLetters++; } if (Char.IsLower(c)) { lowerLetters++; } if (Char.IsDigit(c)) { digitCount++; } if (!Char.IsLetterOrDigit(c)) { specialLetters++; } } // Конкретные проверки заполненного нового ника if (newNickTextBox.Text == "") { MessageBox.Show("Fill the empty box!"); } else if (newNickTextBox.Text.Length > 20) { MessageBox.Show("New nickname must be maximum 20 symbols!"); } else if (newNickTextBox.Text.Length < 8) { MessageBox.Show("New nickname must be minimum 8 symbols!"); } else if (newNickTextBox.Text.Contains(" ")) { MessageBox.Show("New nickname must be wrote without spaces!"); } else { // Процедура изменения ника с занесением нового ника в базу данных var foundNickname = App.Context.users.FirstOrDefault(n => n.user\_Nickname == nickInSession); if (foundNickname != null) { foundNickname.user\_Nickname = newNickTextBox.Text; App.Context.SaveChanges(); } // Занесение нового ника в одно из значений глобального списка программы для регулирования данных (сессия с ником) MessageBox.Show("Nick \"" + nickInSession + "\"" + " was successfully changed on \"" + newNickTextBox.Text + "\"."); App.Current.Properties["nicknameInSession"] = newNickTextBox.Text; this.Hide(); } // Обнуление подсчитанных в имеющемся методе значений upperLetters = 0; lowerLetters = 0; digitCount = 0; specialLetters = 0; } } }}

“xaml.cs” код окна “addingProduct.xaml.cs” представлен ниже:

using internetShopProject.mainEntities;using System;using System.Collections.Generic;using System.Collections.ObjectModel;using System.ComponentModel;using System.Linq;using System.Text;using System.Threading.Tasks;using System.Windows;using System.Windows.Controls;using System.Windows.Data;using System.Windows.Documents;using System.Windows.Input;using System.Windows.Media;using System.Windows.Media.Imaging;using System.Windows.Shapes;namespace internetShopProject{ public partial class addingProduct : Window { public addingProduct() { InitializeComponent(); } void textInputPrevented(object sender, TextCompositionEventArgs e) { // Проверка на написание символов (необходимый символ - цифра, если иной - не осуществляет процедуру введения желаемого символа) if (!char.IsDigit(e.Text, e.Text.Length - 1)) { e.Handled = true; } } void wrongInputPrevented(object sender, TextCompositionEventArgs e) { // Проверка на написание символов (необходимый символ - буква, если иной - не осуществляет процедуру введения желаемого символа) if (!char.IsLetter(e.Text, e.Text.Length - 1)) { e.Handled = true; } } void createPrdClicked(object sender, RoutedEventArgs e) { // Иницилизация коллекции продуктов для последующих манипуляций ObservableCollection<product> prdCollection = App.Current.Properties["mainProductsCollection"] as ObservableCollection<product>; // Иницилизация переменных, принимающих введённые администратором значения string newPrdName = newProductTextBox.Text, newPrdDesc = newDescriptionTextBox.Text, newPrdCount = newCountTextBox.Text, newPrdPrice = newPriceTextBox.Text; // Проверка на отсутствие значения в TextBoxes, проверка на существование пробелов в несанкционированных для этого TextBoxes if (newPrdName == "" || newPrdDesc == "" || newPrdCount == "" || newPrdPrice == "") { MessageBox.Show("Please fill all boxes of this form!"); } else if (newPrdName.Contains(" ") || newPrdCount.Contains(" ") || newPrdPrice.Contains(" ")) { MessageBox.Show("There should be no spaces in this form beside description!"); } else if (newPrdName.Length > 50) { MessageBox.Show("Too big size of product name!"); } else { // Парсинг значения из типа string в тип int для добавления нового продукта в коллекцию и базу данных try { // Значения для проверки на исключение FormatException int checkCountNumber = int.Parse(newPrdCount); int checkPriceNumber = int.Parse(newPrdPrice); } catch (FormatException) { MessageBox.Show("Unacceptable types of data in count or price forms!"); return; } int newPrdCountNumber = int.Parse(newPrdCount); int newPrdPriceNumber = int.Parse(newPrdPrice); // Добавление в коллекцию prdCollection.Add(new product { product\_Name = newPrdName, product\_Count = newPrdCountNumber, product\_Description = newPrdDesc, product\_Price = newPrdPriceNumber }); // Добавление в базу данных App.Context.products.Add(new product { product\_Name = newPrdName, product\_Count = newPrdCountNumber, product\_Description = newPrdDesc, product\_Price = newPrdPriceNumber }); // Сохранение изменений в базу данных, переоткрытие окна продукции для обновления данных из базы данных App.Context.SaveChanges(); Window productsWin = App.Current.Windows.OfType<Products>().FirstOrDefault(); productsWin.Close(); Products newPrdWin = new Products(); newPrdWin.Show(); Close(); MessageBox.Show("Product was successfully added to list!"); } } }}

“xaml.cs” код окна “redactingProduct.xaml.cs” представлен ниже:

using internetShopProject.mainEntities;using System;using System.Collections.Generic;using System.Collections.ObjectModel;using System.Linq;using System.Text;using System.Threading.Tasks;using System.Windows;using System.Windows.Controls;using System.Windows.Data;using System.Windows.Documents;using System.Windows.Input;using System.Windows.Media;using System.Windows.Media.Imaging;using System.Windows.Shapes;namespace internetShopProject{ public partial class redactingProduct : Window { public redactingProduct() { InitializeComponent(); } void textInputPrevented(object sender, TextCompositionEventArgs e) { // Проверка на написание символов (необходимый символ - цифра, если иной - не осуществляет процедуру введения желаемого символа) if (!char.IsDigit(e.Text, e.Text.Length - 1)) { e.Handled = true; } } void wrongInputPrevented(object sender, TextCompositionEventArgs e) { // Проверка на написание символов (необходимый символ - буква, если иной - не осуществляет процедуру введения желаемого символа) if (!char.IsLetter(e.Text, e.Text.Length - 1)) { e.Handled = true; } } void redactBtnClicked(object sender, RoutedEventArgs e) { // Иницилизация коллекции продуктов для последующих манипуляций ObservableCollection<product> prdCollection = App.Current.Properties["mainProductsCollection"] as ObservableCollection<product>; // Иницилизация переменных, принимающих введённые администратором значения string oldPrdName = oldProductNameTextBox.Text, newPrdName = newRedactProductNameTextBox.Text, newPrdDesc = newRedactProductDescriptionTextBox.Text, newPrdCount = newRedactProductCountTextBox.Text, newPrdPrice = newRedactProductPriceTextBox.Text; // Проверка на отсутствие значения в TextBoxes, проверка на существование пробелов в несанкционированных для этого TextBoxes if (oldPrdName == "" || newPrdName == "" || newPrdDesc == "" || newPrdCount == "" || newPrdPrice == "") { MessageBox.Show("Please fill all boxes of this form!"); } else if (oldPrdName.Contains(" ") || newPrdName.Contains(" ") || newPrdCount.Contains(" ") || newPrdPrice.Contains(" ")) { MessageBox.Show("There should be no spaces in this form beside description!"); } else if (newPrdName.Length > 50) { MessageBox.Show("Too big size of product name!"); } else { // Парсинг значения из типа string в тип int для изменения существующего продукта в коллекции и базе данных try { // Значения для проверки на исключение FormatException int checkCountNumber = int.Parse(newPrdCount); int checkPriceNumber = int.Parse(newPrdPrice); } catch (FormatException) { MessageBox.Show("Unacceptable types of data in count or price forms!"); return; } int newPrdCountNumber = int.Parse(newPrdCount); int newPrdPriceNumber = int.Parse(newPrdPrice); // Процедура поиска старого продукта по базе данных и изменение данных в базе данных var foundProduct = App.Context.products.FirstOrDefault(pn => pn.product\_Name == oldPrdName); if (foundProduct != null) { foundProduct.product\_Name = newPrdName; foundProduct.product\_Description = newPrdDesc; foundProduct.product\_Count = newPrdCountNumber; foundProduct.product\_Price = newPrdPriceNumber; App.Context.SaveChanges(); Window productsWin = App.Current.Windows.OfType<Products>().FirstOrDefault(); productsWin.Close(); Products newPrdWin = new Products(); newPrdWin.Show(); Close(); MessageBox.Show("Product was successfully redacted!"); } else { MessageBox.Show("There is no such product with name: " + oldPrdName); } } } }}

“xaml.cs” код окна “removingProduct.xaml.cs” представлен ниже:

using internetShopProject.mainEntities;using System;using System.Collections.Generic;using System.Collections.ObjectModel;using System.ComponentModel;using System.Linq;using System.Text;using System.Threading.Tasks;using System.Windows;using System.Windows.Controls;using System.Windows.Data;using System.Windows.Documents;using System.Windows.Input;using System.Windows.Media;using System.Windows.Media.Imaging;using System.Windows.Shapes;namespace internetShopProject{ public partial class removingProduct : Window { public removingProduct() { InitializeComponent(); } void deleteProductClicked(object sender, RoutedEventArgs e) { // Формирование коллекции продуктов ObservableCollection<product> prdCollectionProducts = App.Current.Properties["mainProductsCollection"] as ObservableCollection<product>; string writtenProductName = mainProductNameTextBox.Text; // Запрос на происк продукта и процедура его удаления из базы данных var foundProductName = App.Context.products.FirstOrDefault(pn => pn.product\_Name == writtenProductName); if (foundProductName != null) { App.Context.products.Remove(foundProductName); App.Context.SaveChanges(); Window productsWin = App.Current.Windows.OfType<Products>().FirstOrDefault(); productsWin.Close(); Products newPrdWin = new Products(); newPrdWin.Show(); Close(); MessageBox.Show("Product was successfully removed from database!"); } else { MessageBox.Show("Product was not found!"); } } }}

3 ТЕСТИРОВАНИЕ

В данном курсовом проекте необходимо провести тестирование общего функционала информационной системы интернет-магазина. При разработке информационной системы было использовано тестирование с помощью тест-кейсов. Тестирование главного окна с помощью тест-кейсов представлено в соответствии с таблицей 1.

Таблица 1. Тестирование главного окна с помощью тест-кейсов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Действие | Ожидаемый результат | Фактический результат | Статус |
| A1 | Нажатие на кнопку “Products” на главном окне до совершённой авторизации | Вывод окна сообщения с информацией: “Please authorize in the system to view products!” | Вывод окна сообщения с информацией: “Please authorize in the system to view products!” | Соответствует |
| A2 | Нажатие на кнопку “Shopping Cart” на главном окне до совершённой авторизации | Вывод окна сообщения с информацией: “Please authorize in the system to view shopping cart!” | Вывод окна сообщения с информацией: “Please authorize in the system to view shopping cart!” | Соответствует |
| A3 | Нажатие на кнопку “Authorization” на главном окне до совершённой авторизации | Открытие нового окна авторизации | Открытие нового окна авторизации | Соответствует |
| A4 | Нажатие на кнопку “Registration” на главном окне | Открытие нового окна регистрации | Открытие нового окна регистрации | Соответствует |

Тестирование авторизационного и регистрационного окна с помощью тест-кейсов представлено в соответствии с приложением 2.

Тестирование окна профиля с помощью тест-кейсов представлено в соответствии с таблицей 3.

Таблица 3. Тестирование окна профиля с помощью тест-кейсов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C1 | Нажатие на кнопку “Refresh” в окне профиля пользователя | Обновление логина и роли пользователя в соответствии с новыми изменёнными или имеющимися данными | Обновление логина и роли пользователя в соответствии с новыми изменёнными или имеющимися данными | Соответствует |
| C2 | Нажатие на кнопку “Sign Out” в окне профиля пользователя | Корректное удаление окна профиля и других окон, доступных лишь при авторизированном входе | Корректное удаление окна профиля и других окон, доступных лишь при авторизированном входе | Соответствует |
| C3 | Принудительный выход из окна профиля без нажатия на специфицированную для этого кнопку “Sign Out” | Корректное удаление окна профиля и других окон, доступных лишь при авторизированном входе | Корректное удаление окна профиля и других окон, доступных лишь при авторизированном входе | Соответствует |
| C4 | Нажатие на кнопку “Add Image” в окне профиля пользователя | Открытие диалогового окна для добавления необходимой картинки из директорий пользователя и корректное отображение выбранной пользователем картинки в профиле | Открытие диалогового окна для добавления необходимой картинки из директорий пользователя и корректное отображение выбранной пользователем картинки в профиле | Соответствует |
| C5 | Нажатие на кнопку “Change Image” в окне профиля пользователя | Открытие диалогового окна для изменения необходимой картинки из директорий пользователя и корректное отображение новой выбранной пользователем картинки в профиле | Открытие диалогового окна для изменения необходимой картинки из директорий пользователя и корректное отображение новой выбранной пользователем картинки в профиле | Соответствует |

Тестирование пользовательского доступа к окнам продукции и корзине продуктов с помощью тест-кейсов представлено в соответствии с таблицей 4.

Таблица 4. Тестирование пользовательского доступа к окнам продукции и корзине продуктов с помощью тест-кейсов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| D1 | Нажатие на кнопку “Products” на главном окне после успешно совершённой авторизации | Открытие нового окна продуктов | Открытие нового окна продуктов | Соответствует |
| D2 | Нажатие на кнопку “Shopping Cart” на главном окне после успешно совершённой авторизации без открытого окна продуктов | Вывод окна сообщения с информацией: “Please open “Products” window to work with a “Cart”!” | Вывод окна сообщения с информацией: “Please open “Products” window to work with a “Cart”!” | Соответствует |
| D3 | Нажатие на кнопку “Shopping Cart” на главном окне после успешно совершённой авторизации с открытым окном продуктов | Открытие нового окна корзины продуктов | Открытие нового окна корзины продуктов | Соответствует |

Тестирование окна корзины продуктов с помощью тест-кейсов представлено в соответствии с таблицей 5.

Таблица 5. Тестирование корзины продуктов с помощью тест-кейсов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| E1 | Двойное нажатие левой кнопкой мыши по одному из продуктов в окне продуктов при отсутствии открытого окна корзины продуктов | Вывод окна сообщения с информацией: “Please open “Cart” window to choose products!” | Отсутствие взаимодействия с пользователем | Не соответствует |
| E2 | Двойное нажатие левой кнопкой мыши по одному из продуктов в окне продуктов при условии открытого окна корзины продуктов | Перенос выбранного пользователем продукта из окна продуктов в окно корзины продуктов в количестве 1 штуки | Перенос выбранного пользователем продукта из окна продуктов в окно корзины продуктов в количестве 1 штуки | Соответствует |
| E3 | Нажатие на кнопку “Order” в окне корзины продуктов при условии отсутствия продуктов в корзине продуктов | Вывод окна сообщения с информацией: “You have no products to order in a “Shopping Cart”!” | Вывод окна сообщения с информацией: “You have no products to order in a “Shopping Cart”!” | Соответствует |
| E4 | Нажатие на кнопку “Order” в окне корзины продуктов при условии присутствия продуктов в корзине продуктов | Формирование чека продуктов со сбором необходимой информации для формирования чека (дата, логин покупателя, количество типов приобретённой продукции) | Формирование чека продуктов со сбором необходимой информации для формирования чека (дата, логин покупателя, количество типов приобретённой продукции) | Соответствует |

4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Разработанная информационная система интернет-магазина должна быть оснащена документационным руководством пользователя для рационального и корректного использования созданного приложения. В руководстве пользователя подробно описан весь основной функционал программы, методы работы с программой, прямое взаимодействие с информационной системой.

В руководстве описаны данные процессы: открытие окон приложения, прохождение авторизации пользователем, прохождение регистрации пользователем, управление функциями профиля пользователем (смена логина, добавление картинки, обновление добавленной картинки), разрешения пользователя на пользование некоторыми окнами после авторизации в профиль, использование функций окна продуктов, использование функций окна корзины для продуктов, функция заказа имеющихся продуктов в корзине и функция формирования продуктового чека директории документов.

Когда пользователь впервые открывает “.exe” файл приложения, он видит перед собой приветствующее окно информационной системы интернет-магазина “Main Page” – это главное окно, с которого начинается весь основной функционал программы. Начальное окно программы представлено в соответствии с рисунком 5.

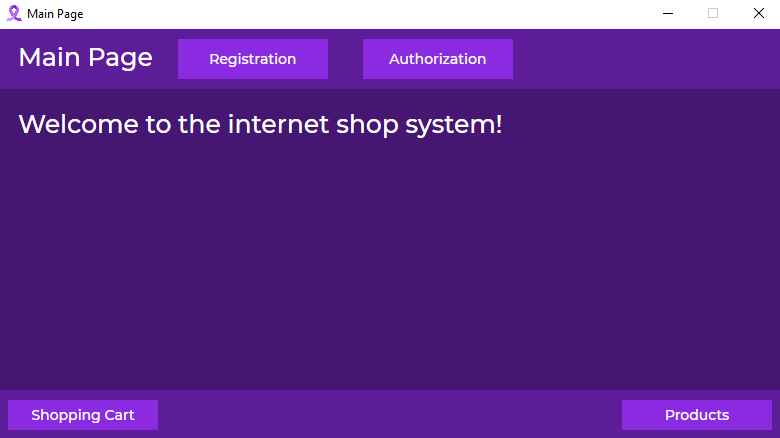


Рисунок 5. Начальное окно программы, наблюдаемое пользователем

Затем, у пользователя присутствует выбор между двумя основными функциональными частями программы – кнопкой “Registration” (регистрация) и кнопкой “Authorization” (авторизация). Выбрав кнопку “Registration”, пользователю откроется новое окно – окно регистрации. Окно регистрации представлено в соответствии с рисунком 6.

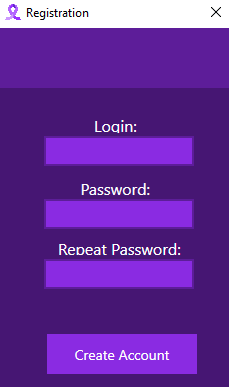


Рисунок 6. Окно регистрации

После введения соответствующих корректных данных в форму регистрации, пользователь может наблюдать новое появившееся окно – окно авторизации. В данном окне можно ввести новые данные из регистрационного окна или войти в уже существующий аккаунт, созданный пользователем ранее. Окно авторизации представлено в соответствии с рисунком 7.

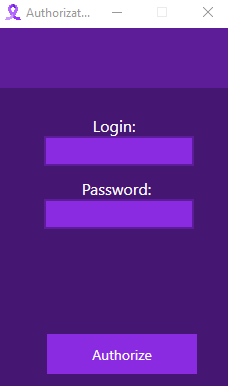


Рисунок 7. Окно авторизации

После авторизации в аккаунт, пользователю открывается новое окно – профиль. В профильном окне указаны: приветственный текст с логином пользователя, роль пользователя в “шапке” окна, кнопки “Refresh” и “Sign Out” в “шапке окна”, кнопки “Change Nickname” и “Add Image” в “подвале” окна, аватар пользователя по умолчанию, если не добавлена картинка, изменённый аватар пользователя с новой картинкой, если добавлена картинка. Кнопка “Refresh” обновляет данные внутри формы профиля, “Sign Out” – покидает профиль, выходит из аккаунта, “Change Nickname” – даёт возможность изменить логин, “Add Image” – даёт возможность добавления картинки в аватар пользователя. Окно профиля представлено в соответствии с рисунком 8.

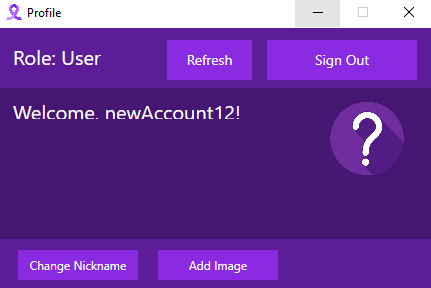


Рисунок 8. Окно профиля

При желании, пользователь может изменить логин, для этого, ему необходимо нажать на кнопку “Change Nickname”, после чего, перед пользователем приложения откроется новое окно – окно изменения логина. Окно изменения логина представлено в соответствии с рисунком 9.

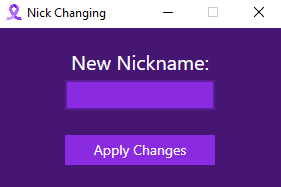


Рисунок 9. Окно изменения логина

При желании, пользователь может добавить картинку, для этого, ему необходимо нажать на кнопку “Add Image”, после чего, перед пользователем приложения откроется новое “диалоговое окно” – окно выбора картинки из директорий на компьютере пользователя. “Диалоговое окно” выбора картинки из директорий представлено в соответствии с рисунком 10.

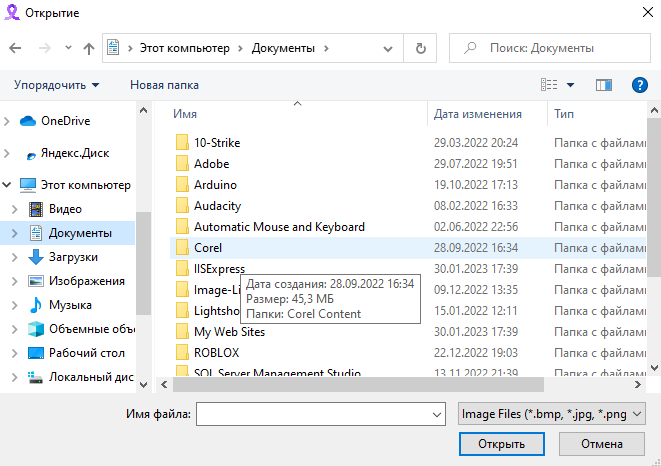


Рисунок 10. “Диалоговое окно” выбора картинки из директорий на компьютере пользователя

Также, после авторизации и открытия окна профиля, пользователь получает доступ сразу к двум кнопкам и окнам на “Main Page” (главной странице) приложения – кнопке “Products” (продукты), кнопке “Shopping Cart” (корзина продуктов). Окна, открываемые данными кнопками, представлены в соответствии с рисунками 11, 12.

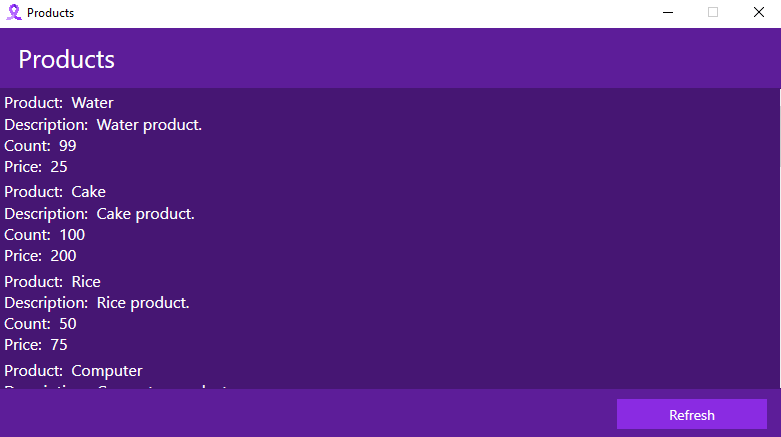


Рисунок 11. Окно продуктов

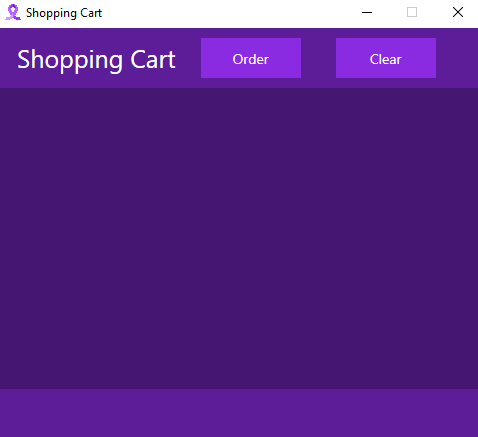


Рисунок 12. Окно корзины продуктов

Пользователь может свободно манипулировать перемещениями продуктов из окна продуктов в корзину продуктов и, наоборот, – из корзины продуктов в окно продуктов (перемещение продуктов туда и обратно происходит посредством двойного нажатия левой кнопкой мыши по конкретному продукту). Итоговый результат перемещения продукта из окна продуктов в корзину продуктов представлен в соответствии с рисунком 13.

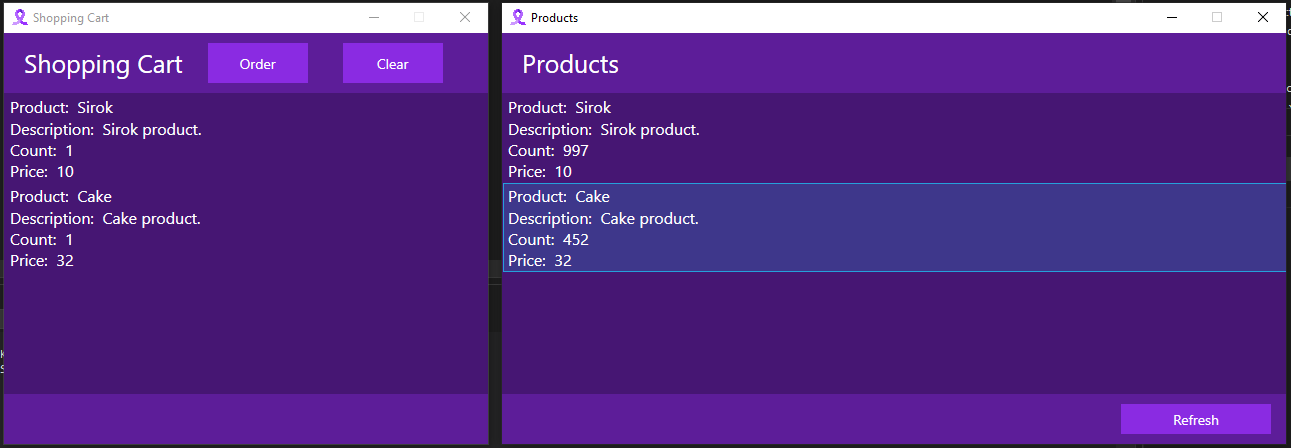
****

Рисунок 13. Итоговый результат перемещения продукта из окна продуктов в корзину продуктов

Пользователь может оформить продуктовый чек, нажав на кнопку “Order” (заказать), если в его корзине продуктов будут присутствовать продукты (в ином случае, продуктовый чек не может быть сформирован). Сформированный продуктовый чек представлен в соответствии с рисунком 14.

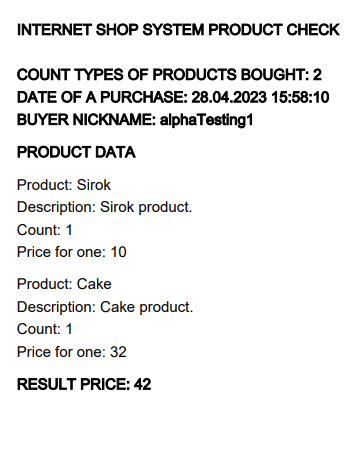
****

Рисунок 14. Сформированный продуктовый чек

ЗАКЛЮЧЕНИЕ

В ходе работы над курсовым проектом была разработана информационная система интернет-магазина. При разработке проекта и самой информационной системы были приобретены и продемонстрированы умения работы со средой разработки Microsoft SQL Server Management Studio (SSMS), интегрированной средой разработки Microsoft Visual Studio (версии 17.4.4), технологией .Net Framework (версии 4.7.2), графической подсистемой WPF (Windows Presentation Foundation), языком программирования C# и набором технологий Entity Framework.

Произведя выбор среды разработки и осуществив реализацию необходимых планов по разработке информационной системы интернет-магазина, требования, поставленные в главе 1 (Аналитический обзор) пункте 1.1 (Описание требований), были успешно реализованы в данном курсовом проекте, а также, цели и задачи проекта, описанные в введении, были достигнуты.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Visual Studio: IDE и редактор кода для разработчиков и групп, работающих с программным обеспечением : официальный сайт. – URL: https://visualstudio.microsoft.com/ru/ (дата обращения: 04.04.2023). – Текст : электронный.

2. C# | Modern, open-source programming language for .NET : официальный сайт. – URL: https://dotnet.microsoft.com/en-us/languages/csharp (дата обращения: 03.04.2023). – Текст : электронный.

3. WPF и C# | Полное руководство : официальный сайт. – URL: https://metanit.com/sharp/wpf/ (дата обращения: 04.04.2023). – Текст : электронный.

4. Entity Framework – Краткое руководство – CoderLessons.com : официальный сайт. – URL: https://coderlessons.com/tutorials/microsoft-technologies/izuchite-entity-framework/entity-framework-kratkoe-rukovodstvo (дата обращения: 04.04.2023). – Текст : электронный.

5. Связи между таблицами базы данных / Хабр : официальный сайт. – URL: https://habr.com/ru/articles/488054/ (дата обращения: 03.04.2023). – Текст : электронный.

6. Работа с Visual Studio : официальный сайт. – URL: https://professorweb.ru/my/programs/visual-studio/level1/ (дата обращения: 04.04.2023). – Текст : электронный.

ПРИЛОЖЕНИЕ 1

ОПИСАНИЕ ТРЕБОВАНИЙ

1. Бизнес-требования:

БТ-1:

Необходимо обеспечить удобство работы с базой данных для сотрудников информационной системы интернет-магазина.

БТ-2:

Необходимо обеспечить автоматизацию таких функций как:

* Общая обработка процессов манипуляции с продукцией;
* Формирование продуктового чека по заказам пользователей;
* Регулирование специализированных процессов удаления, добавления, редактирования продукции, администраторами предприятия.

БТ-3:

Общая цветовая гамма приложения состоит из фиолетовых оттенков. Цветовая гамма приложения представлена в соответствии с данными параметрами цветовой модели: внутренняя контентная часть (“#461673”), “шапка” (“#5d1d99”), “подвал” (“#5d1d99”), кнопки (“BlueViolet”), поля для заполнения (“BlueViolet”).

1. Пользовательские требования:

ПТ-1:

С помощью системы, пользователи смогут выполнять следующие задачи:

* Изменять аватар в профиле после процесса авторизации путём нажатия на кнопку “Change Image”;
* Добавлять картинку в аватар в профиле после процесса авторизации путём нажатия на кнопку “Add Image”;
* Изменять логин в профиле после процесса авторизации путём нажатия на кнопку “Change Nickname”;
* Свободно оперировать перемещением продукции из продуктов в корзину продуктов и наоборот путём двойного нажатия на левую кнопку мыши по выбранному товару;
* Свободно оперировать удалением продукции из корзины продуктов путём нажатия на кнопку “Clear” для очистки всей корзины и путём двойного нажатия на левую кнопку мыши для удаление единицы выбранного товара;
* Получать необходимые сведения о продукции из базы данных;
* Формировать итоговый продуктовый чек, заказывая выбранную продукцию, находящуюся в корзине продуктов.

ПТ-2:

С помощью системы, администраторы смогут выполнять следующие задачи:

* Создавать продукт в информационной системе с занесением продукта в базу данных приложения;
* Удалять продукт в информационной системе с удалением продукта из базы данных приложения;
* Редактировать продукт в информационной системе с занесением отредактированных изменений в базу данных приложения.

1. Бизнес-правила:

БП-1:

* Удаление данных может производить только администратор системы, отвечающий за данный процесс.

БП-2:

* Внесение и изменение информации производится только администратором системы, отвечающим за данные процессы.

1. Функциональные требования:

ФТ-1:

Система должна автоматически выполнять резервное копирование данных ежедневно в указанный момент времени.

1. Нефункциональные требования:

НТ-1:

Программное обеспечение должно быть защищено системой авторизации с 128-битным алгоритмом хеширования “MD5”. Параметры входа администраторам системы выдаются главным руководителем проекта;

1. Ограничения:

О-1:

Приложение должно поддерживаться ОС семейства Microsoft Windows от 7 версии и до 10 включительно.

1. Требования к интерфейсам:

Интерфейс программы должен быть понятным и простым в освоении, а, также, быть логичным и удобным.

1. Требования к данным:

Все данные системы должны храниться в БД под управлением СУБД Microsoft SQL.

1. Требования к производительности:

Система должна обеспечивать высокую скорость работы и быстрый доступ к базе данных, в том числе при большом количестве пользователей и продукции.

1. Требования к безопасности:

Система должна обеспечивать защиту данных и конфиденциальность пользовательской информации.

1. Требования к поддержке и технической поддержке:

Система должна иметь инструкции по пользовательской эксплуатации.

1. Требования к масштабируемости:

Система должна быть способна масштабироваться при увеличении количества товаров и пользователей, без значительного увеличения времени обработки данных и потери производительности системы.

1. Требования к обучаемости:

Система должна быть простой и удобной в освоении для новых пользователей.

ПРИЛОЖЕНИЕ 2

ТЕСТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ С ПОМОЩЬЮ ТЕСТ-КЕЙСОВ

Таблица 2. Тестирование авторизационного и регистрационного окна с помощью тест-кейсов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B1 | Нажатие на кнопку “Authorize” в окне авторизации без занесения данных в форму авторизации | Вывод окна сообщения с информацией: “Fill the empty boxes!” | Вывод окна сообщения с информацией: “Invalid login!” | Не соответствует |
| B2 | Нажатие на кнопку “Create Account” в окне регистрации без занесения данных в форму регистрации | Вывод окна сообщения с информацией: “Fill the empty boxes!” | Вывод окна сообщения с информацией: “Fill the empty boxes!” | Соответствует |
| B3 | Ввод некорректных данных в оба поля авторизации с последующим нажатием на кнопку “Authorize” | Вывод окна сообщения с информацией: “Invalid login!” | Вывод окна сообщения с информацией: “Invalid login!” | Соответствует |
| B4 | Ввод корректного логина и некорректного пароля в поля авторизации с последующим нажатием на кнопку “Authorize” | Вывод окна сообщения с информацией: “Invalid password!” | Вывод окна сообщения с информацией: “Invalid password!” | Соответствует |
| B5 | Ввод корректного логина и корректного пароля в поля авторизации с последующим нажатием на кнопку “Authorize” | Вывод окна сообщения с информацией: “Valid data!” и последующее открытие нового окна профиля | Вывод окна сообщения с информацией: “Valid data!” и последующее открытие нового окна профиля | Соответствует |
| B6 | Ввод логина длиной менее 8 символов в поле логина в окне регистрации с последующим нажатием на кнопку “Create Account” | Вывод окна сообщения с информацией: “Login and password must be minimum 8 symbols!” | Вывод окна сообщения с информацией: “Login and password must be minimum 8 symbols!” | Соответствует |
| B7 | Ввод пароля длиной менее 8 символов в поле пароля в окне регистрации с последующим нажатием на кнопку “Create Account” | Вывод окна сообщения с информацией: “Login and password must be minimum 8 symbols!” | Вывод окна сообщения с информацией: “Login and password must be minimum 8 symbols!” | Соответствует |
| B8 | Ввод логина длиной более 20 символов в поле логина в окне регистрации с последующим нажатием на кнопку “Create Account” | Вывод окна сообщения с информацией: “Login and password must be maximum 20 symbols!” | Вывод окна сообщения с информацией: “Login and password must be maximum 20 symbols!” | Соответствует |
| B9 | Ввод пароля длиной более 20 символов в поле логина в окне регистрации с последующим нажатием на кнопку “Create Account” | Вывод окна сообщения с информацией: “Login and password must be maximum 20 symbols!” | Вывод окна сообщения с информацией: “Login and password must be maximum 20 symbols!” | Соответствует |
| B10 | Ввод пароля корректной длины, без заглавных букв в поле пароля в окне регистрации с последующим нажатием на кнопку “Create Account” | Вывод окна сообщения с информацией: “There are must be atleast one “Upper Letter” in password!” | Вывод окна сообщения с информацией: “There are must be atleast one “Upper Letter” in password!” | Соответствует |
| B11 | Ввод пароля корректной длины, со всеми заглавными буквами, без строчных букв, в поле пароля в окне регистрации с последующим нажатием на кнопку “Create Account” | Вывод окна сообщения с информацией: “There are must be atleast one “Lower Letter” in password!” | Вывод окна сообщения с информацией: “There are must be atleast one “Lower Letter” in password!” | Соответствует |
| B12 | Ввод логина корректной длины с пробелом в поле логина в окне регистрации с последующим нажатием на кнопку “Create Account” | Вывод окна сообщения с информацией: “Login and password must be wrote without spaces!” | Вывод окна сообщения с информацией: “Login and password must be wrote without spaces!” | Соответствует |
| B13 | Ввод пароля корректной длины, с заглавной буквой, со строчными буквами, с пробелом, в поле пароля в окне регистрации с последующим нажатием на кнопку “Create Account” | Вывод окна сообщения с информацией: “Login and password must be wrote without spaces!” | Вывод окна сообщения с информацией: “Login and password must be wrote without spaces!” | Соответствует |
| B14 | Ввод двух отличных друг от друга паролей в поля пароля и повтор пароля в окне регистрации с последующим нажатием на кнопку “Create Account” | Вывод окна сообщения с информацией: “Passwords must be same!” | Вывод окна сообщения с информацией: “Passwords must be same!” | Соответствует |
| B15 | Ввод пароля корректной длины, с заглавной буквой, со строчными буквами, без пробела, без цифр, в поле пароля в окне регистрации с последующим нажатием на кнопку “Create Account” | Вывод окна сообщения с информацией: “ There are must be atleast one “Digit” in password!” | Вывод окна сообщения с информацией: “ There are must be atleast one “Digit” in password!” | Соответствует |
| B16 | Ввод пароля корректной длины, с заглавной буквой, со строчными буквами, без пробела, с цифрой, без специального символа, в поле пароля в окне регистрации с последующим нажатием на кнопку “Create Account” | Вывод окна сообщения с информацией: “ There are must be atleast one “Special Letter” in password!” | Вывод окна сообщения с информацией: “ There are must be atleast one “Special Letter” in password!” | Соответствует |
| B17 | Ввод корректного логина, пароля, повторного пароля в окне регистрации с последующим нажатием на кнопку “Create Account” | Вывод окна сообщения с информацией: “Account was successfully registered!” и открытие нового окна авторизации | Вывод окна сообщения с информацией: “Account was successfully registered!” и открытие нового окна авторизации | Соответствует |