федеральное государственное бюджетное образовательное учреждение высшегообразования

«**Вологодский государственный университет**»

университетский колледж

Информационные системы и программирование

(наименование специальности)

**ОТЧЁТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

Дисциплина: «Инструментальные средства разработки программного обеспечения»

Руководитель Шатов Егор Эдуардович\_\_\_\_\_\_\_

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

Выполнил (а) обучающийся Коновалов Артём Николаевич \_

(Ф.И.О)

Группа, курс ИСП 41-19\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вологда

2022 г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc122636778)

[1 БЕЗОПАСНОСТЬ ТРУДА. ЗНАКОМСТВО С ПРЕДПРИЯТИЕМ. ЗАКРЕПЛЕНИЕ РАБОЧЕГО МЕСТА 4](#_Toc122636779)

[2 ОПИСАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОРГАНИЗАЦИИ ВРОО «СОЗИДАНИЕ» 5](#_Toc122636780)

[3 РАЗРАБОТКА UML ДИАГРАММЫ 6](#_Toc122636781)

[4 РАЗРАБОТКА БАЗЫ ДАННЫХ ДЛЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОРГАНИЗАЦИИ ВРОО «СОЗИДАНИЕ» 7](#_Toc122636782)

[5 РАЗРАБОТКА ПРИЛОЖЕНИЯ 8](#_Toc122636783)

[6 РАЗМЕЩЕНИЕ РАЗРАБОТАННОГО ПРИЛОЖЕНИЯ В РЕПОЗИТОРИЙ GITHUB. 27](#_Toc122636784)

[7 ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ 28](#_Toc122636785)

[ЗАКЛЮЧЕНИЕ 48](#_Toc122636786)

ВВЕДЕНИЕ

Для прохождение производственной практики была выбрана ВРОО «Созидание», так как она входит в ИТ сферу.

Компания занимается цифровым волонтёрством.

1БЕЗОПАСНОСТЬ ТРУДА. ЗНАКОМСТВО С ПРЕДПРИЯТИЕМ. ЗАКРЕПЛЕНИЕ РАБОЧЕГО МЕСТА

В начале производственной практики был проведен инструктаж безопасности труда, в котором были разъяснены такие вещи как проверка подключения проводов к электросети, подготовка рабочего места, правила эксплуатации техники.

Технические характеристики рабочего компьютера:

* ОС: Windows 10 64-bit;
* Имя устройства DESKTOP-NCDLCE1;
* Процессор Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz;
* Оперативная память 12,0 ГБ (доступно: 11,8 ГБ);
* Код устройства 8E6ADD8C-DD3C-4932-A0B8-AEDA575039AF;
* Код продукта 00331-10000-00001-AA255;
* Тип системы 64-разрядная операционная система, процессор x64;

2 ОПИСАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОРГАНИЗАЦИИ ВРОО «СОЗИДАНИЕ»

У компании ВРОО «Созидание» отсутствует единая система хранения всех данных по оборудованию, однако имеется Яндекс диск, на котором хранится множество файлов по каждому оборудованию.

Для упрощения работы с документами и списками была создана система «RestComp», которая хранит в себе список всего оборудования и периферии, а также подробное описание и их количество.

Информационная система для предприятия была разработана тремя практикантами: Коноваловым Артёмом, Перемышленниковым Романом и Лавренчуком Ильёй.

3 РАЗРАБОТКА UMLДИАГРАММЫ

Перед разработкой продукта была создана UMLдиаграмма, демонстрирующая работу приложения.Данная UML диаграмма представлена в соответствии с рисунком 1.

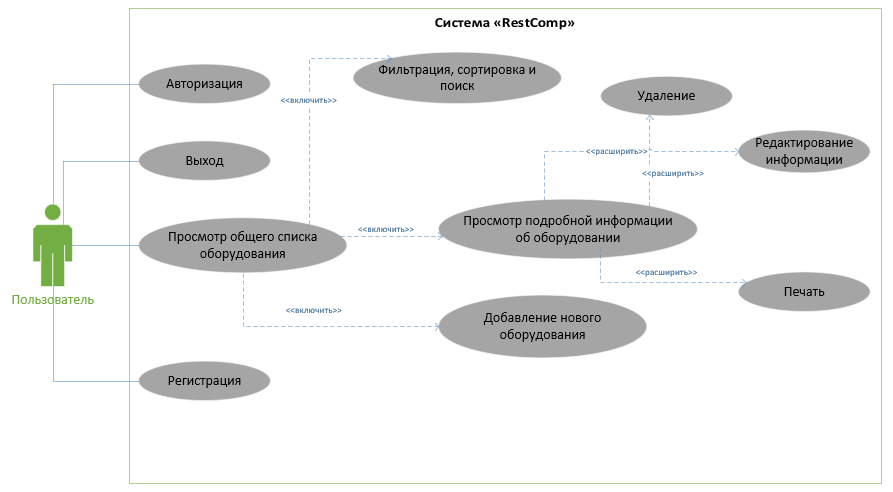


Рисунок 1 – UML диаграмма

4 РАЗРАБОТКА БАЗЫ ДАННЫХ ДЛЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОРГАНИЗАЦИИ ВРОО«СОЗИДАНИЕ»

Диаграмма базы данных представлена в соответствии с рисунком 2.

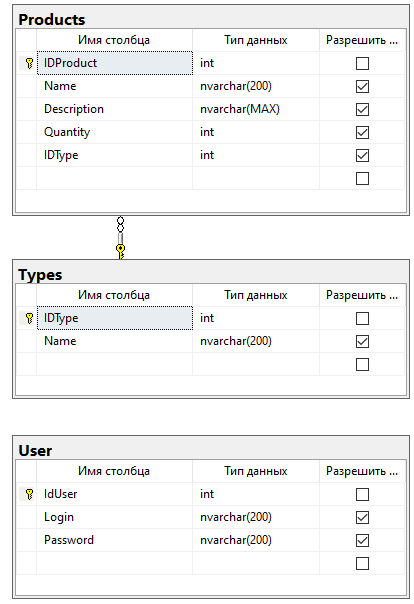


Рисунок 2 – Диаграмма базы данных

5 РАЗРАБОТКА ПРИЛОЖЕНИЯ

Страница авторизации представлена в соответствии с рисунком3

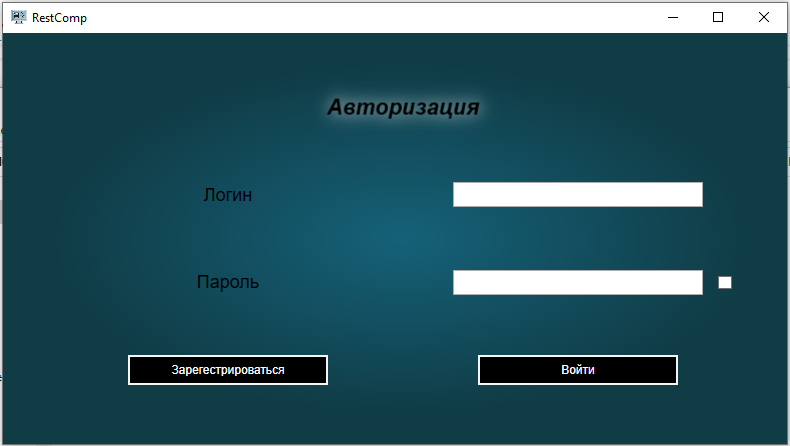


Рисунок 3 – Страница авторизации

Кодверсткистраницыавторизации:

*<Page x:Class="Practice\_Production.Pages.Autorisation"*

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

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

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

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

*xmlns:local="clr-namespace:Practice\_Production.Pages"*

*mc:Ignorable="d"*

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

*Title="Autorisation" Background="#FF103C46">*

*<StackPanelHorizontalAlignment="Center" VerticalAlignment="Center">*

*<Grid Margin="50" MinHeight="350" MinWidth="700" MaxWidth="700" MaxHeight="350">*

*<Grid.Background>*

*<RadialGradientBrush>*

*<GradientStop Color="#FF156178" Offset="0"/>*

*<GradientStop Color="#FF103C46" Offset="1"/>*

*</RadialGradientBrush>*

*</Grid.Background>*

*<Grid.ColumnDefinitions>*

*<ColumnDefinition Width="\*"/>*

*<ColumnDefinition Width="\*"/>*

*</Grid.ColumnDefinitions>*

*<Grid.RowDefinitions>*

*<RowDefinition Height="\*"/>*

*<RowDefinition Height="\*"/>*

*<RowDefinition Height="\*"/>*

*<RowDefinition Height="\*"/>*

*</Grid.RowDefinitions>*

*<Label Content="Авторизация" Style="{StaticResourceLabelTitle}" Grid.Row="0" Grid.ColumnSpan="2">*

*<Label.BitmapEffect>*

*<DropShadowBitmapEffect Color="#FFFDFDFD" Direction="1" ShadowDepth="1" Opacity=".9" Softness="1" />*

*</Label.BitmapEffect>*

*</Label>*

*<Label Content="Логин" Style="{StaticResourceLabelAvtorize}" Grid.Column="0" Grid.Row="1"/>*

*<TextBoxx:Name="Login" Style="{StaticResourceTBoxClassic}" Grid.Column="1" Grid.Row="1" TextAlignment="Left" />*

*<Label Content="Пароль" Style="{StaticResourceLabelAvtorize}" Grid.Column="0" Grid.Row="2"/>*

*<CheckBox Name="CheckPassword" Grid.Column="1" Grid.Row="2" Click="CheckPassword\_Checked"*

*HorizontalAlignment="Right" VerticalAlignment="Center" Margin="0,0,20,0"/>*

*<PasswordBox Name="PasswordHiden" Grid.Column="1" Grid.Row="2"/>*

*<TextBoxx:Name="Password" Style="{StaticResourceTBoxClassic}" Grid.Column="1" Grid.Row="2" Visibility="Hidden" IsReadOnly="True" TextAlignment="Left"/>*

*<Button x:Name="Enter" Click="Enter\_Click" Content="Войти" Grid.Column="1" Grid.Row="3" Style="{StaticResource Buttons}"/>*

*<Button x:Name="Registration" Click="Registration\_Click" Content="Зарегестрироваться" Grid.Column="0" Grid.Row="3" Style="{StaticResource Buttons}"/>*

*</Grid>*

*</StackPanel>*

*</Page>*

Исходный код страницы авторизации:

*using System;*

*using System.Linq;*

*using System.Windows;*

*using System.Windows.Controls;*

*using System.Windows.Navigation;*

*using ClassLibraryForPracticeProduction;*

*namespace Practice\_Production.Pages*

*{*

*public partial class Autorisation : Page*

*{*

*public string PasswordString = "";*

*public Autorisation()*

*{*

*InitializeComponent();*

*}*

*public void Enter\_Click(object sender, RoutedEventArgs e)*

*{*

*try*

*{*

*string InputPassword = ClassConvertToMD5.ConvertToMD5(PasswordHiden.Password);*

*var auto = App.Context.Users.Where(p =>p.Login == Login.Text&&p.Password == InputPassword).FirstOrDefault();*

*if (Login.Text.Length == 0 || PasswordHiden.Password.Length == 0)*

*{*

*MessageBox.Show("Вы не заполнили все поля.",*

*"Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);*

*return;*

*}*

*else if (auto == null)*

*{*

*MessageBox.Show("Такого пользователя нет. Пожалуйста, проверьте правильность ввода данных аккаунта, или зарегестрируйтесь.",*

*"Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);*

*return;*

*}*

*else*

*{*

*MessageBox.Show("Вызашликак: " + auto.Login,"Успех",MessageBoxButton.OK,MessageBoxImage.Information);*

*NavigationService.Navigate(new Pages.PageStore());*

*}*

*}*

*catch (Exception ex)*

*{*

*MessageBox.Show("Ошибка" + ex, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);*

*}*

*}*

*public void Registration\_Click(object sender, RoutedEventArgs e)*

*{*

*NavigationService.Navigate(new Pages.Registration());*

*}*

*private void CheckPassword\_Checked(object sender, RoutedEventArgs e)*

*{*

*if (CheckPassword.IsChecked==true)*

*{*

*PasswordHiden.Visibility = Visibility.Hidden;*

*PasswordString = PasswordHiden.Password;*

*Password.Text = PasswordString;*

*Password.Visibility = Visibility.Visible;*

*}*

*else*

*{*

*Password.Visibility = Visibility.Hidden;*

*PasswordHiden.Visibility = Visibility.Visible;*

*}*

*}*

*}*

*}*

Страница регистрации представлена в соответствии с рисунком 4.

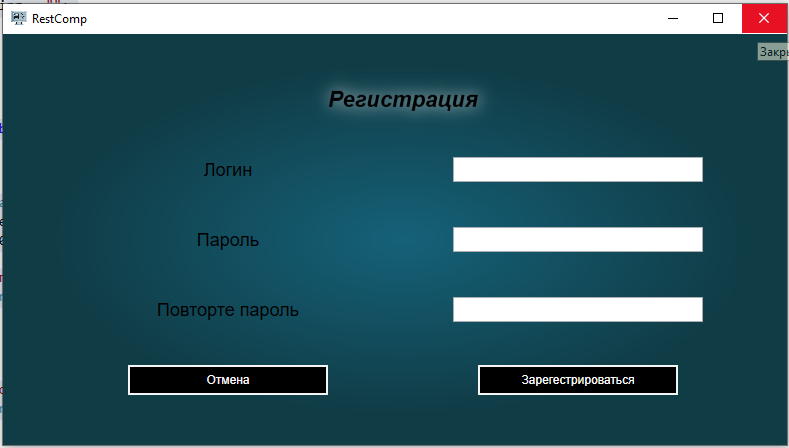


Рисунок 4 – Страница регистрации

Кодверсткистраницырегистрации:

*<Page x:Class="Practice\_Production.Pages.Registration"*

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

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

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

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

*xmlns:local="clr-namespace:Practice\_Production.Pages"*

*mc:Ignorable="d"*

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

*Title="Registration" Background="#FF103C46">*

*<StackPanelHorizontalAlignment="Center" VerticalAlignment="Center">*

*<Grid Margin="50" MinHeight="350" MinWidth="700" MaxWidth="700" MaxHeight="350">*

*<Grid.Background>*

*<RadialGradientBrush>*

*<GradientStop Color="#FF156178" Offset="0"/>*

*<GradientStop Color="#FF103C46" Offset="1"/>*

*</RadialGradientBrush>*

*</Grid.Background>*

*<Grid.ColumnDefinitions>*

*<ColumnDefinition Width="\*"/>*

*<ColumnDefinition Width="\*"/>*

*</Grid.ColumnDefinitions>*

*<Grid.RowDefinitions>*

*<RowDefinition Height="\*"/>*

*<RowDefinition Height="\*"/>*

*<RowDefinition Height="\*"/>*

*<RowDefinition Height="\*"/>*

*<RowDefinition Height="\*"/>*

*</Grid.RowDefinitions>*

*<Label Content="Регистрация" Style="{StaticResourceLabelTitle}" Grid.Row="0" Grid.ColumnSpan="2">*

*<Label.BitmapEffect>*

*<DropShadowBitmapEffect Color="#FFFDFDFD" Direction="1" ShadowDepth="1" Opacity=".9" Softness="1" />*

*</Label.BitmapEffect>*

*</Label>*

*<Label Content="Логин" Style="{StaticResourceLabelAvtorize}" Grid.Row="1" Grid.Column="0"/>*

*<TextBoxx:Name="Login" Grid.Row="1" Grid.Column="1" Style="{StaticResourceTBoxClassic}"/>*

*<Label Content="Пароль" Style="{StaticResourceLabelAvtorize}" Grid.Row="2" Grid.Column="0"/>*

*<TextBoxx:Name="Password" Grid.Row="2" Grid.Column="1" Style="{StaticResourceTBoxClassic}"/>*

*<Label Content="Повтортепароль" Style="{StaticResourceLabelAvtorize}" Grid.Row="3" Grid.Column="0"/>*

*<TextBoxx:Name="CopyPassword" Grid.Row="3" Grid.Column="1" Style="{StaticResourceTBoxClassic}"/>*

*<Button x:Name="Registr" Click="Registr\_Click" Content="Зарегестрироваться" Grid.Row="4" Grid.Column="1" Style="{StaticResource Buttons}"/>*

*<Button x:Name="Cancel" Click="Cancel\_Click" Content="Отмена" Grid.Row="4" Grid.Column="0" Style="{StaticResource Buttons}"/>*

*</Grid>*

*</StackPanel>*

*</Page>*

Исходный код страницы регистрации:

*using System;*

*using System.Linq;*

*using System.Windows;*

*using System.Windows.Controls;*

*using System.Windows.Navigation;*

*using ClassLibraryForPracticeProduction;*

*namespace Practice\_Production.Pages*

*{*

*public partial class Registration : Page*

*{*

*public Registration()*

*{*

*InitializeComponent();*

*}*

*public void Registr\_Click(object sender, RoutedEventArgs e)*

*{*

*try*

*{*

*var errorMessage = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login.Text, Password.Text, CopyPassword.Text);*

*//проверканаошибки*

*if (errorMessage.Length> 0)*

*{*

*MessageBox.Show(errorMessage, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);*

*return;*

*}*

*//проверка на наличие логина в базе*

*if (App.Context.Users.Count(p =>p.Login == Login.Text) > 0)*

*{*

*MessageBox.Show("Данный логин уже существует.");*

*return;*

*}*

*string result = ClassConvertToMD5.ConvertToMD5(Password.Text);*

*Entity.User user = new Entity.User()*

*{*

*Login = Login.Text,*

*Password = result*

*};*

*App.Context.Users.Add(user);*

*App.Context.SaveChanges();*

*MessageBox.Show("Пользовательуспешнозарегестрирован.");*

*NavigationService.GoBack();*

*}*

*catch (Exception ex)*

*{*

*MessageBox.Show("Ошибка" + ex);*

*}*

*}*

*public void Cancel\_Click(object sender, RoutedEventArgs e)*

*{*

*NavigationService.GoBack();*

*}*

*}*

*}*

Главная страница пользователя представлена в соответствии с рисунком 5.

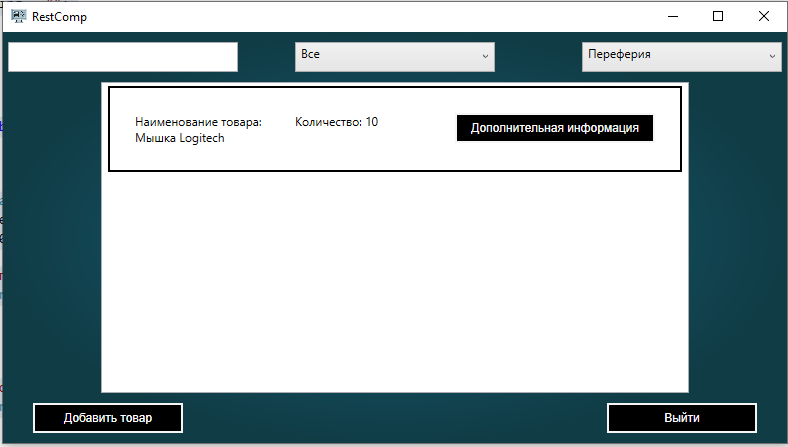


Рисунок 5 – Главная страница пользователя

Кодверсткиглавнойстраницы:

*<Page x:Class="Practice\_Production.Pages.PageStore"*

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

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

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

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

*xmlns:local="clr-namespace:Practice\_Production.Pages"*

*mc:Ignorable="d"*

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

*Title="SpisokOborudovania">*

*<Grid>*

*<Grid.Background>*

*<RadialGradientBrush>*

*<GradientStop Color="#FF156178" Offset="0"/>*

*<GradientStop Color="#FF103C46" Offset="1"/>*

*</RadialGradientBrush>*

*</Grid.Background>*

*<Grid.RowDefinitions>*

*<RowDefinition Height="50"/>*

*<RowDefinition Height="\*"/>*

*<RowDefinition Height="50"/>*

*</Grid.RowDefinitions>*

*<!--Поискоборудования -->*

*<TextBoxGrid.Row="0" Margin="5" Width="230" Height="30" Name="Search" HorizontalAlignment="Left" TextChanged="Search\_TextChanged"/>*

*<!--Сортировка -->*

*<ComboBoxGrid.Row="0" SelectionChanged="ComboSort\_SelectionChanged" Name="ComboSort" HorizontalAlignment="Center">*

*<ComboBoxItem Content="Все"/>*

*<ComboBoxItem Content="Повозрастанию"/>*

*<ComboBoxItem Content="Поубыванию"/>*

*</ComboBox>*

*<!--Фильтрация -->*

*<ComboBoxGrid.Row="0" HorizontalAlignment="Right" Name="ComboFiltr" SelectionChanged="ComboFiltr\_SelectionChanged"/>*

*<ListViewGrid.Row="1" Name="ListProduct" HorizontalAlignment="Center">*

*<ListView.ItemTemplate>*

*<DataTemplate>*

*<Border BorderBrush="Black" BorderThickness="2" Background="{Binding BgColor}">*

*<StackPanel Margin="20" Orientation="Horizontal" HorizontalAlignment="Center" VerticalAlignment="Center">*

*<Grid>*

*<Grid.ColumnDefinitions>*

*<ColumnDefinition Width="\*"/>*

*<ColumnDefinition Width="\*"/>*

*<ColumnDefinition Width="\*"/>*

*</Grid.ColumnDefinitions>*

*</Grid>*

*<StackPanelGrid.Column="0" Margin="5">*

*<TextBlock Text="{Binding Names}" Style="{StaticResourceTBlockInSpisok}"/>*

*</StackPanel>*

*<StackPanelGrid.Column="1" Margin="5">*

*<TextBlock Text="{Binding Quantities}" Style="{StaticResourceTBlockInSpisok}"/>*

*</StackPanel>*

*<StackPanelGrid.Column="2" Margin="5">*

*<Button Name="Edit" Content="Дополнительнаяинформация" Click="Edit\_Click" Style="{StaticResource Buttons}"/>*

*</StackPanel>*

*</StackPanel>*

*</Border>*

*</DataTemplate>*

*</ListView.ItemTemplate>*

*</ListView>*

*<Button x:Name="Add" Click="Add\_Click" Grid.Row="2" Width="150" Height="30" Content="Добавитьтовар" Style="{StaticResource Buttons}"*

*HorizontalAlignment="Left" VerticalAlignment="Center" Margin="30,0,0,0"/>*

*<Button x:Name="Exit" Click="Exit\_Click" Grid.Row="2" Width="150" Height="30" Content="Выйти" Style="{StaticResource Buttons}"*

*HorizontalAlignment="Right" VerticalAlignment="Center" Margin="0,0,30,0"/>*

*</Grid>*

*</Page>*

Исходный код главной страницы:

*using System;*

*using System.Linq;*

*using System.Windows;*

*using System.Windows.Controls;*

*using System.Windows.Navigation;*

*namespace Practice\_Production.Pages*

*{*

*/// <summary>*

*/// ЛогикавзаимодействиядляPageStore.xaml*

*/// </summary>*

*public partial class PageStore : Page*

*{*

*public PageStore()*

*{*

*InitializeComponent();*

*ComboFiltr.ItemsSource = App.Context.Types.Select(p =>p.Name).ToList();*

*ComboFiltr.SelectedIndex = 0;*

*ComboSort.SelectedIndex = 0;*

*}*

*public void Update()*

*{*

*try*

*{*

*var render = App.Context.Products.ToList();*

*//Поиск*

*render = render.Where(p =>p.Name.ToLower().Contains(Search.Text.ToLower())).ToList();*

*//Сортировка*

*if (ComboSort.SelectedIndex == 0)*

*render = render.DefaultIfEmpty().ToList();*

*if (ComboSort.SelectedIndex == 1)*

*render = render.OrderBy(p =>p.Quantity).ToList();*

*if (ComboSort.SelectedIndex == 2)*

*render = render.OrderByDescending(p =>p.Quantity).ToList();*

*//Фильтрация*

*if (render.Where(p =>p.Type.Name.ToLower() == ComboFiltr.SelectedItem.ToString().ToLower()).ToList() == null)*

*{*

*MessageBox.Show("Данноготоваранет", "Error", MessageBoxButton.OK, MessageBoxImage.Error);*

*return;*

*}*

*else*

*render = render.Where(p =>p.Type.Name.ToLower() == ComboFiltr.SelectedItem.ToString().ToLower()).ToList();*

*ListProduct.ItemsSource = render;*

*}*

*catch (Exception e)*

*{ MessageBox.Show(e.Message, "Error", MessageBoxButton.OK, MessageBoxImage.Error); }*

*}*

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

*{*

*NavigationService.Navigate(new Pages.AddEditProduct());*

*Update();*

*}*

*private void Edit\_Click(object sender, RoutedEventArgs e)*

*{*

*var edit = (sender as Button).DataContext as Entity.Product;*

*NavigationService.Navigate(new Pages.AddEditProduct(edit));*

*}*

*private void ComboFiltr\_SelectionChanged(object sender, SelectionChangedEventArgs e)*

*{*

*Update();*

*}*

*private void ComboSort\_SelectionChanged(object sender, SelectionChangedEventArgs e)*

*{*

*Update();*

*}*

*private void Search\_TextChanged(object sender, TextChangedEventArgs e)*

*{*

*Update();*

*}*

*private void Exit\_Click(object sender, RoutedEventArgs e)*

*{*

*if (MessageBox.Show($"Выдействительнохотитевыйти?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)*

*{*

*NavigationService.Navigate(new Pages.Autorisation());*

*}*

*}*

*}*

*}*

Страница добавления и редактирования представлена в соответствии с рисунком 6.

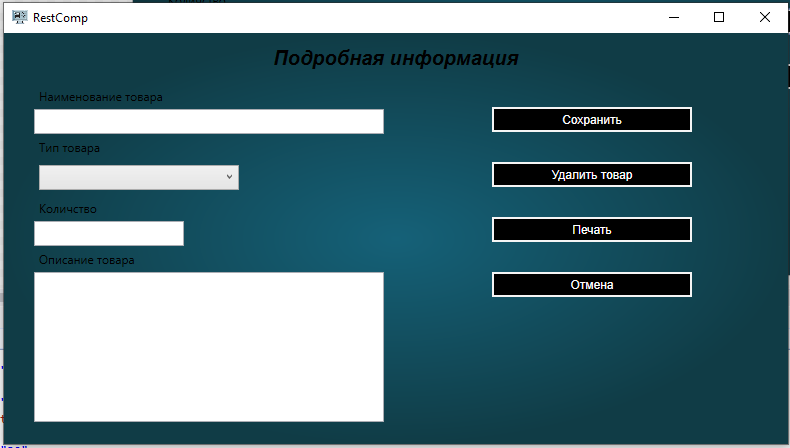


Рисунок 6 – Страница добавления и редактирования

Код верстки страницы добавления и редактирования:

*<Page x:Class="Practice\_Production.Pages.AddEditProduct"*

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

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

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

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

*xmlns:local="clr-namespace:Practice\_Production.Pages"*

*mc:Ignorable="d"*

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

*Title="AddEditProduct">*

*<Grid>*

*<Grid.Background>*

*<RadialGradientBrush>*

*<GradientStop Color="#FF156178" Offset="0"/>*

*<GradientStop Color="#FF103C46" Offset="1"/>*

*</RadialGradientBrush>*

*</Grid.Background>*

*<Grid.RowDefinitions>*

*<RowDefinition Height="50"/>*

*<RowDefinition Height="\*"/>*

*</Grid.RowDefinitions>*

*<Grid.ColumnDefinitions>*

*<ColumnDefinition Width="\*"/>*

*<ColumnDefinition Width="\*"/>*

*</Grid.ColumnDefinitions>*

*<Label Content="Подробнаяинформация" Grid.Row="0" Grid.ColumnSpan="2" FontSize="20"*

*FontFamily="Arial" FontStyle="Italic" FontWeight="Bold"*

*HorizontalAlignment="Center" VerticalAlignment="Center"/>*

*<StackPanelGrid.Row="1" Grid.Column="0" Margin="30,0,0,0">*

*<StackPanel Orientation="Vertical">*

*<Label Content="Наименованиетовара"/>*

*<TextBoxx:Name="Name" Width="350" Height="25" HorizontalAlignment="Left"/>*

*</StackPanel>*

*<StackPanel Orientation="Vertical">*

*<Label Content="Типтовара" />*

*<ComboBox Name="ComboBoxType" Width="200" Height="25" HorizontalAlignment="Left"/>*

*</StackPanel>*

*<StackPanel Orientation="Vertical">*

*<Label Content="Количство"/>*

*<TextBoxx:Name="Quantity" Width="150" Height="25" HorizontalAlignment="Left"/>*

*</StackPanel>*

*<StackPanel Orientation="Vertical">*

*<Label Content="Описаниетовара"/>*

*<TextBoxx:Name="Description" Width="350" Height="150" HorizontalAlignment="Left" TextWrapping="Wrap"/>*

*</StackPanel>*

*</StackPanel>*

*<StackPanelGrid.Row="1" Grid.Column="1" Margin="0,24,0,0">*

*<StackPanel>*

*<Button x:Name="Save" Width="200" Height="25" Content="Сохранить" Click="Save\_Click"*

*Style="{StaticResource Buttons}"/>*

*</StackPanel>*

*<StackPanel Margin="30">*

*<Button x:Name="Delete" Width="200" Height="25" Content="Удалитьтовар" Click="Delete\_Click"*

*Style="{StaticResource Buttons}"/>*

*</StackPanel>*

*<StackPanel Margin="0,0,0,30">*

*<Button x:Name="PaintProd" Width="200" Height="25" Content="Печать" Click="PaintProd\_Click"*

*Style="{StaticResource Buttons}"/>*

*</StackPanel>*

*<StackPanel>*

*<Button x:Name="Cancel" Width="200" Height="25" Content="Отмена" Click="Cancel\_Click"*

*Style="{StaticResource Buttons}"/>*

*</StackPanel>*

*</StackPanel>*

*</Grid>*

*</Page>*

Исходный код страницы добавления и редактирования:

*using System;*

*using System.Linq;*

*using System.Text;*

*using System.Windows;*

*using System.Windows.Controls;*

*using System.Windows.Navigation;*

*using ClassLibraryForPracticeProduction;*

*namespace Practice\_Production.Pages*

*{*

*public partial class AddEditProduct : Page*

*{*

*public Entity.Product \_product = null;*

*public AddEditProduct()*

*{*

*InitializeComponent();*

*ComboBoxType.ItemsSource = App.Context.Types.Select(p =>p.Name).ToList();*

*}*

*public AddEditProduct(Entity.Product product)*

*{*

*InitializeComponent();*

*ComboBoxType.ItemsSource = App.Context.Types.Select(p =>p.Name).ToList();*

*\_product = product;*

*Name.Text = \_product.Name;*

*Quantity.Text = \_product.Quantity.ToString();*

*ComboBoxType.SelectedIndex = (int)\_product.IDType - 1;*

*Description.Text = \_product.Description;*

*}*

*public void Save\_Click(object sender, RoutedEventArgs e)*

*{*

*try*

*{*

*var errorMessage = CheckingForErrorsWhenFillingOut.CheckingErrorsAddEdit(Name.Text, ComboBoxType.Text, Quantity.Text, Description.Text);*

*if (errorMessage.Length> 0)*

*{*

*MessageBox.Show(errorMessage, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);*

*}*

*else*

*{*

*if (\_product == null)*

*{*

*var \_Product = new Entity.Product*

*{*

*Name = Name.Text,*

*Quantity = Convert.ToInt32(Quantity.Text),*

*IDType = ComboBoxType.SelectedIndex + 1,*

*Description = Description.Text,*

*};*

*App.Context.Products.Add(\_Product);*

*App.Context.SaveChanges();*

*MessageBox.Show("Товаруспешнодобавлен", "Добавлениезаписи", MessageBoxButton.OK, MessageBoxImage.Information);*

*NavigationService.Navigate(new Pages.PageStore());*

*}*

*else*

*{*

*\_product.Name = Name.Text;*

*\_product.Quantity = Convert.ToInt32(Quantity.Text);*

*\_product.IDType = ComboBoxType.SelectedIndex + 1;*

*\_product.Description = Description.Text;*

*MessageBox.Show("Записьизменена!", "Редактированиезаписи", MessageBoxButton.OK, MessageBoxImage.Information);*

*App.Context.SaveChanges();*

*NavigationService.Navigate(new Pages.PageStore());*

*}*

*}*

*}*

*catch (Exception ex)*

*{*

*MessageBox.Show("Ошибка" + ex);*

*}*

*}*

*public void Delete\_Click(object sender, RoutedEventArgs e)*

*{*

*if (MessageBox.Show($"Выхотитеудалитьпродукт {\_product.Name}?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)*

*{*

*App.Context.Products.Remove(\_product);*

*App.Context.SaveChanges();*

*NavigationService.Navigate(new Pages.PageStore());*

*}*

*}*

*public void PaintProd\_Click(object sender, RoutedEventArgs e)*

*{*

*Description.TextWrapping = TextWrapping.NoWrap;*

*NavigationService.Navigate(new Pages.SpecificReceipt(Name.Text, ComboBoxType.Text, Quantity.Text, Description.Text));*

*}*

*public void Cancel\_Click(object sender, RoutedEventArgs e)*

*{*

*NavigationService.GoBack();*

*}*

*}*

*}*

Страница печати представлена в соответствии с рисунком 7.

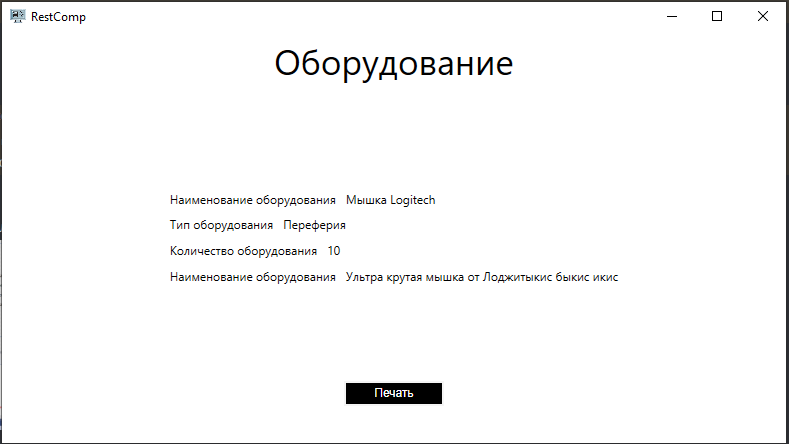


Рисунок 7 – Страница печати

Кодверсткистраницыпечати:

*<Page x:Class="Practice\_Production.Pages.SpecificReceipt"*

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

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

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

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

*xmlns:local="clr-namespace:Practice\_Production.Pages"*

*mc:Ignorable="d"*

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

*Title="SpecificReceipt">*

*<Grid>*

*<Grid.RowDefinitions>*

*<RowDefinition Height="100"/>*

*<RowDefinition/>*

*<RowDefinition Height="100"/>*

*</Grid.RowDefinitions>*

*<Label Content="Оборудование" HorizontalAlignment="Center" VerticalAlignment="Top" FontSize="35" Grid.Row="0"/>*

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

*<StackPanel Orientation="Horizontal">*

*<Label Content="Наименованиеоборудования"/>*

*<Label x:Name="Name"/>*

*</StackPanel>*

*<StackPanel Orientation="Horizontal">*

*<Label Content="Типоборудования"/>*

*<Label x:Name="Type"/>*

*</StackPanel>*

*<StackPanel Orientation="Horizontal">*

*<Label Content="Количествооборудования"/>*

*<Label x:Name="Quantity"/>*

*</StackPanel>*

*<StackPanel Orientation="Horizontal">*

*<Label Content="Наименованиеоборудования"/>*

*<Label x:Name="Description"/>*

*</StackPanel>*

*</StackPanel>*

*<Button Style="{StaticResource Buttons}" Content="Печать" Grid.Row="2" Width="100" Height="25"*

*x:Name="Print" Click="Print\_Click"/>*

*</Grid>*

*</Page>*

Исходный код страницы печати представлен:

*usingSystem;*

*using System.Printing;*

*using System.Windows;*

*using System.Windows.Controls;*

*using System.Windows.Media;*

*using System.Windows.Navigation;*

*namespace Practice\_Production.Pages*

*{*

*public partial class SpecificReceipt : Page*

*{*

*public SpecificReceipt()*

*{*

*InitializeComponent();*

*}*

*public SpecificReceipt(string name, string type, string quantity, string description)*

*{*

*InitializeComponent();*

*Name.Content = name;*

*Type.Content = type;*

*Quantity.Content = quantity;*

*Description.Content = description;*

*}*

*public static void PrintCheck(Visual elementToPrint, string something)*

*{*

*using (var printServer = new LocalPrintServer())*

*{*

*var dialog = new PrintDialog();*

*var qs = printServer.GetPrintQueues();*

*dialog.PrintTicket.PageOrientation = PageOrientation.Portrait;*

*dialog.PrintVisual(elementToPrint, something);*

*}*

*}*

*public void Print\_Click(object sender, RoutedEventArgs e)*

*{*

*Print.Visibility = Visibility.Hidden;*

*PrintCheck(this, Convert.ToString(Name.Content));*

*NavigationService.GoBack();*

*}*

*private void Cancel\_Click(object sender, RoutedEventArgs e)*

*{*

*NavigationService.GoBack();*

*}*

*}*

*}*

6 РАЗМЕЩЕНИЕ РАЗРАБОТАННОГО ПРИЛОЖЕНИЯ В РЕПОЗИТОРИЙ GITHUB.

После того как проект был создан, он был скинут в репозиторий Git. Репозиторий с проектом представлен в соответствии с рисунком 8.

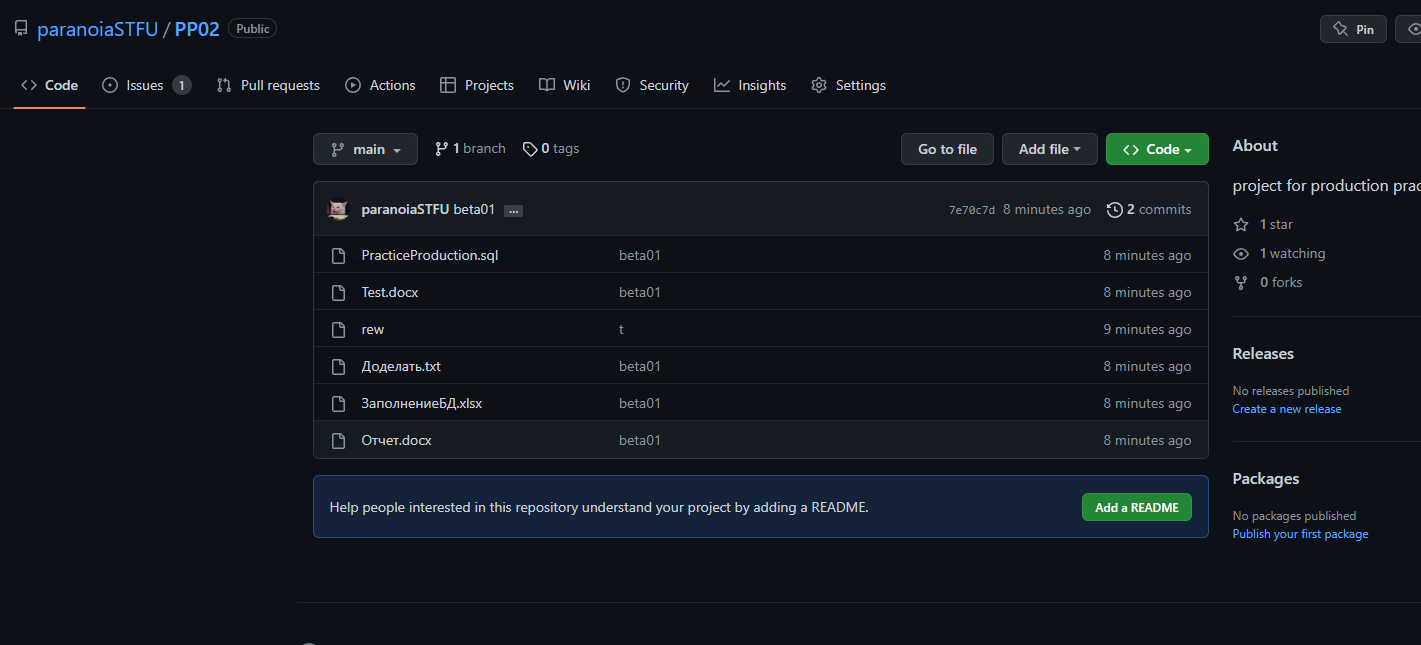


Рисунок 8 – Проект в Git

7 ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ

Результаты UnitTestпредставлены в соответствии с рисунком 9.

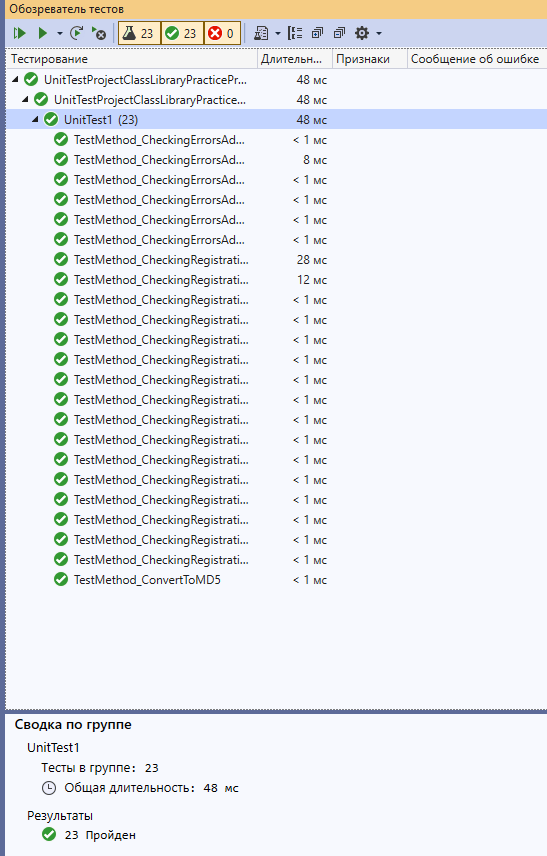


Рисунок 9 – Результаты UnitTest

Исходный код файла UnitTest.

*using ClassLibraryForPracticeProduction;*

*using Microsoft.VisualStudio.TestTools.UnitTesting;*

*using System.Text;*

*namespace UnitTestProjectClassLibraryPracticeProduction*

*{*

*[TestClass]*

*public class UnitTest1*

*{*

*[TestMethod]*

*public void TestMethod\_ConvertToMD5()*

*{*

*string a = "Текстдлякодировки";*

*Assert.AreEqual(ClassConvertToMD5.ConvertToMD5(a), "5d2f1166382e1f870e7581607ffba685");*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Login\_Length\_False()*

*{*

*string Login = "Login";*

*string Password = "Password01!";*

*string PasswordCopy = "Password01!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*varres = newStringBuilder();*

*res.AppendLine("Логин не должен быть меньше шести символов");*

*res.Insert(0, "Устраните ошибки: \n");*

*Assert.AreEqual<string>(ErrorMesseg, res.ToString());*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Login\_Length\_True()*

*{*

*stringres = "Логин не должен быть меньше шести символов";*

*string Login = "LoginNormal01!";*

*string Password = "Password01!";*

*string PasswordCopy = "Password01!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*if(ErrorMesseg.Length==0)*

*res = "Пользователь успешно зарегистрирован.";*

*Assert.AreEqual(res, "Пользователь успешно зарегистрирован.");*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Login\_Fill\_False()*

*{*

*string Login = "";*

*string Password = "Password01!";*

*string PasswordCopy = "";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*StringBuilder res = new StringBuilder();*

*res.AppendLine("Вынезаполниливсеполя");*

*res.Insert(0, "Устраните ошибки: \n");*

*Assert.AreEqual<string>(ErrorMesseg, res.ToString());*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Login\_Fill\_True()*

*{*

*stringres = "Вы не заполнили все поля";*

*string Login = "LoginNormal01!";*

*string Password = "Password01!";*

*string PasswordCopy = "Password01!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*if (ErrorMesseg.Length == 0)*

*res = "Пользователь успешно зарегистрирован.";*

*Assert.AreEqual(res, "Пользователь успешно зарегистрирован.");*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Copy\_Password\_False()*

*{*

*string Login = "LoginNormal1!";*

*string Password = "Password01!";*

*string PasswordCopy = "Пароль01!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*StringBuilder res = new StringBuilder();*

*res.AppendLine("Паролинесовпадают.");*

*res.Insert(0, "Устраните ошибки: \n");*

*Assert.AreEqual<string>(ErrorMesseg, res.ToString());*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Copy\_Password\_True()*

*{*

*stringres = "Пароли не совпадают.";*

*string Login = "LoginOnTest";*

*string Password = "Password01!";*

*string PasswordCopy = "Password01!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*if (ErrorMesseg.Length == 0)*

*res = "Пользователь успешно зарегистрирован.";*

*Assert.AreEqual(res, "Пользователь успешно зарегистрирован.");*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Password\_Length\_False()*

*{*

*string Login = "LoginNormal1!";*

*string Password = "Pas1!";*

*string PasswordCopy = "Pas1!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*StringBuilder res = new StringBuilder();*

*res.AppendLine("Парольнедолженбытьменьшешестисимволов");*

*res.Insert(0, "Устраните ошибки: \n");*

*Assert.AreEqual<string>(ErrorMesseg, res.ToString());*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Password\_Length\_True()*

*{*

*stringres = "Пароль не должен быть меньше шести символов";*

*string Login = "LoginOnTest";*

*string Password = "PasswordOnTest01!";*

*string PasswordCopy = "PasswordOnTest01!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*if (ErrorMesseg.Length == 0)*

*res = "Пользователь успешно зарегистрирован.";*

*Assert.AreEqual(res, "Пользователь успешно зарегистрирован.");*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Password\_Char\_False()*

*{*

*string Login = "LoginNormal1!";*

*string Password = "a123456!";*

*string PasswordCopy = "a123456!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*StringBuilderres = newStringBuilder();*

*res.AppendLine("В пароле должна присутствовать хотя бы одна прописная буква");*

*res.Insert(0, "Устраните ошибки: \n");*

*Assert.AreEqual<string>(ErrorMesseg, res.ToString());*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Password\_Char\_True()*

*{*

*stringres = "В пароле должна присутствовать хотя бы одна прописная буква";*

*string Login = "LoginOnTest";*

*string Password = "PasswordOnTest01!";*

*string PasswordCopy = "PasswordOnTest01!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*if (ErrorMesseg.Length == 0)*

*res = "Пользователь успешно зарегистрирован.";*

*Assert.AreEqual(res, "Пользователь успешно зарегистрирован.");*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Login\_No\_Char\_RU\_False()*

*{*

*string Login = "ЭтоМойЛгин1!";*

*string Password = "A123456!";*

*string PasswordCopy = "A123456!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*StringBuilder res = new StringBuilder();*

*res.AppendLine("Логиндолженсостоятьизбуквлатинскогоалфавита");*

*res.Insert(0, "Устраните ошибки: \n");*

*Assert.AreEqual<string>(ErrorMesseg, res.ToString());*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Login\_No\_Char\_RU\_True()*

*{*

*stringres = "Логин должен состоять из букв латинского алфавита";*

*string Login = "LoginOnTest";*

*string Password = "PasswordOnTest01!";*

*string PasswordCopy = "PasswordOnTest01!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*if (ErrorMesseg.Length == 0)*

*res = "Пользователь успешно зарегистрирован.";*

*Assert.AreEqual(res, "Пользователь успешно зарегистрирован.");*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Int\_in\_Password\_False()*

*{*

*string Login = "MyLogin!";*

*string Password = "MyPassword!";*

*string PasswordCopy = "MyPassword!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*StringBuilder res = new StringBuilder();*

*res.AppendLine("Впароледолжнаприсутствоватьхотябыоднацифра");*

*res.Insert(0, "Устраните ошибки: \n");*

*Assert.AreEqual<string>(ErrorMesseg, res.ToString());*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Int\_in\_Password\_True()*

*{*

*stringres = "В пароле должна присутствовать хотя бы одна цифра";*

*string Login = "LoginOnTest";*

*string Password = "PasswordOnTest01!";*

*string PasswordCopy = "PasswordOnTest01!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*if (ErrorMesseg.Length == 0)*

*res = "Пользователь успешно зарегистрирован.";*

*Assert.AreEqual(res, "Пользователь успешно зарегистрирован.");*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Special\_Char\_in\_Password\_False()*

*{*

*string Login = "MyLogin!";*

*string Password = "MyPassword1";*

*string PasswordCopy = "MyPassword1";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*StringBuilder res = new StringBuilder();*

*res.AppendLine("Впароледолженприсутствоватьодинсимволизнабора ( ! @ # $ % ^)");*

*res.Insert(0, "Устранитеошибки: \n");*

*Assert.AreEqual<string>(ErrorMesseg, res.ToString());*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingRegistration\_Special\_Char\_in\_Password\_True()*

*{*

*stringres = "В пароле должен присутствовать один символ из набора ( !@ # $ % ^)";*

*string Login = "LoginOnTest";*

*string Password = "PasswordOnTest01!";*

*string PasswordCopy = "PasswordOnTest01!";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingRegistration(Login, Password, PasswordCopy);*

*if (ErrorMesseg.Length == 0)*

*res = "Пользователь успешно зарегистрирован.";*

*Assert.AreEqual(res, "Пользователь успешно зарегистрирован.");*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingErrorsAddEdit\_Zapolnenie\_False()*

*{*

*string Name = "Удленитель USB";*

*string ComboBoxType = "";*

*string Quantity = "5";*

*string Description = "Удлинитель USB на USB, Male-Male, 2 метра";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingErrorsAddEdit(Name, ComboBoxType, Quantity, Description);*

*StringBuilder res = new StringBuilder();*

*res.AppendLine("Вынезаполниливсеполя");*

*res.Insert(0, "Устраните ошибки: \n");*

*Assert.AreEqual<string>(ErrorMesseg, res.ToString());*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingErrorsAddEdit\_Zapolnenie\_True()*

*{*

*stringres = "Вы не заполнили все поля";*

*string Name = "Удленитель USB";*

*string ComboBoxType = "Провода";*

*string Quantity = "5";*

*string Description = "Удлинитель USB на USB, Male-Male, 2 метра";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingErrorsAddEdit(Name, ComboBoxType, Quantity, Description);*

*if (ErrorMesseg.Length == 0)*

*res = "Сохранено";*

*Assert.AreEqual<string>(res, "Сохранено");*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingErrorsAddEdit\_Quantity\_its\_chislo\_False()*

*{*

*string Name = "Удленитель USB";*

*string ComboBoxType = "Провода";*

*string Quantity = "Пять";*

*string Description = "Удлинитель USB на USB, Male-Male, 2 метра";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingErrorsAddEdit(Name, ComboBoxType, Quantity, Description);*

*StringBuilder res = new StringBuilder();*

*res.AppendLine("Количествооборудованиядолжнобытьчислом!");*

*res.Insert(0, "Устраните ошибки: \n");*

*Assert.AreEqual<string>(ErrorMesseg, res.ToString());*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingErrorsAddEdit\_Quantity\_Its\_Chislo\_True()*

*{*

*stringres = "Кол-во должно быть числом";*

*string Name = "Удленитель USB";*

*string ComboBoxType = "Провода";*

*string Quantity = "5";*

*string Description = "Удлинитель USB на USB, Male-Male, 2 метра";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingErrorsAddEdit(Name, ComboBoxType, Quantity, Description);*

*if (ErrorMesseg.Length == 0)*

*res = "Сохранено";*

*Assert.AreEqual<string>(res, "Сохранено");*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingErrorsAddEdit\_Quantity\_Its\_Not\_Negative\_Chislo\_False()*

*{*

*string Name = "Удленитель USB";*

*string ComboBoxType = "Провода";*

*string Quantity = "-5";*

*string Description = "Удлинитель USB на USB, Male-Male, 2 метра";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingErrorsAddEdit(Name, ComboBoxType, Quantity, Description);*

*StringBuilder res = new StringBuilder();*

*res.AppendLine("Количествооборудованиянедолжнобытьотрицательным!");*

*res.Insert(0, "Устраните ошибки: \n");*

*Assert.AreEqual<string>(ErrorMesseg, res.ToString());*

*}*

*[TestMethod]*

*public void TestMethod\_CheckingErrorsAddEdit\_Quantity\_Its\_Not\_Negative\_Chislo\_True()*

*{*

*stringres = "Кол-во должно быть не отрицателным числом";*

*string Name = "Удленитель USB";*

*string ComboBoxType = "Провода";*

*string Quantity = "5";*

*string Description = "Удлинитель USB на USB, Male-Male, 2 метра";*

*string ErrorMesseg = CheckingForErrorsWhenFillingOut.CheckingErrorsAddEdit(Name, ComboBoxType, Quantity, Description);*

*if (ErrorMesseg.Length == 0)*

*res = "Сохранено";*

*Assert.AreEqual<string>(res, "Сохранено");*

*}*

*}*

*}*

Проверка корректности кодировки пароля в MD5представлена в соответствии с рисунком 10.

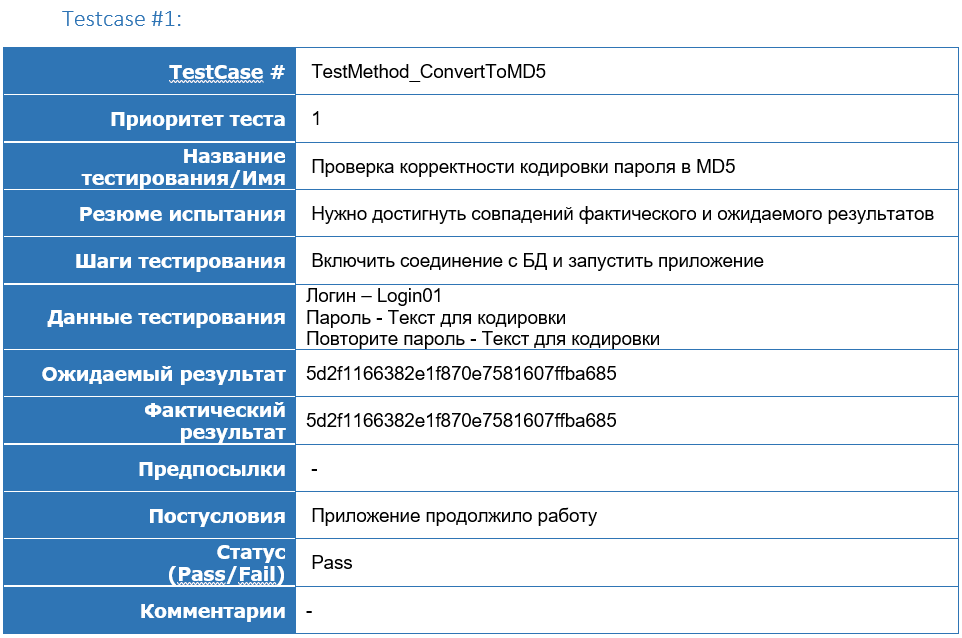


Рисунок10­– Проверка корректности кодировки пароля в MD5

Проверка вывода сообщения при вводе логина, не соответствующей длиныпредставленав соответствии с рисунком 11.

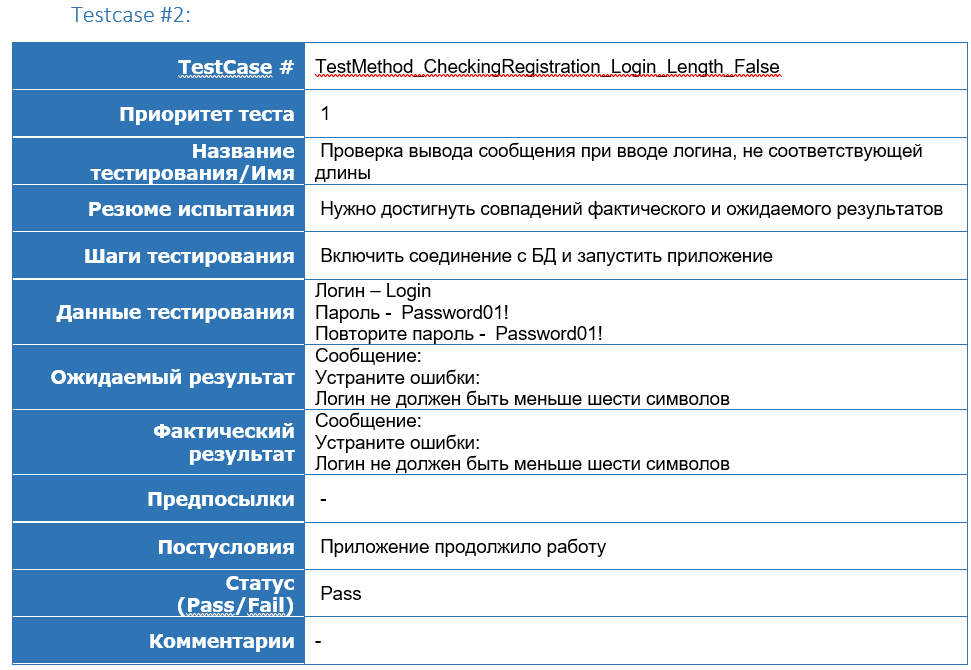


Рисунок 11­­– Проверка вывода сообщения при вводе логина, не соответствующей длины

Проверка вывода сообщения при вводе логина, соответствующей длиныпредставленав соответствии с рисунком 12.

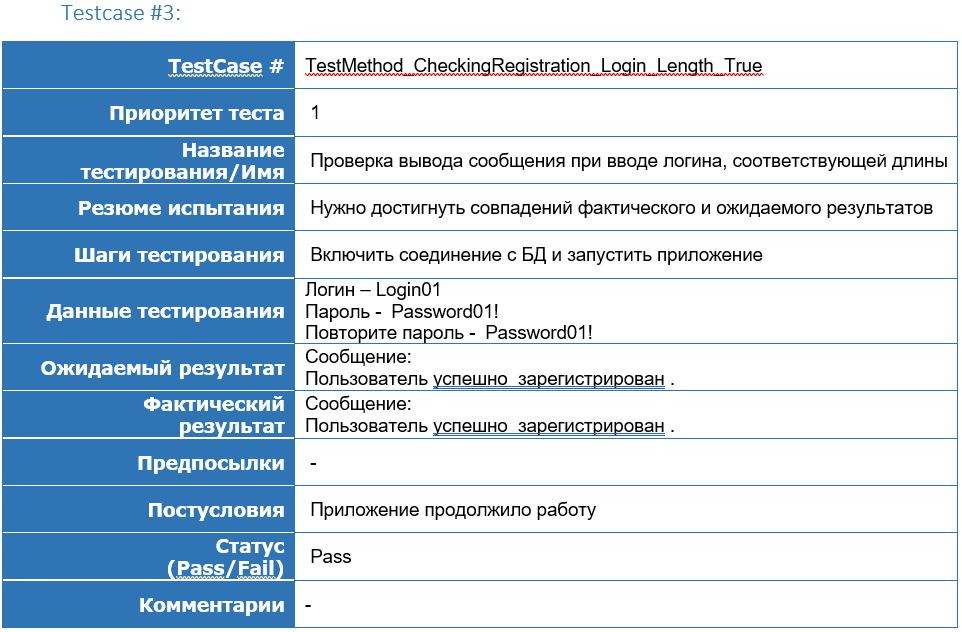


Рисунок 12­­– Проверка вывода сообщения при вводе логина, соответствующей длины

Проверка вывода сообщения при заполнении не всех полейпредставленав соответствии с рисунком 13.

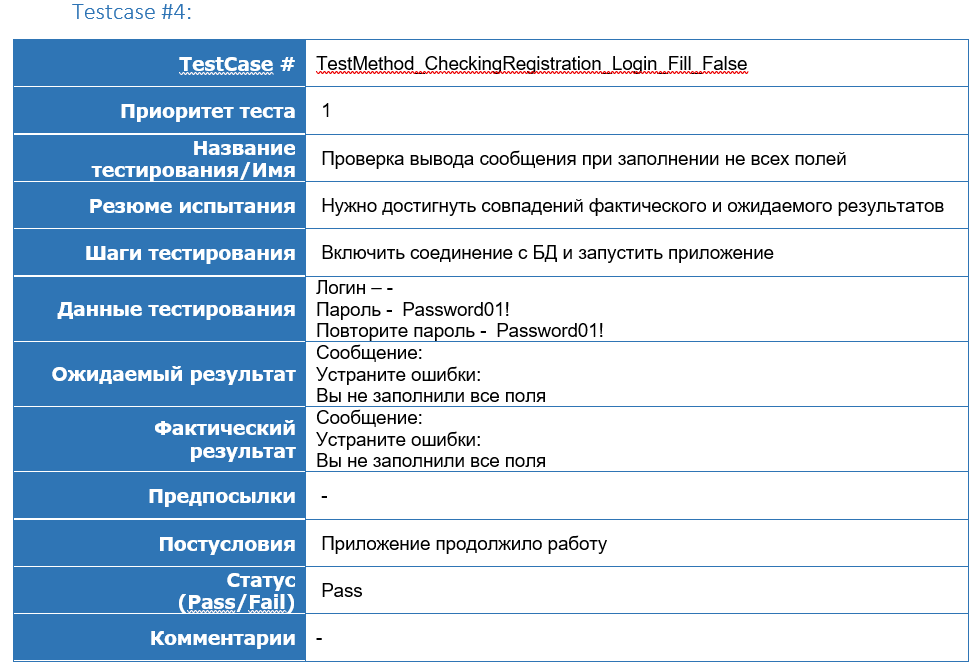


Рисунок 13­­– Проверка вывода сообщения при заполнении не всех полей

Проверка вывода сообщения при заполнении всех полей представленав соответствии с рисунком 14.

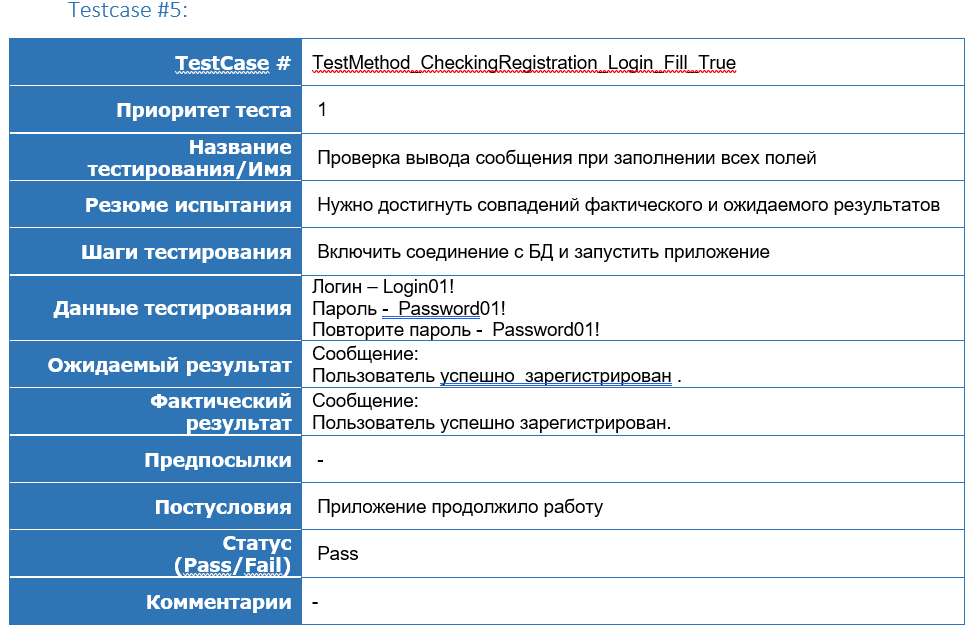


Рисунок 14­­– Проверка вывода сообщения при заполнении всех полей

Проверка вывода сообщения при не верном заполнении поля «повторите пароль»представленав соответствии с рисунком 15.

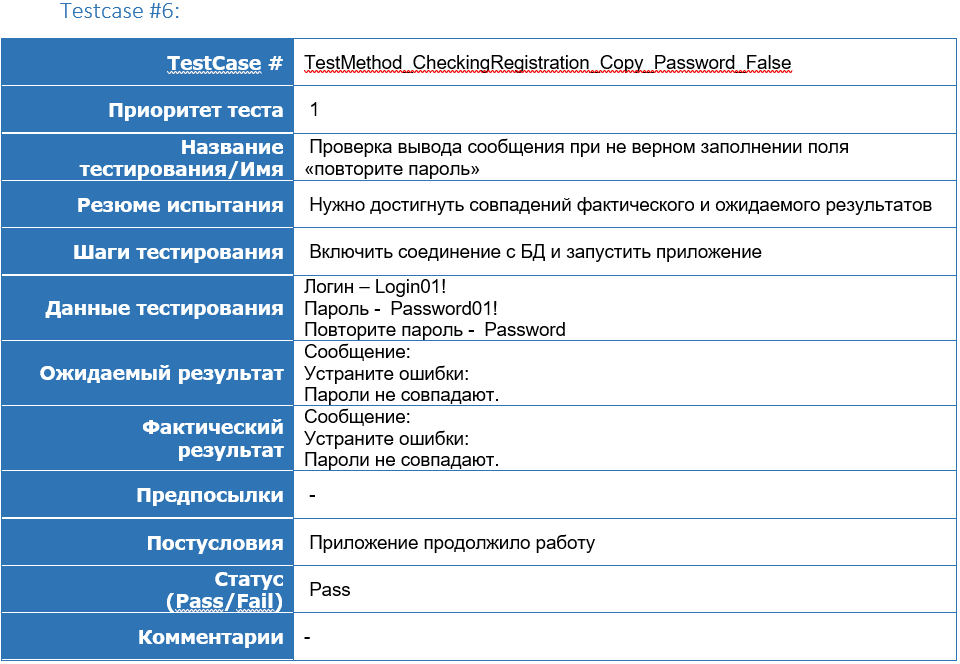


Рисунок 15­­– Проверка вывода сообщения при не верном заполнении поля «повторите пароль»

Проверка вывода сообщения при верном заполнении поля «повторите пароль»представленав соответствии с рисунком 16.

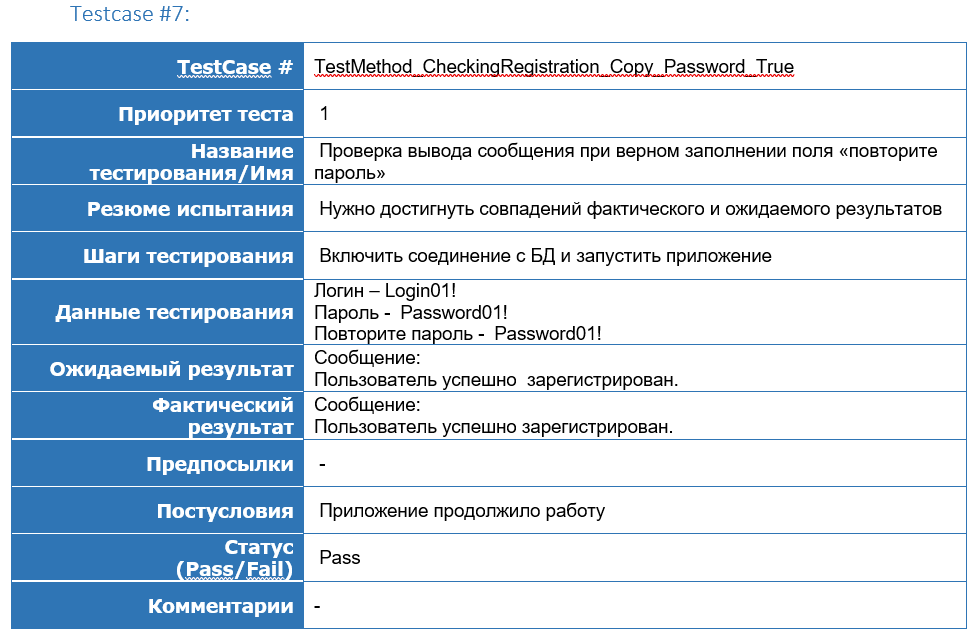


Рисунок 16­­– Проверка вывода сообщения при верном заполнении поля «повторите пароль»

Проверка вывода сообщения при заполнении поля «пароль» неверной длиныпредставленав соответствии с рисунком 17.

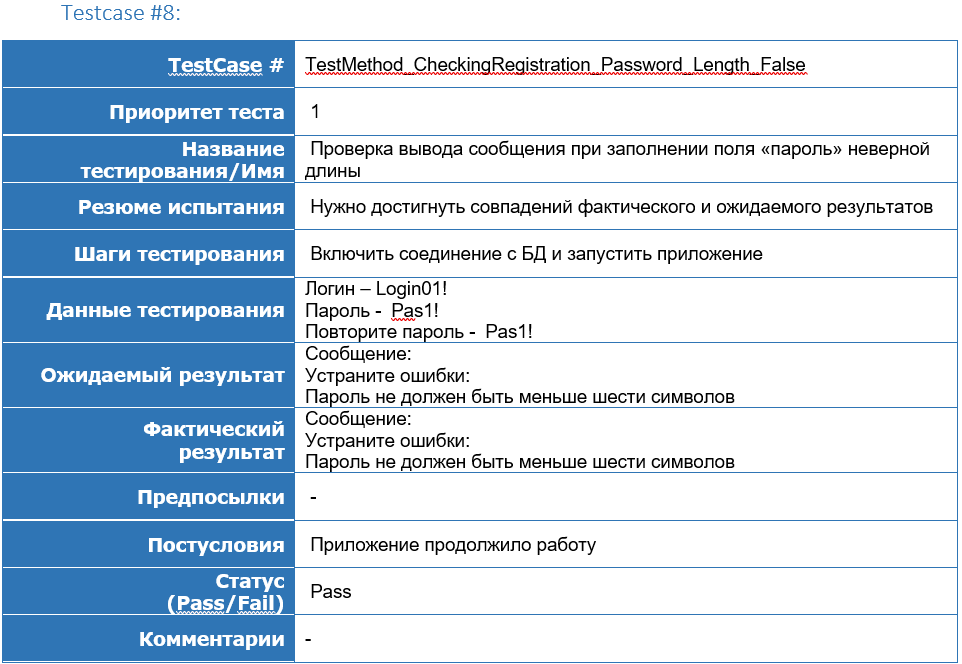


Рисунок 17­­– Проверка вывода сообщения при заполнении поля «пароль» неверной длины

Проверка вывода сообщения при заполнении поля «пароль» верной длиныпредставлена в соответствии с рисунком 18.

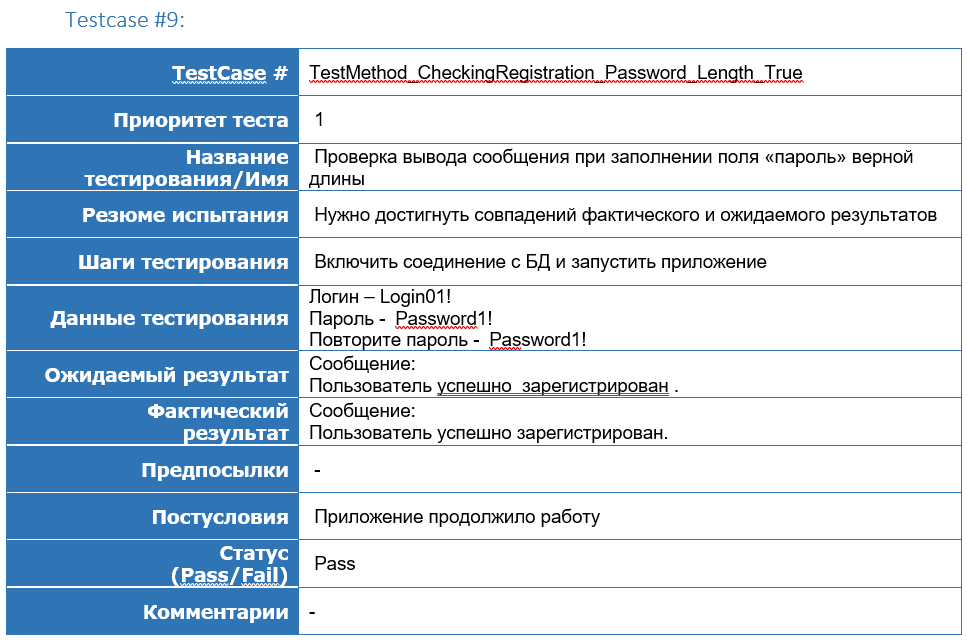


Рисунок 18­­– Проверка вывода сообщения при заполнении поля «пароль» верной длины

Проверка вывода сообщения при отсутствии в пароле прописной буквыпредставленав соответствии с рисунком 19.

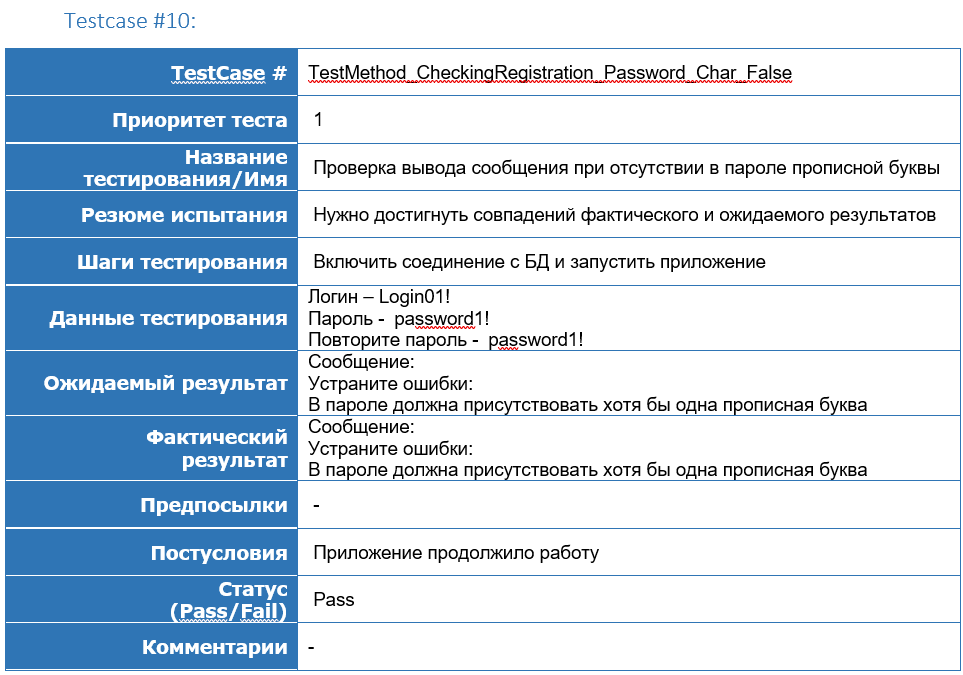


Рисунок 19­­– Проверка вывода сообщения при отсутствии в пароле прописной буквы

Проверка вывода сообщения при наличии в пароле прописной буквы представлена в соответствии с рисунком 20.

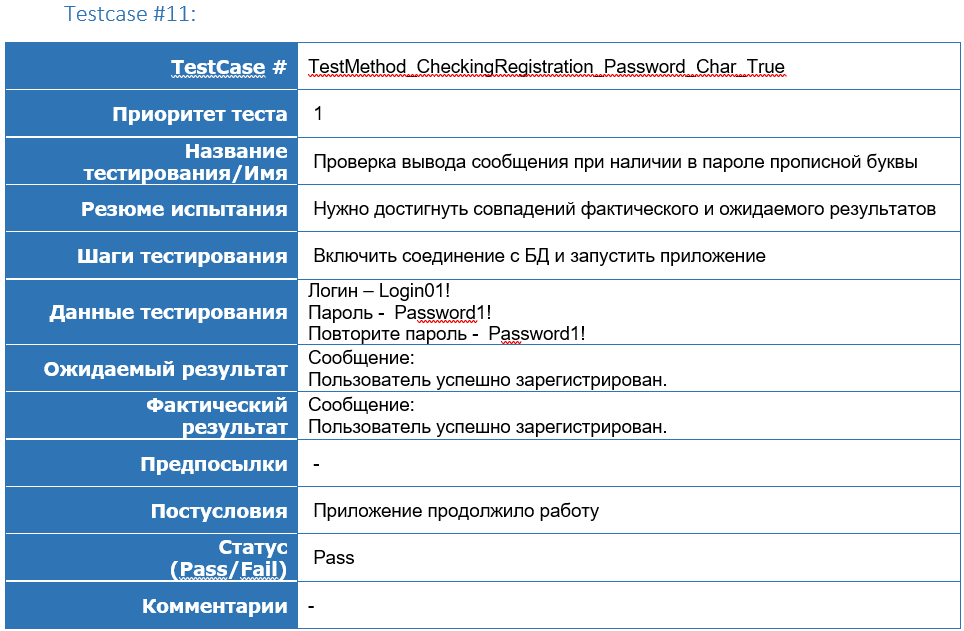


Рисунок 20­­– Проверка вывода сообщения при наличии в пароле прописной буквы

Проверка вывода сообщения при наличии в пароле кириллицыпредставленав соответствии с рисунком 21.

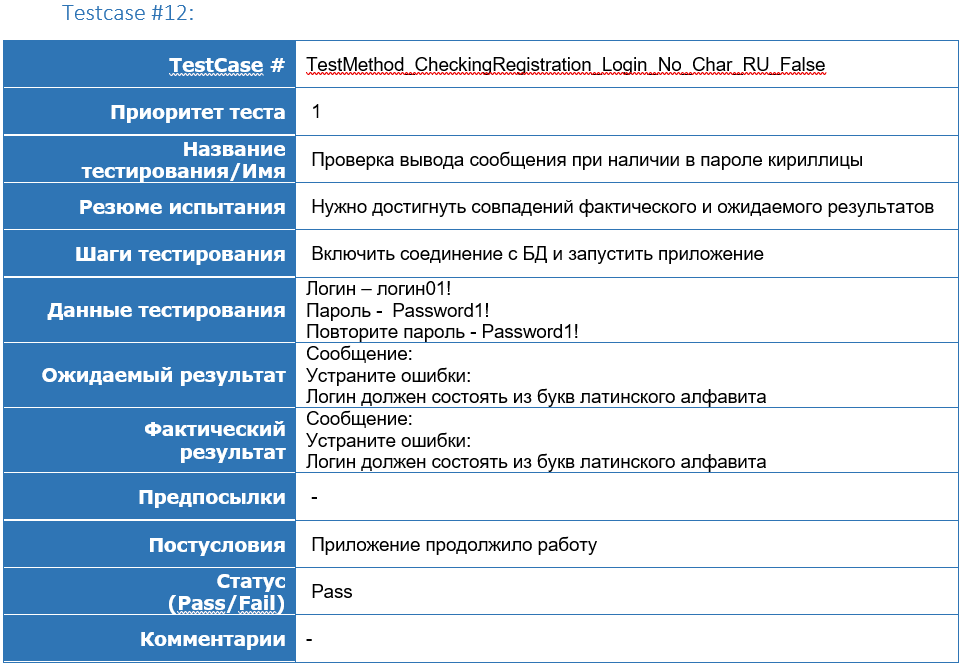


Рисунок 21­­– Проверка вывода сообщения при наличии в пароле кириллицы

Проверка вывода сообщения при отсутствии в пароле кириллицыпредставленав соответствии с рисунком 22.

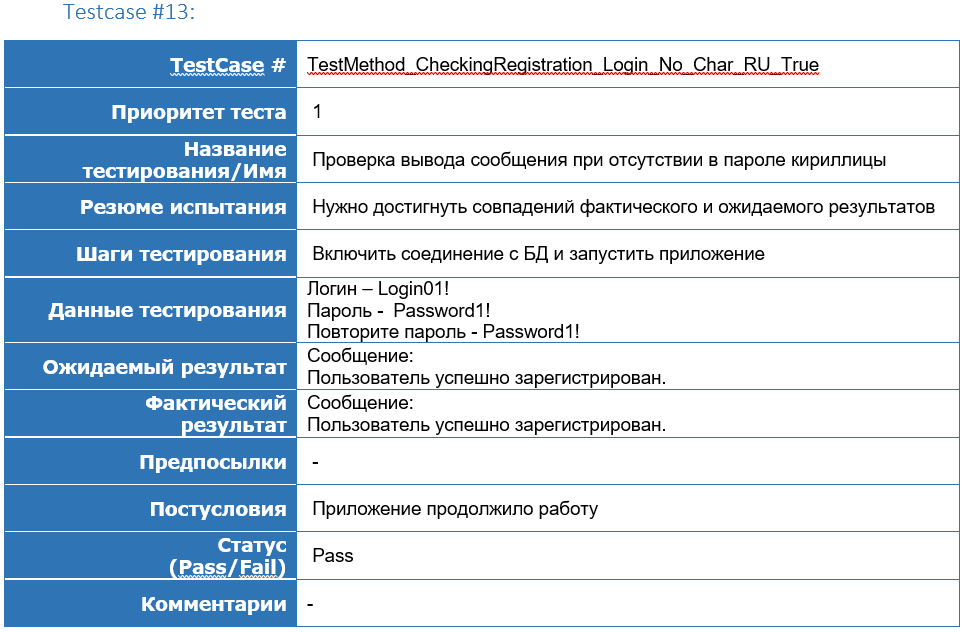


Рисунок 22­­– Проверка вывода сообщения при отсутствии в пароле кириллицы

Проверка вывода сообщения при отсутствии в пароле цифрыпредставленав соответствии с рисунком 23.

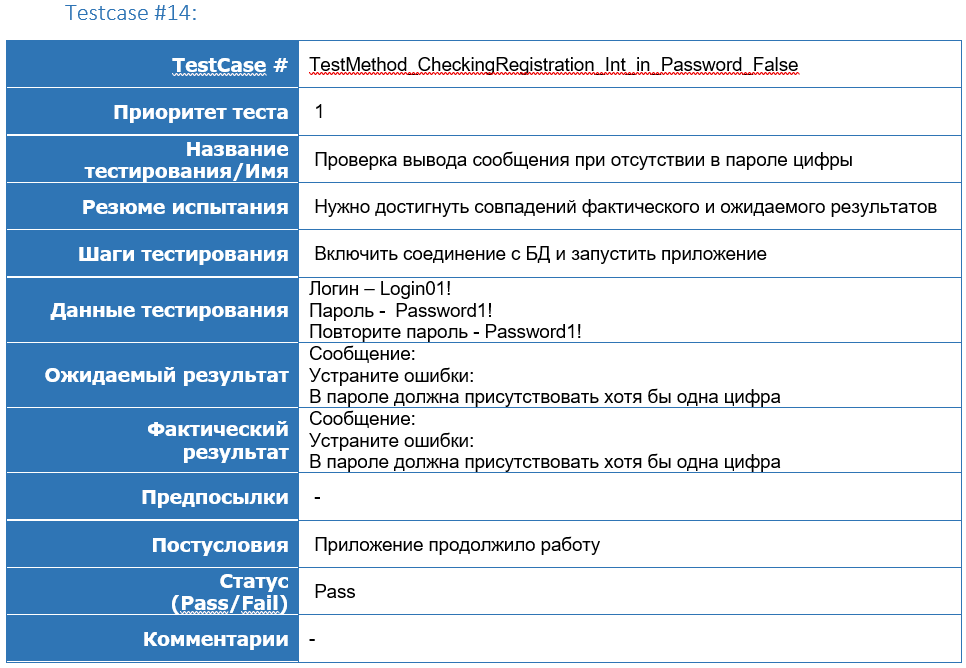


Рисунок 23­­– Проверка вывода сообщения при отсутствии в пароле цифры

Проверка вывода сообщения при наличии в пароле цифрыпредставленав соответствии с рисунком 24.

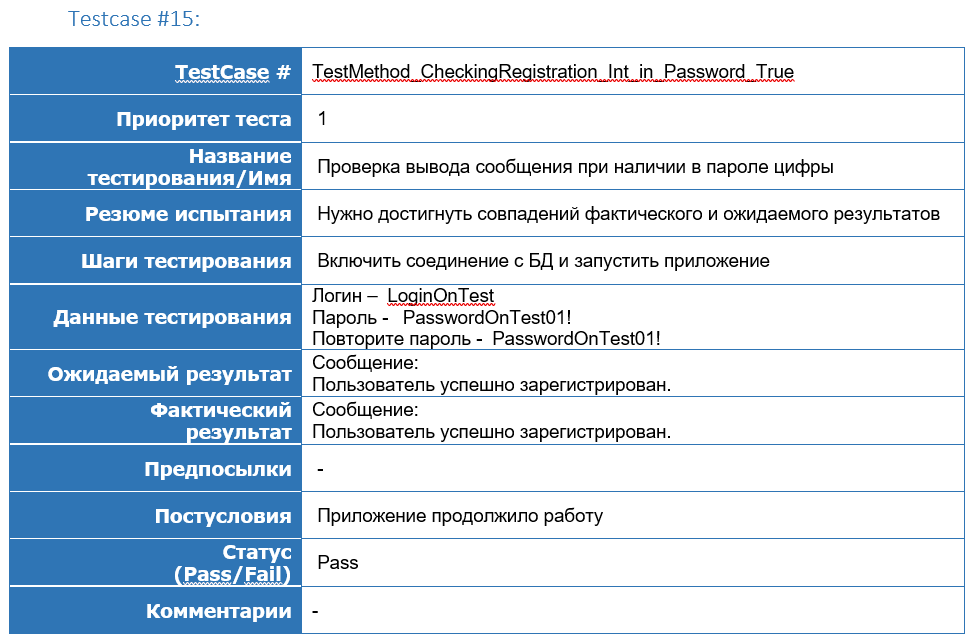


Рисунок 24­­– Проверка вывода сообщения при наличии в пароле цифры

Проверка вывода сообщения при отсутствии в пароле специального символапредставленав соответствии с рисунком 25.

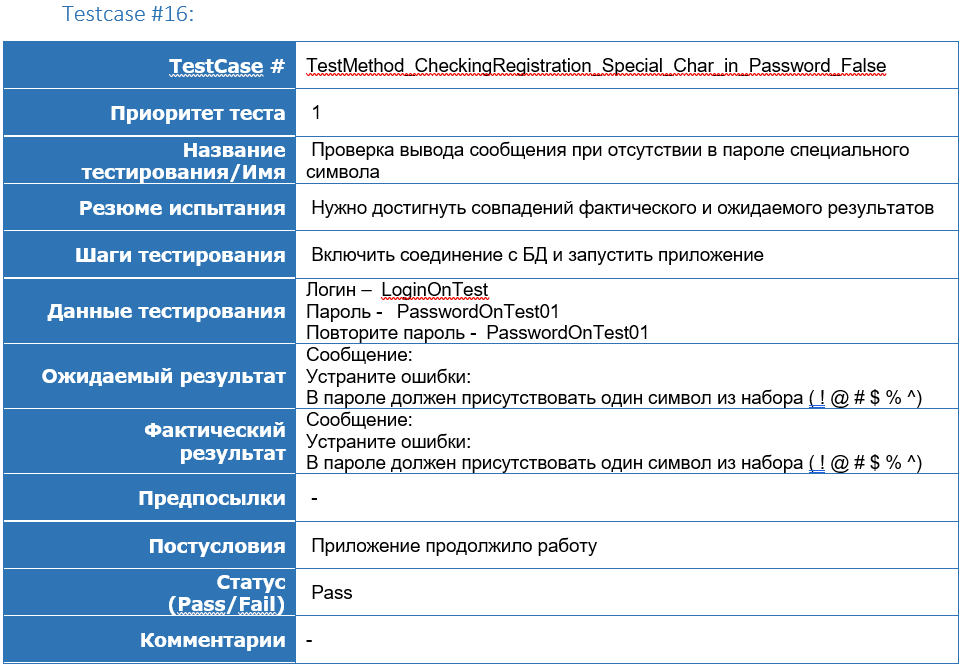


Рисунок 25­­– Проверка вывода сообщения при отсутствии в пароле специального символа

Проверка вывода сообщения при наличии в пароле специального символапредставленав соответствии с рисунком 26.

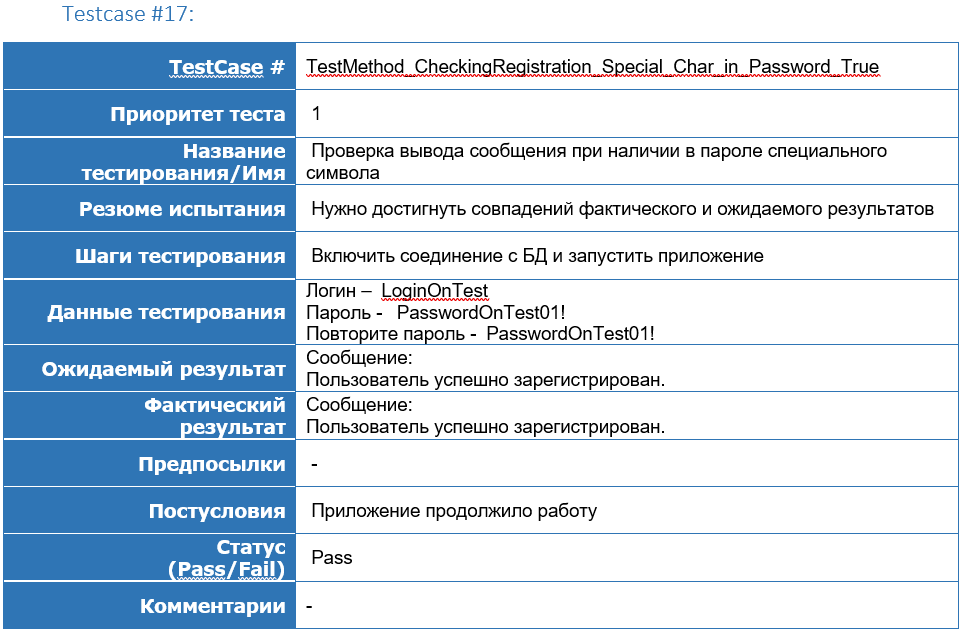


Рисунок 26­­– Проверка вывода сообщения при наличии в пароле специального символа

Проверка вывода сообщения заполнении не всех полей при добавлении или редактированиипредставленав соответствии с рисунком 27.

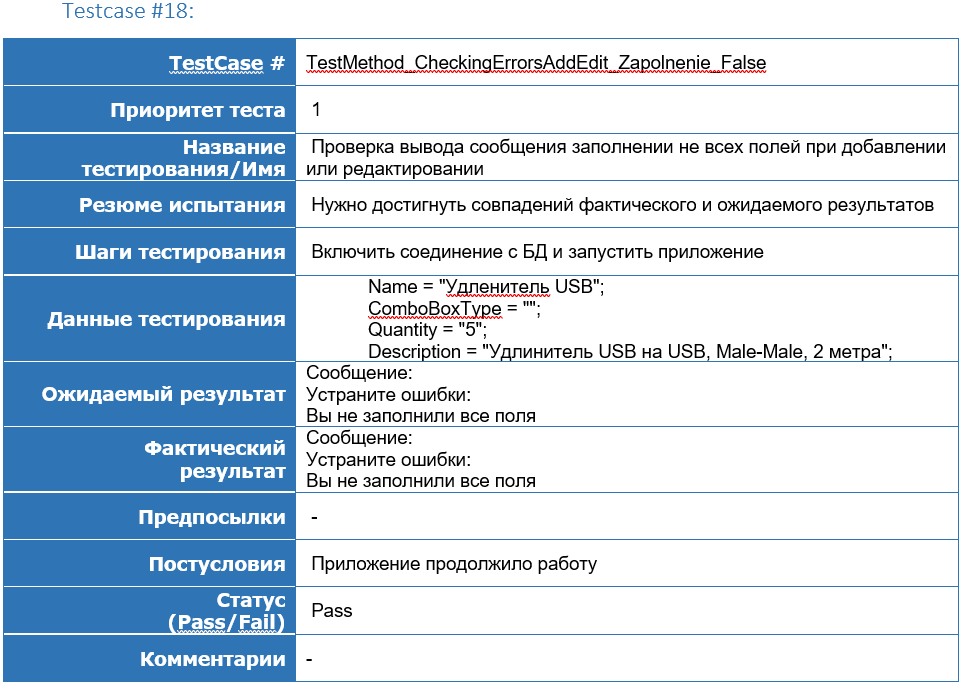


Рисунок 28­­– Проверка вывода сообщения заполнении не всех полей при добавлении или редактировании

Проверка вывода сообщения заполнении всех полей при добавлении или редактированиипредставленав соответствии с рисунком 29.

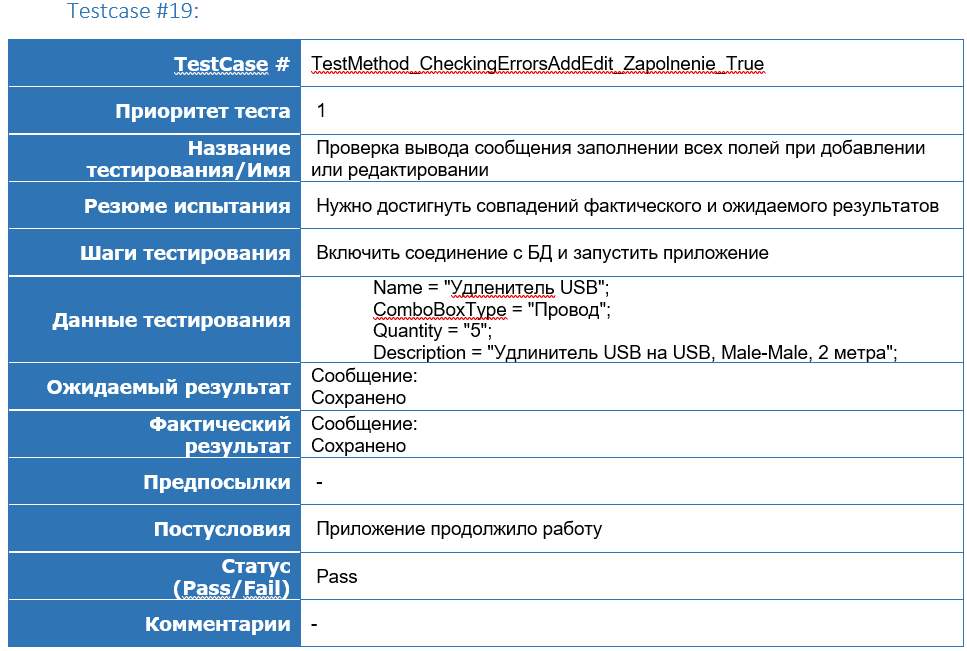


Рисунок 30­­– Проверка вывода сообщения заполнении всех полей при добавлении или редактировании

Проверка вывода сообщения при заполнении в поле «количество» строкипредставленав соответствии с рисунком 31.

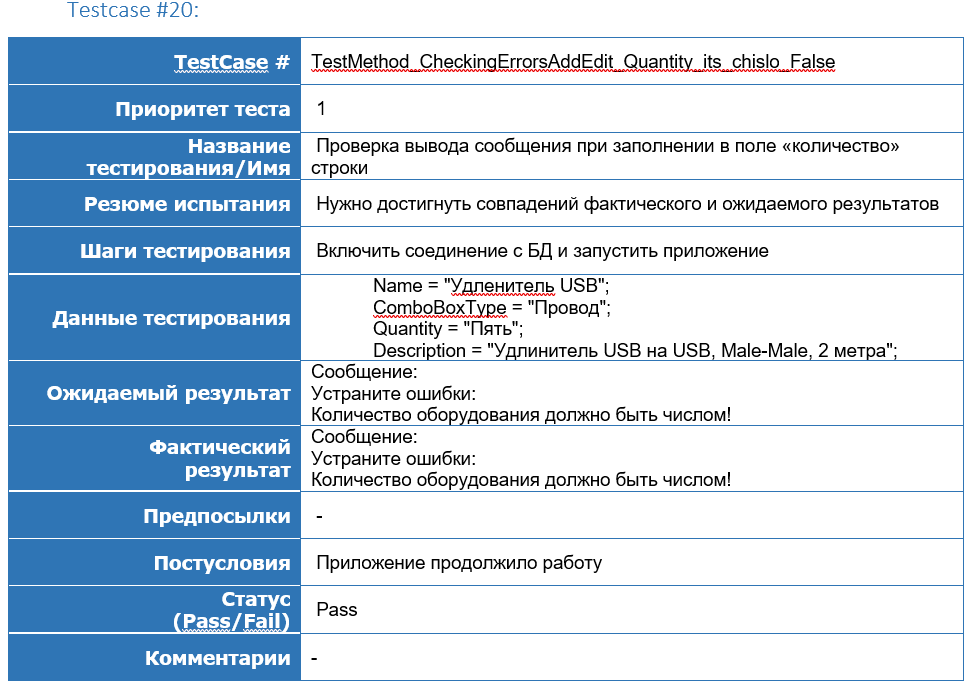


Рисунок 31­­– Проверка вывода сообщения при заполнении в поле «количество» строки

Проверка вывода сообщения при заполнении в поле «количество» числапредставленав соответствии с рисунком 32.

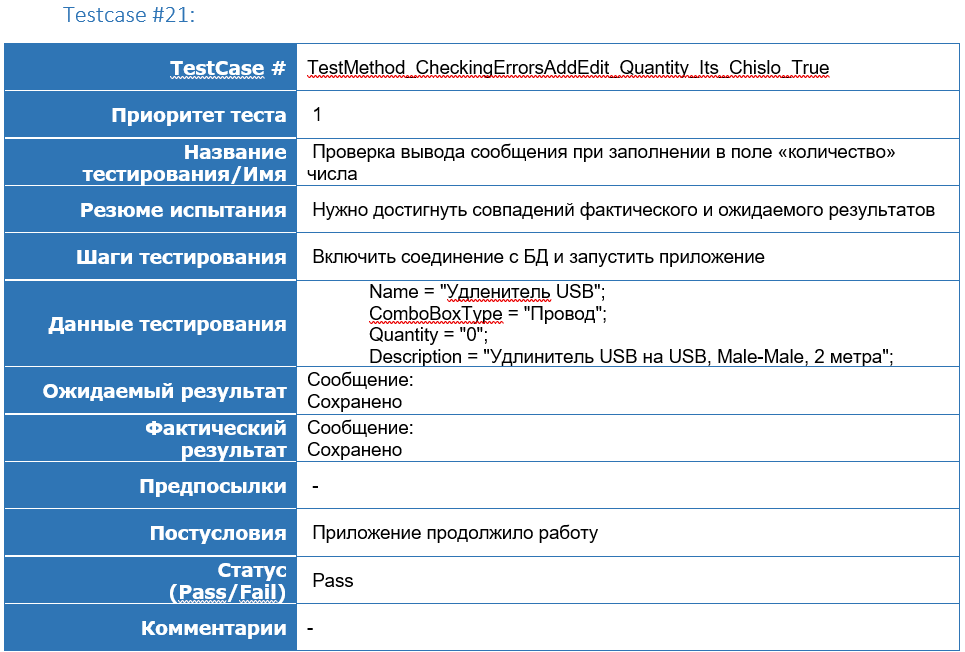


Рисунок 32­­– Проверка вывода сообщения при заполнении в поле «количество» числа

Проверка вывода сообщения при заполнении в поле «количество» отрицательного числапредставленав соответствии с рисунком 33.

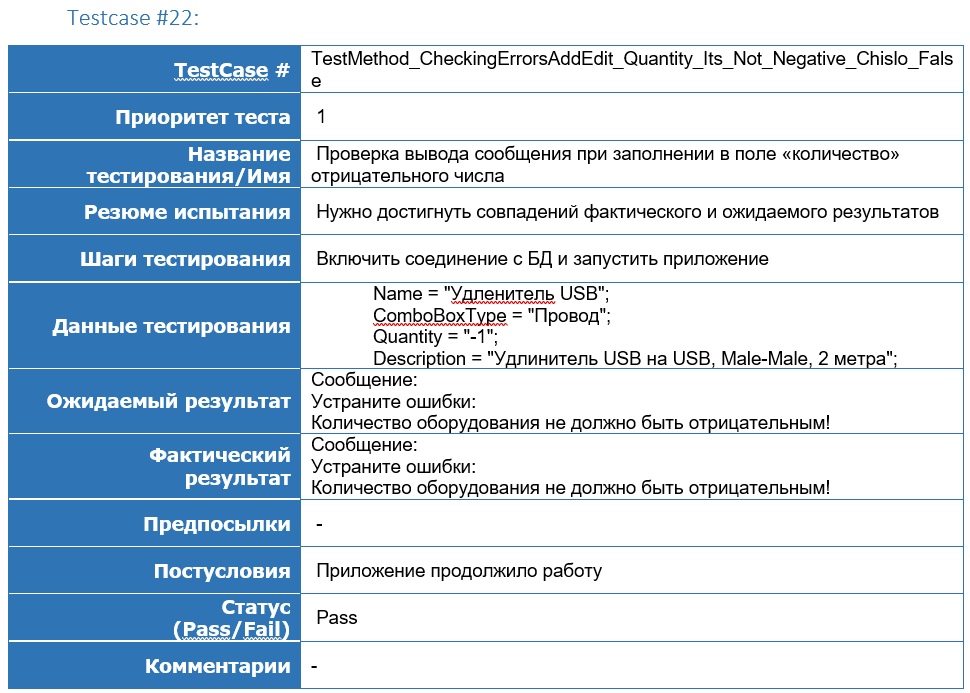


Рисунок 33­­– Проверка вывода сообщения при заполнении в поле «количество» отрицательного числа

Проверка вывода сообщения при заполнении в поле «количество» положительного числапредставленав соответствии с рисунком 34.

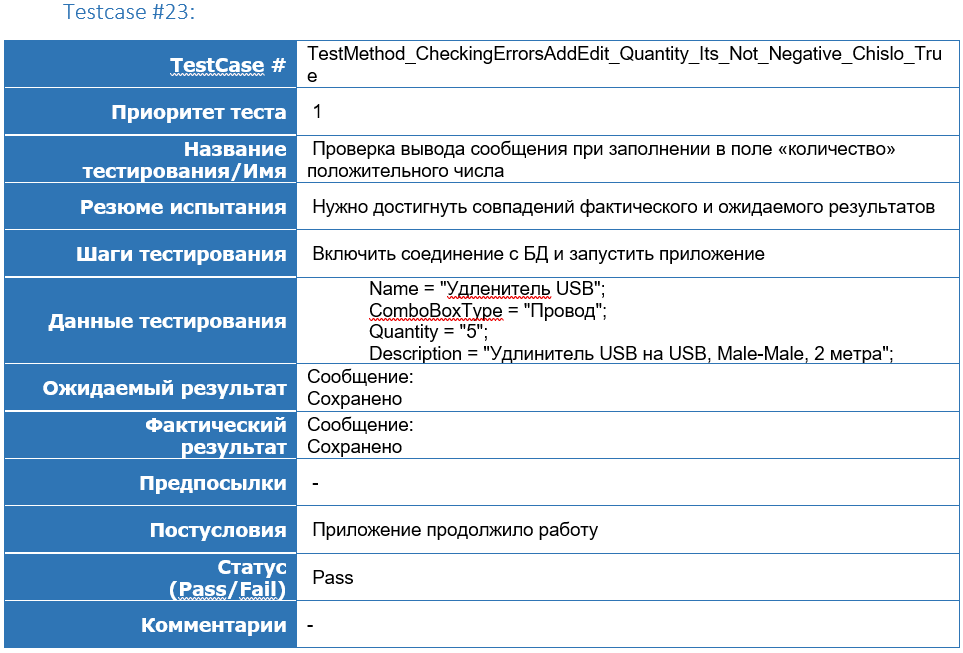


Рисунок 34­­– Проверка вывода сообщения при заполнении в поле «количество» положительного числа

ЗАКЛЮЧЕНИЕ

Итогом производственной практики является программное приложение для ведения списка оборудования организации ВРОО «Созидание», которое было написано на языке программирования C# в среде разработки VisualStudio 2022 с использованием графического интерфейса WPF, а также была создана база данных.

В ходе данной производственной практики был приобретен опыт работы с коллективом в организации.