

Отчет по тестированию

«Книжный магазин»

# Постановка задачи

Провести отладку и тестирование данного приложения.

a) Провести модульное-тестирование функции подсчета скидки за покупку (для трех различных наборов данных).

b) Провести тестирование дизайна интерфейса приложения.

c) Разработать отчет по тестированию, включающий в себя описание тестовых случаев, таблицу результатов тестирования, отчет об ошибках и отчет по тестированию дизайна интерфейса.

# Цель тестирования

**Чек-лист: Книжный магазин**

1. ID 001 - Появление всего списка книг
2. ID 002 - Возможность заказа книг

# Аппаратные и программные средства

Visual Studio и MS SQL SQL Server

# Теоретические сведения

Область применения:

Данное приложение предназначено для просмотра списка книг с возможностью заказа.

Краткое описание возможностей:

* Просмотр каталога книг;
* Возможность заказа книги;
* Расчет скидки за покупку.

# Тест-кейсы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID 001 | | Тест-кейс: Появление всего списка книг | | |
| Дергачев Д.А. | Панкова Е.М. | |  | С# |
|  | | | | |
| Нажать на кнопку запуск в приложении VS | | Ожидаемый результат:  Программа открывает весь каталог книг без ошибок | | |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID 002 | | Тест-кейс: Возможность заказа книг | | |
| Дергачев Д.А. | Панкова Е.М. | |  | С# |
|  | | | | |
| 1. Выбрать из каталога книгу 2. Нажать на кнопку добавить | | Ожидаемый результат:  Все добавленные книги отображаются выше | | |
| Макет основной страницы приложения | | | | |

# Итоги тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Введенные данные | Ожидаемый результат | Полученный результат | Примечание |
| Нажата кнопка «Добавить» | Книга добавилась в корзину | Не работает | Нужно реализовать функцию добавления книг в корзину |
| Добавлено больше 3 книг | Сумма общей стоимости книг посчиталась с акцией | Не работает | Прописать функцию расчета скидки |
| Запуск приложения | Появляется каталог книг | Не все содержание в каталоге прописано | Дописать все остальное содержания к каталогу |

# Исходный код программы

Код разметки:

<Page x:Class="ПанковаЕМ.Admin"

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:ПанковаЕМ"

mc:Ignorable="d"

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

Title="Admin">

<Grid Background="#fff">

<ScrollViewer>

<ScrollViewer>

<StackPanel Name="MSP" Visibility="Visible">

<StackPanel>

<DataGrid Name="DGServises" AutoGenerateColumns="False" HorizontalAlignment="Left" IsReadOnly="False" CanUserAddRows="False" FontSize="16px">

<DataGrid.Columns>

<DataGridTemplateColumn>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<StackPanel Orientation="Vertical" Width="500">

<StackPanel>

<TextBlock Initialized="TextBlock\_Initialized"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Initialized="TextBlock\_Initialized\_Cost" Margin="0 0 6px 0" />

</StackPanel>

<StackPanel Orientation="Horizontal">

<Button Name="BReg" Margin="0 30px 15px 0" Width="130px" Content="Добавить" Initialized="Button\_Initialized\_Red" />

</StackPanel>

</StackPanel>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</StackPanel>

</StackPanel>

</ScrollViewer>

</ScrollViewer>

</Grid>

</Page>

Программная часть разметки:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace ПанковаЕМ

{

/// <summary>

/// Логика взаимодействия для Admin.xaml

/// </summary>

public partial class Admin : Page

{

List<Книги> Book = Class.Book.Книги.ToList();

public Admin()

{

InitializeComponent();

DGServises.ItemsSource = Book;

}

int i = -1;

private void TextBlock\_Initialized(object sender, EventArgs e)

{

if (i < Book.Count)

{

i++;

TextBlock TB = (TextBlock)sender;

Книги S = Book[i];

TB.Text = S.Название;

//i++;

}

}

private void TextBlock\_Initialized\_Cost(object sender, EventArgs e)

{

if (i < Book.Count)

{

TextBlock Price = (TextBlock)sender;

Книги S = Book[i];

Price.Text = Convert.ToInt32(S.Цена) + "";

}

}

private void Button\_Initialized\_Red(object sender, EventArgs e)

{

Button BtnRed = (Button)sender;

if (BtnRed != null)

{

BtnRed.Uid = Convert.ToString(i);

}

}

}

}

Код класса базы данных:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ПанковаЕМ

{

class Class

{

public static Entities Book;

}

}

Программный код главного класса

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Controls;

namespace ПанковаЕМ

{

class F

{

public static Frame Mframe;

}

}

# Сводка

Диаграммы и графики, отображающие результаты тестирования. Данные должны основываться на данных, полученных