Одесский колледж компьютерных технологий «Сервер»

Пояснительная записка к техническому проекту на создание приложения «Maksimum\_Reader»

Выполнил студент группы К13.1

Рыбак Максим

Одесса 2016

**Введение**

Приложение книжная библиотека предназначена для выбора книг из библиотеки и их чтение. Также возможен поиск по библиотеке и добавление книг в общую библиотеку. В библиотеке читатель может отыскать книги различных жанров и эпох; выбор около 100 книг. В набор инструментов входит изменение фонового цвета, пометка текста, изменение размера шрифта.

Нам было дано задание на разработку программы для чтения книг, которая хранила бы в себе целую библиотеку книг. Поставленная задача была выполнена.

Наша команда состояла из:

* + Илья Лукашов.
  + Максим Рыбак.
  + Богдан Хмельницкий.
  + Игорь Тищенко.

Работа которую необходимо было реализовать: разработать простой и функциональный интерфейс, написать код программы, написание документации.

Выполнение задач

В мои обязанности входило разработка основного кода приложения внедрение и создания графической оболочки приложения, так же я помогал организовал работу команды создал переписку и распределил обязанности между командой. Ниже представлена моя работа. Часть функционала приложения было отдано на реализацию Ильи Лукашева.

Описание выполненной задачи со скриншотами

Мною было реализован основной интерфейс программы ниже представлен скриншот главного окна:

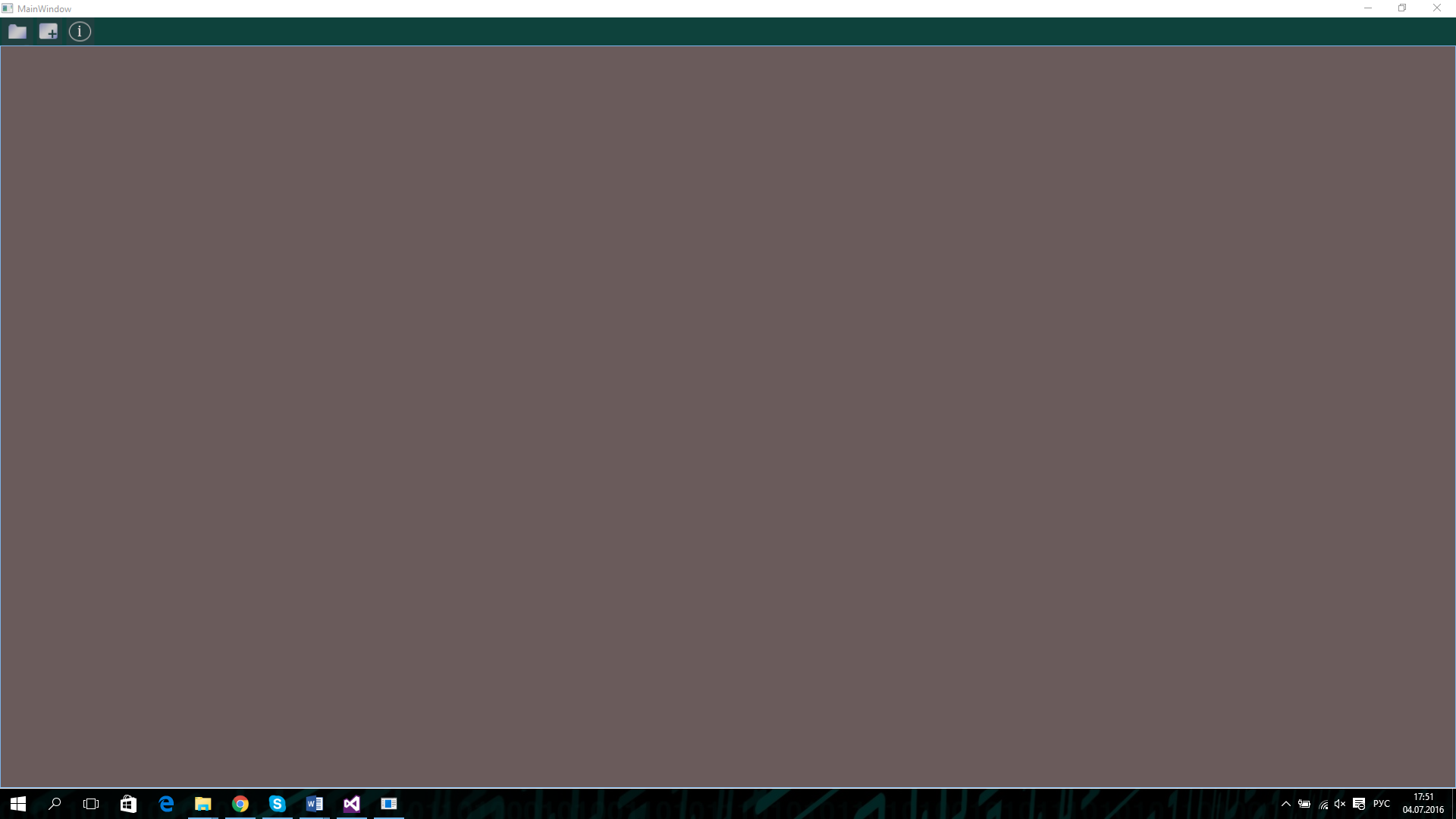


Рисунок 1 – Основное окно программы

Так же было выполнено просмотр всех книг в отдельной форме и выбор любой книги на чтение.

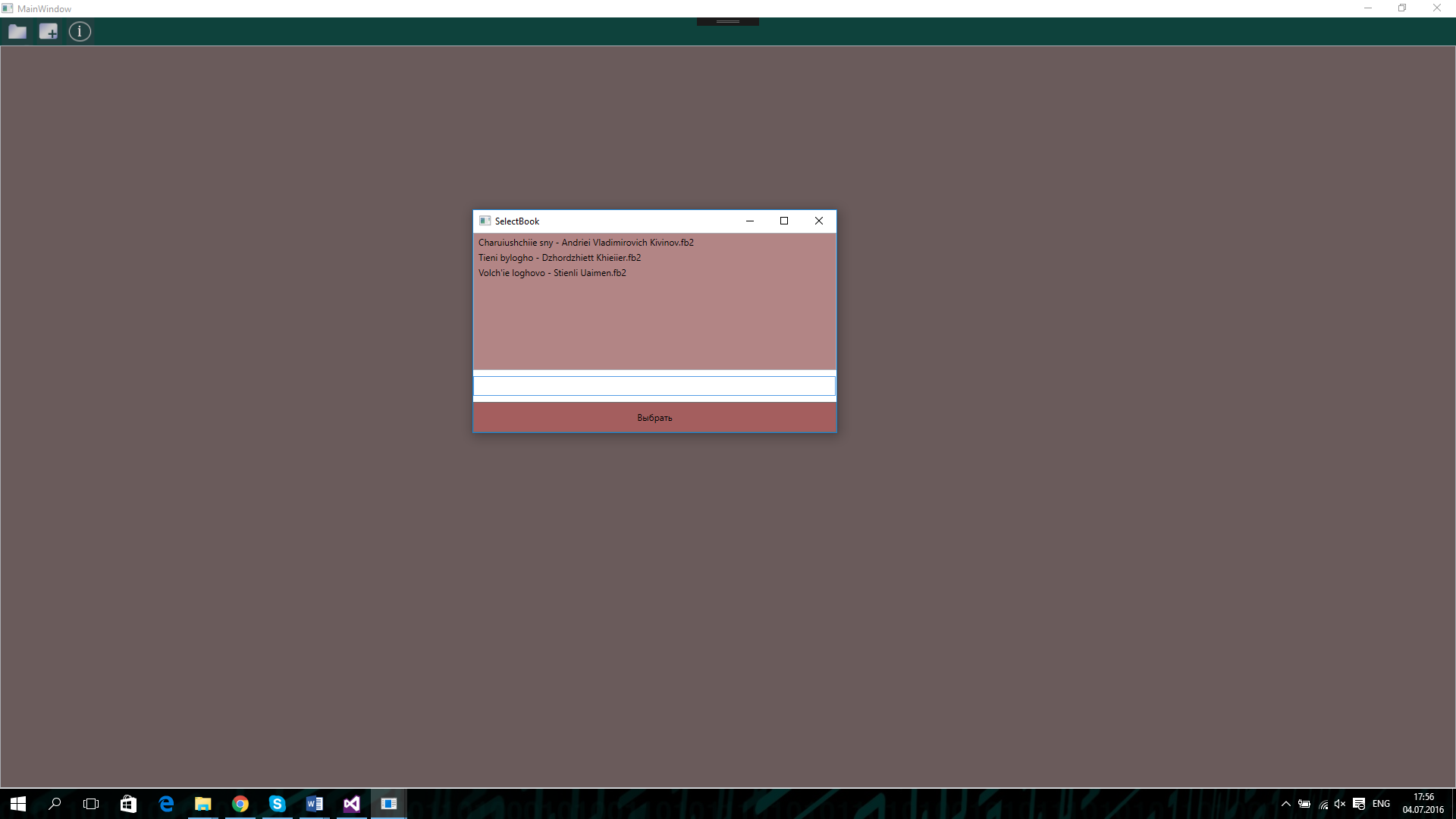


Рисунок 2 - Выбор книги из локального хранилища

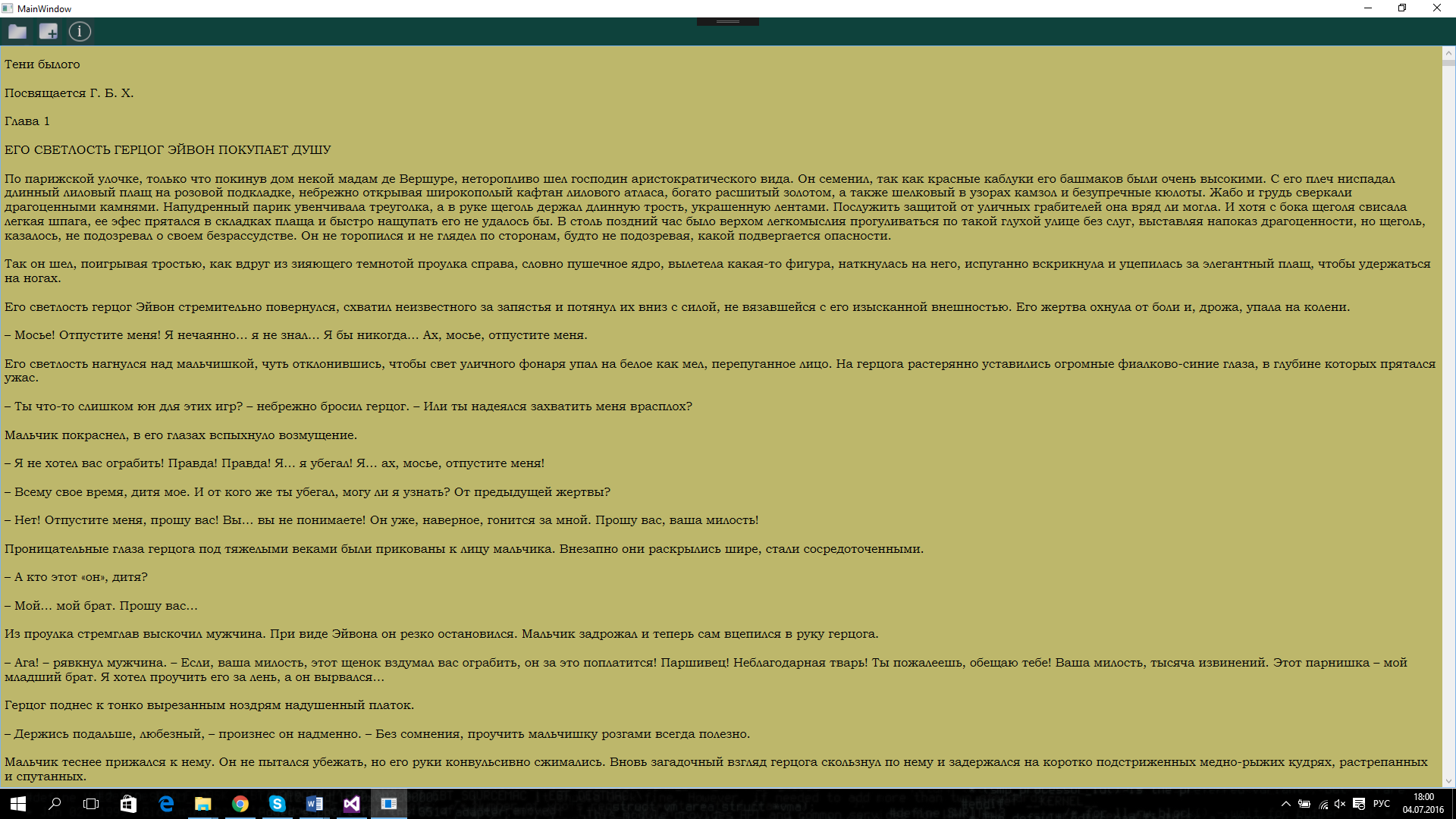


Рисунок 3 – Вывод на экран выбранной книги.

Реализована смена цветовой гаммы при желании пользователя. При выборе соответствующего пункта меню, открывается форма в которой пользователь может просмотреть как будет выглядеть применяемый стиль и его применить.

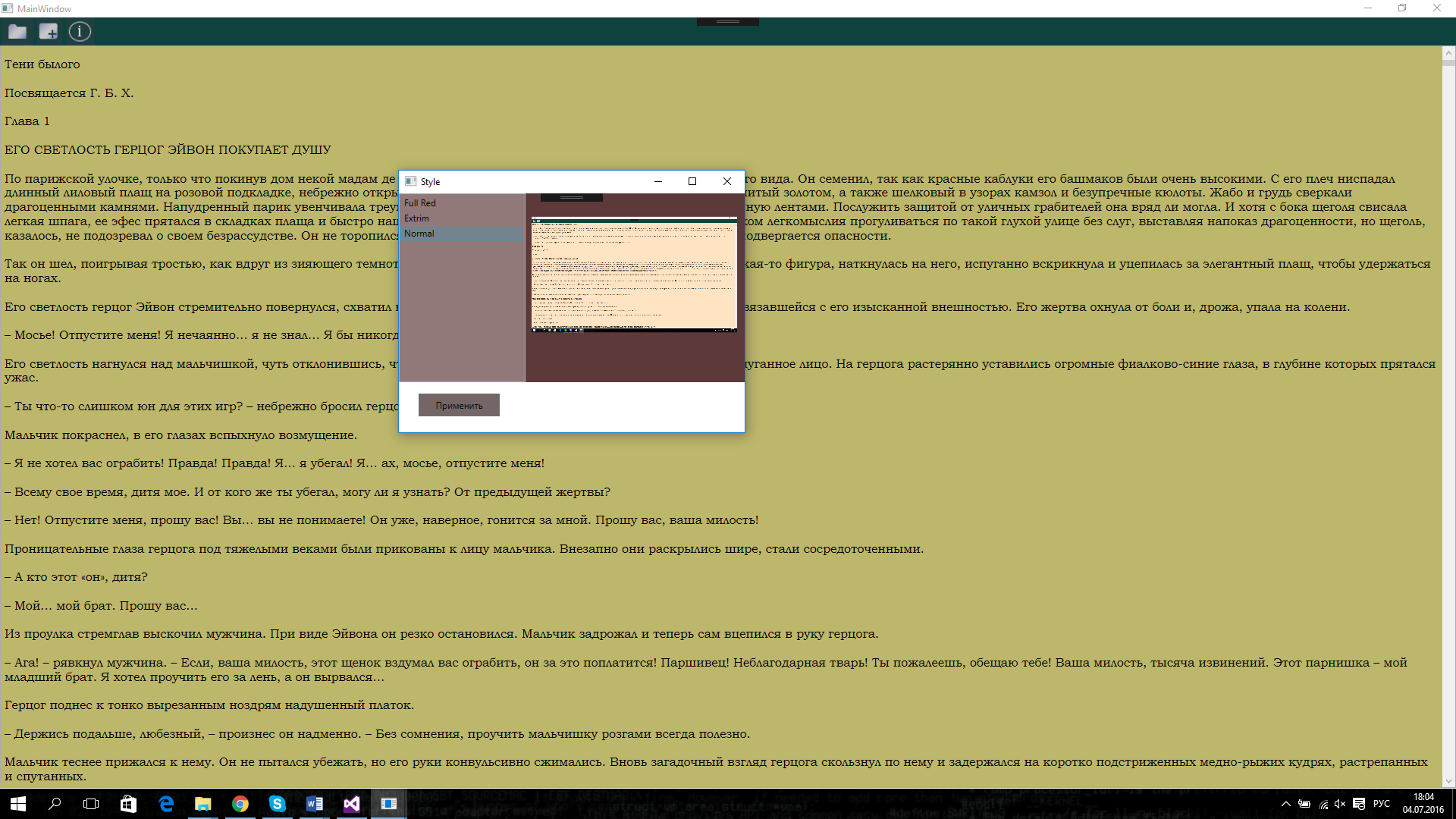


Рисунок 4 – Форма выбора оформления

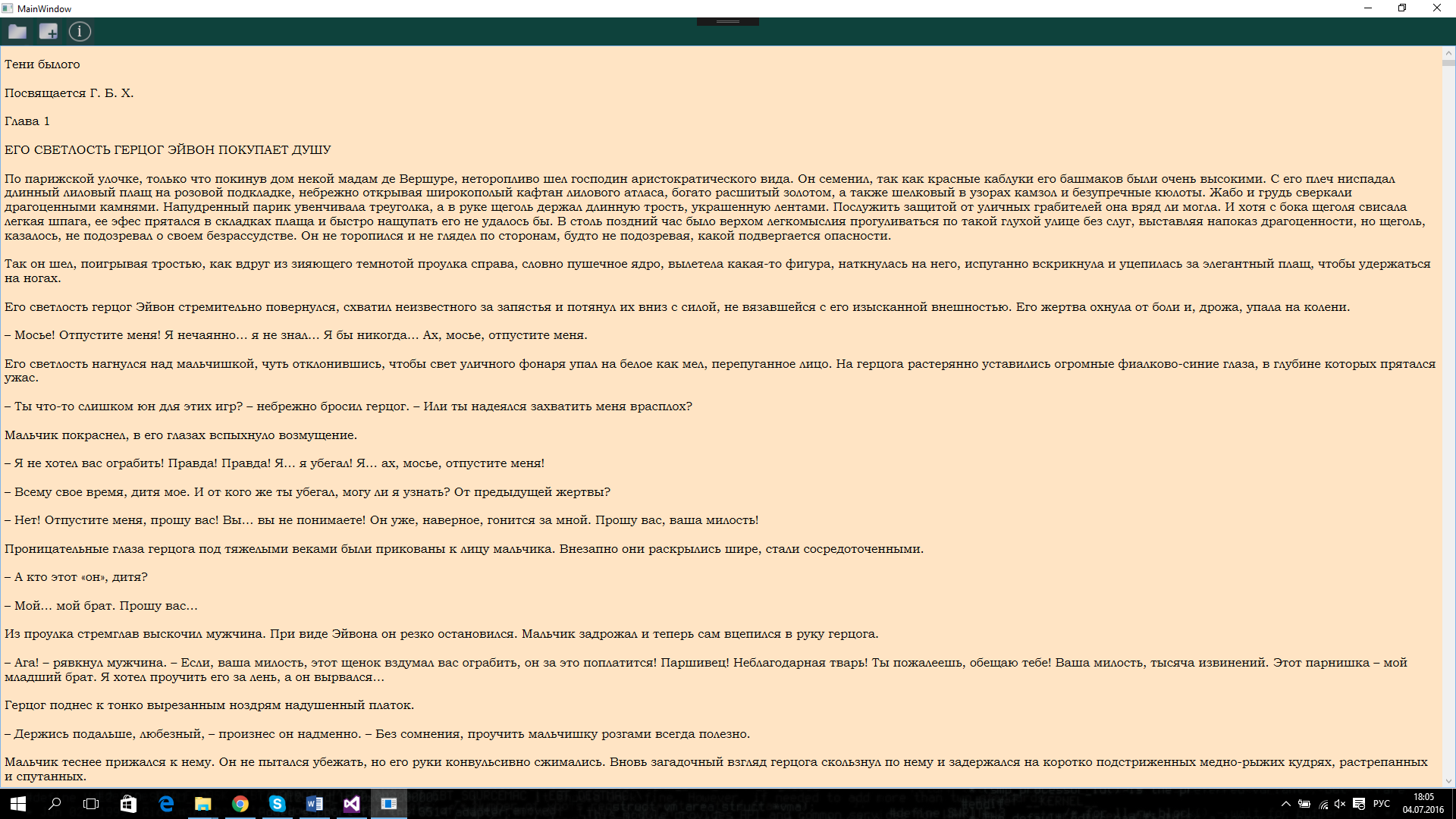


Рисунок 5 – Применение оформления

Ниже приведен листинг кода программы который мною был реализован

//Стиль Читалки

string str\_color = "Black";

public void Smena\_Color(string str\_color1)

{

str\_color = str\_color1;

//Проерка на обновление стиля////////////////////////////////////////////////////////////////

if (flag\_update\_style)

this.Update\_style(flag\_update\_style, update\_style);

}

//////////////////////////////Обновление стиля //////////////////////////////////////////////////////

public void Update\_style(bool flag,string str)

{

richTextBox.Document.Blocks.Clear();

this.Style\_Redaer(str,str\_color);

}

/////////////////////////////////////////////////////////////////////////////////////////////////////

public void Style\_Redaer(string str, string Color)

{

TextRange tr = new TextRange(richTextBox.Document.ContentEnd, richTextBox.Document.ContentEnd);

tr.Text = "textToColorize";

if (Color == "Red")

{ tr.ApplyPropertyValue(TextElement.ForegroundProperty, Brushes.Red); richTextBox.Background = new SolidColorBrush(Colors.Black); }

if (Color == "Black")

{ tr.ApplyPropertyValue(TextElement.ForegroundProperty, Brushes.Black); richTextBox.Background = new SolidColorBrush(Colors.DarkKhaki); }

if (Color== "Normal")

{ tr.ApplyPropertyValue(TextElement.ForegroundProperty, Brushes.Black); richTextBox.Background = new SolidColorBrush(Colors.Bisque); }

richTextBox.AppendText(str);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////

public void Show\_book(bool flag,int index)

{

if (flag)

{

MessageBox.Show("Подождите пока загрузится книга");

richTextBox.Document.Blocks.Clear();

var xmlDoc = new XmlDocument();

string file = paths[index];

////////////////////ДЛя добавления в Историю//////////////////////////////////////////////////

flag\_History = true;

file\_dir= paths[index];

this.Favorites(flag\_Favorites,flag\_History,paths[index]);

/////////////////////////////////////////////////////////////////////////////////////////////

xmlDoc.Load(file);

XmlReader Reader = XmlReader.Create(file);

string str = "";

while (Reader.Read())

{

str = str + Reader.Value;

}

////////////////////////////////////////////////////////////////////////////////////////////////

this.Style\_Redaer(str, str\_color);

/////Для динамического обновления стиля ////////////////////////////////////////////////////////

flag\_update\_style = true;

update\_style = str;

}

}

public void MenuItem\_Click\_3(object sender, RoutedEventArgs e) {

SelectBook form = new SelectBook(this);

form.ShowDialog();//в модальном режиме ,то что надо :)

}

//Смена цветовой расскраски

private void MenuItem\_Click\_4(object sender, RoutedEventArgs e)

{

Style style = new Style(this);

style.ShowDialog();

}

//Открытие формы избранного///////////////////////////////////////////

private void MenuItem\_Click\_6(object sender, RoutedEventArgs e)

{

Favorites f = new Favorites(this);

f.ShowDialog();

}

Вывод книг из локального хранилища в форму:

string[] str;

int index;bool flag\_click = false;

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

str = Directory.GetFiles(@"book\", "\*.fb2");

for (int i = 0; i <= str.Length - 1; i++)listBox.Items.Add(str[i].Remove(0, 5));

}

public void button\_Click(object sender, RoutedEventArgs e)

{flag\_click = true;m\_parent.Show\_book(flag\_click,index);}

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

{index = listBox.SelectedIndex;}

////////////////////////поиск////////////////////////////////////////////////////////

private void textBox\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{ string str\_find = textBox.Text; }