Содержание

1.Введение ...................................................................................................

2.Выбор технологии для реализации проекта ..........................................

2.1 Язык С# ..................................................................................................

2.2 WPF..........................................................................................................

2.3 Базы данных и Entity Framework...........................................................

3.Реализация функций..................................................................................

4.Вывод...........................................................................................................

5.Литература...................................................................................................

Введение

Проект The collection создан для ведения собственной коллекции просмотренных медиа продуктов или медиа продукты которые вы в будущем хотели просмотреть

В проект входит добавления элементы где вы указываете.

-Что это(фильм,книга,сериал и т д)

-Название медиа проекта

-Просмотрели вы его или нет

-Ссылку для просмотра этого продукта или для прочтения описания и отзывов об продукте

-Фото или обложку медиа проекта

Данный проект имеет место быть в современном мире так как практически каждый человек потребляет медиа продукты в том или ином виде каждый день.

Выбор технологий для реализации проекта

**Язык C#**

C#.Данный язык использует объектно-ориентированный подход к программированию во всем. Это означает, что тебе нужно будет описывать абстрактные конструкции на основе предметной области, а потом реализовывать между ними взаимодействие. Данный подход пользуется большой популярностью, потому что позволяет не держать в голове всю информацию



Также в языке присутствует обилие синтаксического сахара, который делает тяжелую жизнь программиста капельку слаще. Вместо того, чтобы писать 100500 строк кода, ты просто используешь готовую конструкцию, а компилятор сделает за тебя всю грязную работу. Но некоторые такие конструкции являются не самыми оптимальными с точки зрения производительности. Но все это перекрывается за счет удобочитаемости кода и высокой скоростью разработки.

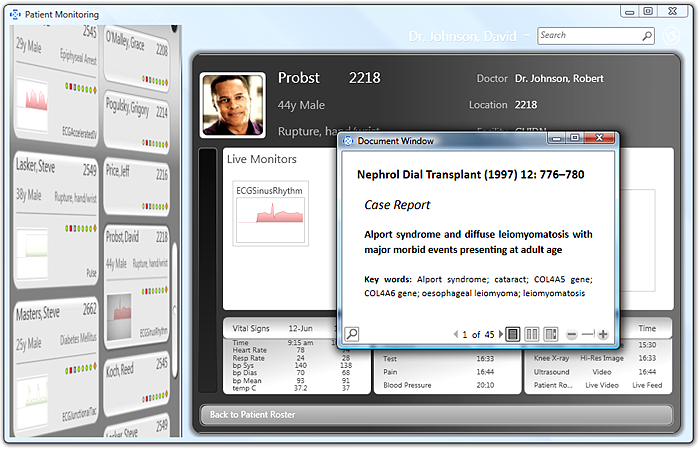
Еще стоит упомянуть, что все это работает на базе платформы .NET Framework. Что это означает? Для многих непосвященных, это просто какая-то приблуда, которую нужно установить на комп, чтобы программа запустилась, но дело обстоит значительно глубже. Написанный тобой код на языке C# транслируется в промежуточный язык (IL), который в свою очередь уже преобразуется в машинный код на твоем компьютере прямо во время выполнения приложения (JIT). Спрашивается, зачем это все? А суть в том, что ты можешь пилить со своим другом Васей на разных языках один и тот же проект и ни одному из вас не придется переучиваться. Но я никогда не видел, чтобы это реально использовали на практике. Но это еще не все. Так как окончательная компиляция из промежуточного кода выполняется в живую на твоей конкретной машине, то возможно увеличение производительности за счет использования специфических команд именно твоего процессора.

Основная идея C# — это универсальность. Ты можешь реализовать все на одном языке и для любой платформы, но с большой долей вероятности это будет работать медленнее, чем разработанное специально под конкретную платформу.

Он неплохо подходит как для начинающих в качестве первого языка, так и для реализации крупных коммерческих проектов.

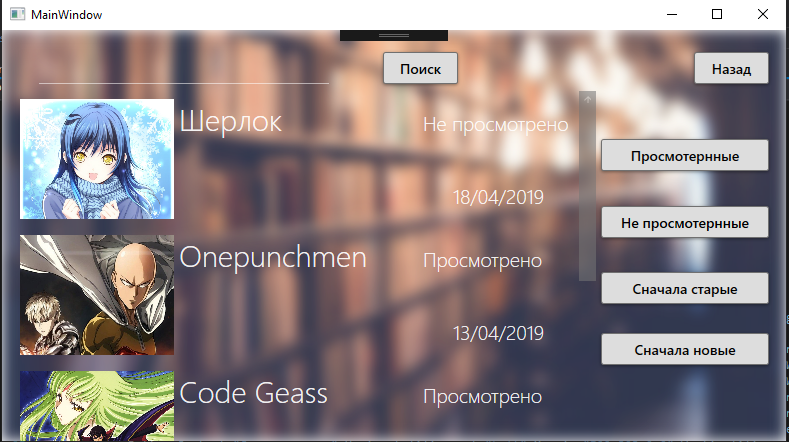
**WPF**

WPF. Windows Presentation Foundation (WPF) ― это система следующего поколения для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем. С помощью WPF можно создавать широкий спектр как автономных, так и размещенных в браузере приложений. На следующем рисунке показан пример одного из таких приложений



В основе WPF лежит векторная система визуализации, не зависящая от разрешения и созданная с расчетом на возможности современного графического оборудования. WPF расширяет базовую систему полным набором функций разработки приложений, в том числе Extensible Application Markup Language (XAML), элементами управления, привязкой данных, макетом, 2-D- и 3-D-графикой, анимацией, стилями, шаблонами, документами, мультимедиа, текстом и оформлением. WPF входит в состав Microsoft .NET Framework и позволяет создавать приложения, включающие другие элементы библиотеки классов .NET Framework. WPF является частью экосистемы .NET и развивается вместе с фреймворком .NET и имеет те же версии. Первая версия WPF 3.0 вышла вместе с .NET 3.0 и операционной системой Windows Vista в 2006 году. С тех пор платформа последовательно развивается. Последняя версия WPF 4.6 вышла параллельно с .NET 4.6 в июле 2015 года, ознаменовав девятилетие данной платформы.

Вот пример интерфейса wpf из моей программы:



Базы данных и Entity Framework

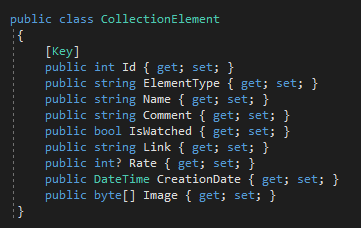
Для создания баз данных я использовал EntityFramework на основе СУБД MySQL.

Entity Framework был представлен корпорацией Microsoft в 2008 году как основное средством взаимодействия между приложениями .NET и реляционными базами данных. Entity Framework — это инструмент, упрощающий сопоставление объектов в программном обеспечении с таблицами и столбцами реляционной базы данных.

База данных – совокупность взаимосвязанных данных, которые можно использовать для большого числа приложений, быстро получать и модифицировать необходимую информацию.

Модели базы данных базируются на современном подходе к обработке информации. Структура информации базы позволяет формировать логические записи их элементов и их взаимосвязи. Взаимосвязи могут быть: один к одному, один ко многои и многие ко многим.

В моей базе данных хранилась только одна таблица с данными столбцами:



**Реализация функций**.

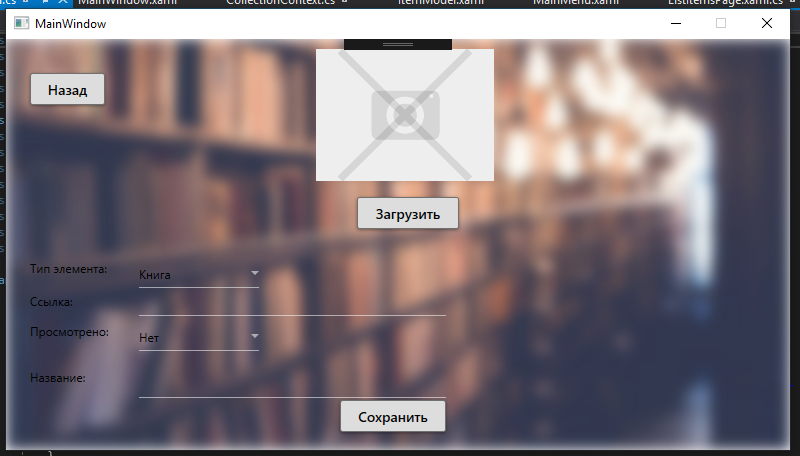
Сейчас я хотел бы показать куски кода где я реализую те или иные функции которые выполняет мое приложение.

Для начала навигация в моем приложении происходит по одному окну. Это означает что у меня задействовано только одно окно где я уже меняю только контент окна страницами вот пример:



Здесь я указываю у окна что я создаю класс новой страницы и заменю свойство Content и также указываю что передаю окно в конструктор страницы чтобы я мог изменять контент окна уже оттуда.

Сохранение элементов при их создании происходит при нажатии на кнопку сохранить:



При нажатии этой кнопке я делаю обращение к базе данных через специальную библиотеку Entity framework и сохраняю все что ввел пользователь в базу данных

А также делаю проверки на неправильность заполнения:

private void SaveElementClick(object sender, RoutedEventArgs e)

{

if (comboBoxType.SelectedItem.ToString() == "Другое..." && OtherTextBox.Text == String.Empty)

{

MessageBox.Show("Вы ввели данные неверно");

}

else

{

if (nameText.Text == String.Empty)

{

MessageBox.Show("Вы ввели данные неверно");

}

else

{

using (var context = new CollectionContext())

{

CollectionElement element = new CollectionElement();

element.CreationDate = DateTime.Now;

element.Image = File.ReadAllBytes(filePath);

element.Name = nameText.Text;

if (comboBoxIsWatched.SelectedItem.ToString() == "Да")

{

element.IsWatched = true;

}

else

{

element.IsWatched = false;

}

if (LinkText.Text == String.Empty)

{

element.Link = "-";

}

else

{

element.Link = LinkText.Text;

}

if (RateComboBox.SelectedItem == null)

{

element.Rate = null;

}

else

{

int kek;

if (int.TryParse(RateComboBox.SelectedItem.ToString(), out kek))

{

element.Rate = kek;

}

}

if (comboBoxType.SelectedItem.ToString() == "Другое...")

{

element.ElementType = OtherTextBox.Text;

}

else

{

element.ElementType = comboBoxType.SelectedItem.ToString();

}

context.Elements.Add(element);

context.SaveChanges();

}

MessageBox.Show("Добавлено!");

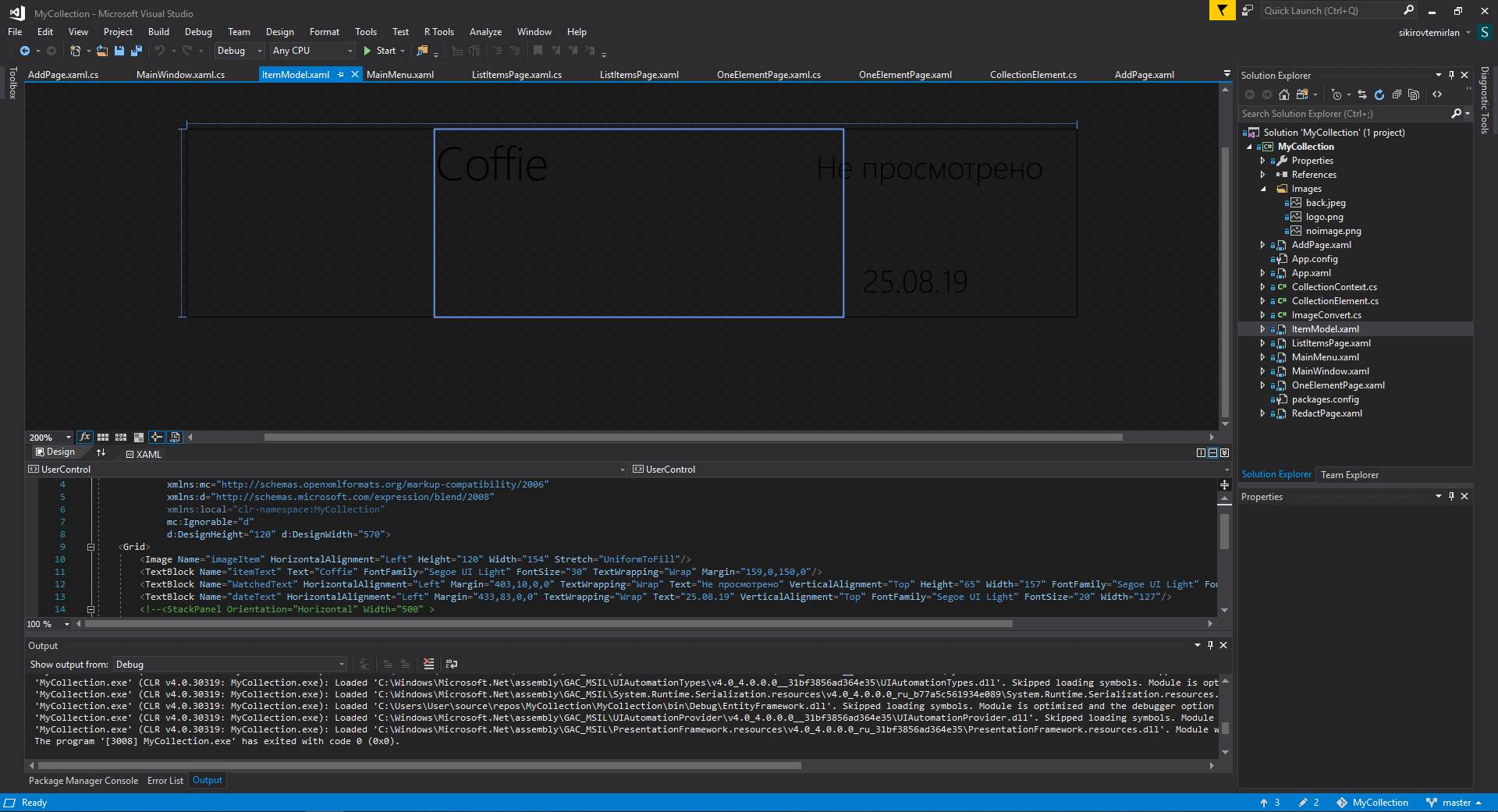
window.Content = new MainMenu(window);

}

}

}

На странице где отображаются все элементы для их демонстрации я использовал элемент из WPF ListBox.В него можно положить любой элемента типа object а это почти все элементы в c#.Этой особенность я и воспользовался отдавая ListBox-у свой собственный элемент управления это тоже одна из функций WPF.



Я создал небольшой элемент управления который состоит из картинки,названия,даты и информации был ли просмотрен медиа элемент коллекции или нет.И именно этими элементами и заполняется мой ListBox.

Вывод

Я постарался использовать в своем проекте все знания по программированию которые я получил в университете и изучил дополнительные технологии для реализации некоторых функций.

Мне было интересно заниматься этим и надеюсь что в будущем буду делать более обширные проекты.

Литература

C# - <https://shwanoff.ru/plus-minus-c-sharp/amp/>

WPF - <https://ru.wikipedia.org/wiki/Windows_Presentation_Foundation>

Entity Framework - <https://www.internet-technologies.ru/articles/chto-takoe-entity-framework.html>