|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)»**  **(ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)»)** |
|  |
| **УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ** |

|  |
| --- |
| **КУРСОВОЙ ПРОЕКТ** |
|  |
| по междисциплинарному курсу: МДК.02.02. Технология разработки и защиты баз данных. |
|  |
| на тему: Разработка информационной системы для любителя видеоигр. |
|  |
| студентки группы ПКС-404  специальности 09.02.03 Программирование в компьютерных системах |
| Беляевой Ангелины Вячеславовны |

|  |  |  |
| --- | --- | --- |
| Студент |  | А.В. Беляева |
| Руководитель курсового проекта |  | Е.А. Ларионова |
| Председатель ПЦК специальности 09.02.03 Программирование в компьютерных системах |  | А.И. Глускер |

|  |  |  |
| --- | --- | --- |
| Дата защиты «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2021 г. | | |
| Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Заведующий отделением № 1 |  | И. А. Миланова |

Москва

2021

**Оглавление**

[ВВЕДЕНИЕ 4](#_heading=h.4f1mdlm)

[1.](#_heading=h.2u6wntf) ОБЩАЯ ЧАСТЬ 6

[1.1.](#_heading=h.19c6y18) Назначение и цели создания системы 6

[1.1.1.](#_heading=h.3tbugp1) Объект исследования 6

[1.1.2.](#_heading=h.28h4qwu) Предметная область 6

[1.1.3.](#_heading=h.nmf14n) Основная цель проекта 6

[1.2.](#_heading=h.37m2jsg) Обзор и анализ предметной области 6

[1.3.](#_heading=h.1mrcu09) Жизненный цикл базы данных 6

[1.4.](#_heading=h.46r0co2) Выбор и характеристика СУБД 7

[1.5.](#_heading=h.2lwamvv) Выбор и характеристика среды разработки приложения 7

[2.](#_heading=h.111kx3o) СПЕЦИАЛЬНАЯ ЧАСТЬ 9

[2.1.](#_heading=h.3l18frh) Постановка задачи 9

[2.2.](#_heading=h.206ipza) Архитектура информационной системы 9

[2.3.](#_heading=h.4k668n3) Логическая модель базы данных 10

[2.4.](#_heading=h.2zbgiuw) Нормализация отношений 11

[2.5.](#_heading=h.1egqt2p) Описание таблиц 11

[2.6.](#_heading=h.3ygebqi) Другие объекты базы данных (представления, хранимые процедуры, триггеры, имена входа, пользователи) 12

[2.7.](#_heading=h.2dlolyb) Разработка приложения 12

[2.7.1.](#_heading=h.sqyw64) Диаграмма вариантов использования 13

[2.7.2.](#_heading=h.3cqmetx) Решение главной задачи проекта 13

[2.7.3.](#_heading=h.4bvk7pj) Тестирование приложения 17

[2.7.4.](#_heading=h.2r0uhxc) Защита информационной системы 17

[2.8.](#_heading=h.1664s55) Инструкция пользователю 18

[2.8.1.](#_heading=h.3q5sasy) Общие сведения об информационной системе 18

[2.8.2.](#_heading=h.25b2l0r) Требования к техническим средствам 18

[2.8.3.](#_heading=h.kgcv8k) Требования к программным средствам 19

[2.8.4.](#_heading=h.34g0dwd) Настройка информационной системы 19

[ЗАКЛЮЧЕНИЕ 20](#_heading=h.1jlao46)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 21](#_heading=h.43ky6rz)

[ПРИЛОЖЕНИЕ А 22](#_heading=h.2iq8gzs)

[Приложение Б 32](#_heading=h.xvir7l)

# ВВЕДЕНИЕ

Курсовой проект посвящен одной из частей современной культуры, а точнее – видеоиграм. Игровая индустрия в наше время является часто обсуждаемой темой и ей посвящены некоторые мероприятия, на которых обсуждают предстоящие новинки. Такими мероприятиями являются, например, известное в мире «Electronic Entertainment Expo» (E3) и Российское мероприятие «Игромир». Однако многие стали забывать о тех играх, которые выходили раньше и тех, которым уже десятки лет.

Основная цель проекта – создание видеоигровой библиотеки с подключением базы данных с SQL.

Краткий анализ предметной области

Анализ существующих систем, связанных с моей темой, показал, что информационная система имеет клиент-серверную архитектуру и может состоять из следующих компонентов:

* база данных,
* бэкенд (логика),
* фронтенд (клиентская часть),

Обзор существующих приложений в рассматриваемой предметной области

В курсовом проекте рассматривались следующие «Библиотеки игр»:

1. Steam – магазин игровой дистрибуции с возможностью покупки игр (не входит в основную цель проекта). Известный онлайн магазин с внушительной библиотекой игр и красивым интуитивно понятным дизайном.
2. Epic Games Store – такой же магазин игровой дистрибуции, появившийся в 2019 году и став популярным за счёт бесплатных раздач игр, среди который были и крупные проекты.

Анализ этих приложений показал, как можно с помощью интуитивно понятного интерфейса представлять игры различных жанров.

Задачи проекта

1. Анализ предметной области;
2. Проектирование базы данных средствами SQL;
3. Разработка приложения средствами Visual Studio 2019;
4. Защита базы данных и приложения;

Возможности приложения

1. Возможность переходить на интересующие разделы (игры, жанры, студии, платформы)
2. Возможность оценивать игры по 5 балльной шкале

# ОБЩАЯ ЧАСТЬ

## Назначение и цели создания системы

## Объект исследования

Информационная система для любителя видеоигр

## Предметная область

Настольное приложение игровой библиотеки с подключением к базе данных MS SQL SERVER

## Основная цель проекта

Создание настольного приложения игровой библиотеки с подключением к базе данных MS SQL SERVER

## Обзор и анализ предметной области

База данных создаётся для игроков компьютерных игр с желанием изучить те игры, которые выходили в различные периоды, либо иных жанров. БД должна содержать данные о играх, разработчиках, издателях, датах релиза, жанрах, ценах, статусах, режимах, информации о пользователях (Логин и Пароль) и статусе аккаунта.

## Жизненный цикл базы данных

Жизненный цикл всегда состоит из таких аспектов как:

1. Анализ. Самая быстрая часть, Данный этап занял 3 дня;
2. Проектирование. Было сложно придумать логическую модель и придумать то, как будет выглядеть само приложение. Данный этап 60 дней;
3. Реализация. Данный этап длился долго и с углублением возможностей WPF и C# с использованием методов Entity Framework был выбран по причине того, лично мне удобнее работать с таблицами БД SQL, как с классами C#. Заняло 2 месяца.;
4. Тестирование. Этап проходил параллельно с этапом реализации по причине того, что были реализованы проверки после добавления кода. Этап длительностью также 60 дней.

## Выбор и характеристика СУБД

Для реализации курсового проекта был выбран Microsoft SQL Server по приниче того, что в нём присутствуют мощные средства разработки и возможность хранить большой объём данных.

«Microsoft SQL Server — система управления реляционными базами данных, разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.» [1]

## Выбор и характеристика среды разработки приложения

Для реализации приложения была выбрана среда разработки Visual Studio 2019, потому что является:

1. Одной из известных сред разработки,
2. Имеет возможность добавлять программные пакеты
3. На нём имеется возможность реализовывать кроссплатформенность
4. На нём можно реализовать множество программ различной степени сложности

«Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.» [2]

# СПЕЦИАЛЬНАЯ ЧАСТЬ

## Постановка задачи

За счёт использования базы данных, разработанной с помощью MS SQL Server, и средств разработки Visual Studio с использованием языка программирования C# реализовать приложение, которое бы позволяло:

1. Вывод игр;
2. Добавление, редактирование и удаление;
3. Поиск и фильтрация в динамическом режиме;
4. Поддерживать базу данных в актуальном состоянии.

## Архитектура информационной системы

Архитектурой данного приложения является клиент-серверная архитектура, вид которой реализован на Рисунок 1 - архитектура клиент-сервер.

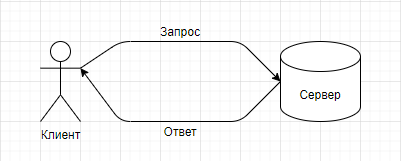


Рисунок 1 - архитектура клиент-сервер

«Клиент — сервер» — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Фактически клиент и сервер — это программное обеспечение.

Преимущества:

* Отсутствие дублирования кода программы-сервера программами-клиентами,
* Так как все вычисления выполняются на сервере, то требования к компьютерам, на которых установлен клиент, снижаются,
* Все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов. На сервере проще организовать контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.

Недостатки:

* Неработоспособность сервера может сделать неработоспособной всю вычислительную сеть. Неработоспособным сервером следует считать сервер, производительности которого не хватает на обслуживание всех клиентов, а также сервер, находящийся на ремонте, профилактике и т. п.
* Поддержка работы данной системы требует отдельного специалиста — системного администратора.
* Высокая стоимость оборудования. [3]

## Логическая модель базы данных

Логическая модель содержит 6 таблиц содержание которых находится на рисунке «Логическая модель базы данных».

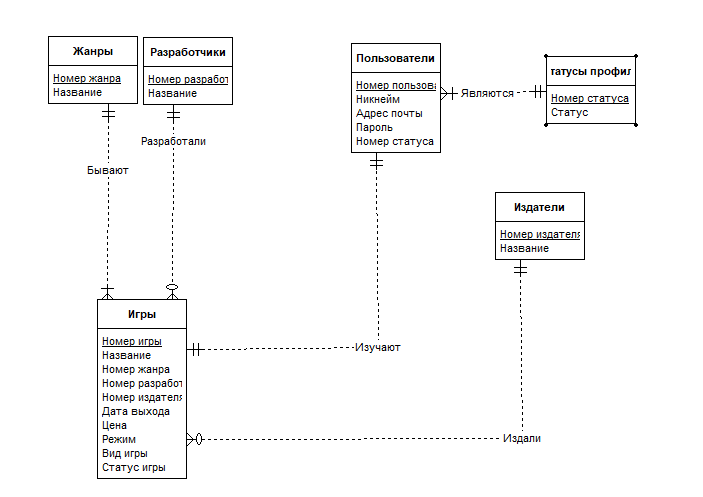


Рисунок 2 - Логическая модель базы данных

## Нормализация отношений

Все таблицы приведены к 3-й нормальной форме, т.е. все атрибуты атомарны, имеют взаимосвязь с первичным ключом и у первичного ключа отсутствует зависимость с не ключевыми атрибутами.

## Описание таблиц

Таблица 1 – Developer

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Описание** | **Тип** | **Примечание** |
| ID | Код записи | int | PK, Not NULL |
| GameID | Код игры | int | Not NULL |
| GameName | Название игры | nvarchar(100) | Not NULL |
| IDGenre | Код жанра | int | Not NULL |
| IDDeveloper | Код разработчика | int | NULL |
| IDPublisher | Код издателя | int | NULL |
| RealeseDate | Дата релиза | date | Not NULL |
| Price | Цена | money | Not NULL |
| Mode | Режим | nvarchar(50) | Not NULL |
| GameStatus | Статус игры | nvarchar(30) | Not NULL |
| ImagePriview | Изображение | varbinary(MAX) | NULL |

Таблица 2 – Developer

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Описание** | **Тип** | **Примечание** |
| IdDeveloper | Код разработчика | int | PK, Not NULL |
| DeveloperName | Название разработчика | nvarchar(50) | NULL |

Таблица 3 – GameImage

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Описание** | **Тип** | **Примечание** |
| ImageID | Код картинки | int | PK, Not NULL |
| GameID | Код игры | int | Not NULL |
| ImageSourse | Изображение | varbinary(MAX) | NULL |

Таблица 4 – Genre

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Описание** | **Тип** | **Примечание** |
| IDGenre | Код жанра | int | PK,Not NULL |
| GenreName | Название жанра | nvarchar(50) | PK, Not NULL |

Таблица 5 – Publisher

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Описание** | **Тип** | **Примечание** |
| IdPublisher | Код издателя | int | PK, Not NULL |
| PublisherName | Название издателя | nvarchar(50) | NULL |

Таблица 6 – User

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Описание** | **Тип** | **Примечание** |
| UserID | Код пользователя | int | PK, Not NULL |
| Nickname | Логин | varchar(30) | NULL |
| Password | Пароль | varchar(50) | NULL |
| Email | Электрон. почта | nvarchar(50) | NULL |
| UserStatus | Статус аккаунта | int | Not NULL |

Таблица 7 – UserStatus

|  |  |  |  |
| --- | --- | --- | --- |
| **Название столбца** | **Описание** | **Тип** | **Примечание** |
| ID | Код статуса | int | PK, Not NULL |
| Status | Статус | int | Not NULL |
| UserID | Код пользователя | int | Not NULL |

## Другие объекты базы данных (представления, хранимые процедуры, триггеры, имена входа, пользователи)

Отсутствуют, функции объектов БД были реализованы в C#

## Разработка приложения

Приложение с названием InGame разрабатывалось с учётом на долгосрочный срок, ибо многие люди, которые увлекаются играми, рано или поздно устают от привычных им игр и появляется желание найти что-то новое для себя.

## Диаграмма вариантов использования

Диаграмма взаимодействия реализована на рисунке «Диаграмма вариантов использования».

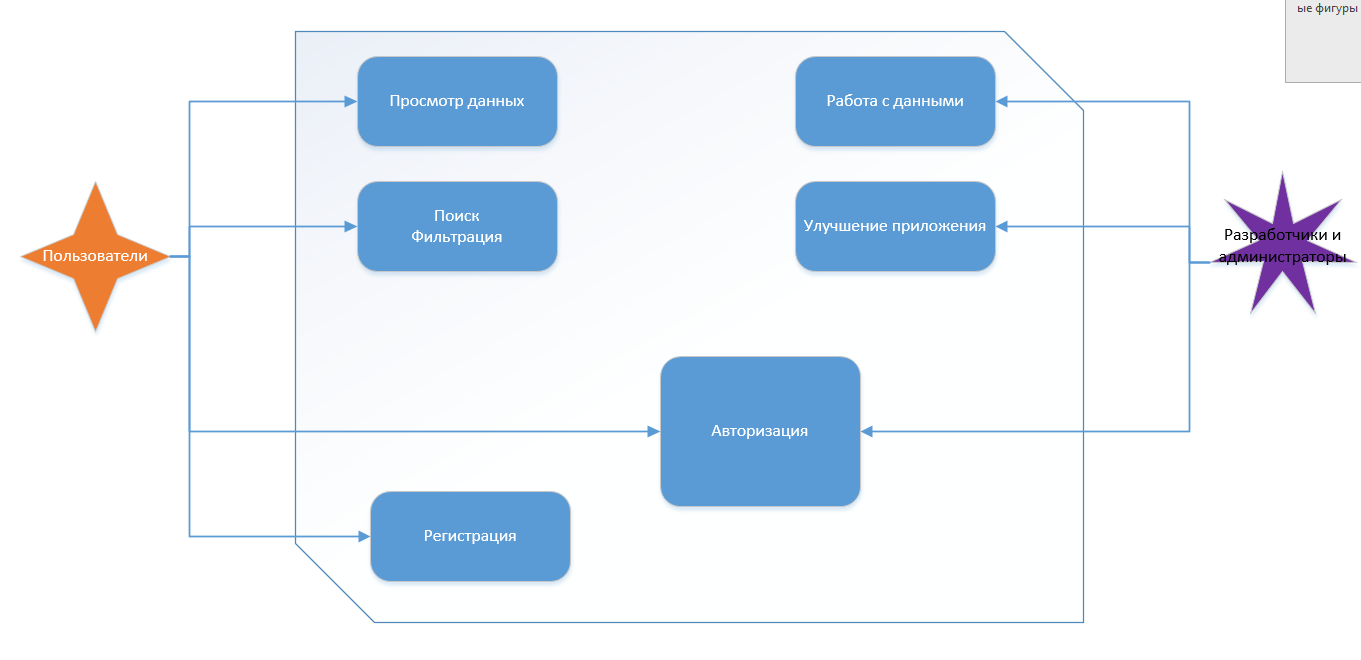


Рисунок 3 - Диаграмма вариантов использования

## Решение главной задачи проекта

Главная задача проекта – динамичные поиск и фильтрация данных.

Схематичное решение данной задачи изображено на рисунке «Схема реализации главной задачи».

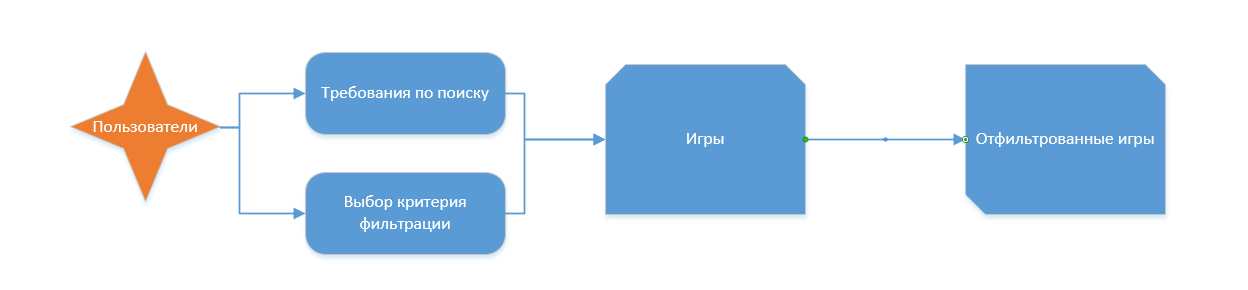


Рисунок 4 - Схема реализации главной задачи

Нас интересует только левая часть схемы. Ее код на C# выглядит следующим способом:

public GamesPagee()

{

InitializeComponent();

var allgenre = Games2Entities.GetContext().Genre.ToList();

allgenre.Insert(0, new Genre

{

GenreName = "Все жанры"

});

ComboGenre.ItemsSource = allgenre;

ComboGenre.SelectedIndex = 0;

}

private void UpdateGames()

{

var currentGames = Games2Entities.GetContext().Game.ToList();

if (ComboGenre.SelectedIndex > 0)

currentGames = currentGames.Where(p => p.Genre==ComboGenre.SelectedItem).ToList();

currentGames = currentGames.Where(p => p.GameName.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();

LViewGames.ItemsSource = currentGames.OrderBy(p => p.GameName).ToList();

}

private void TBoxSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

UpdateGames();

}

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

{

UpdateGames();

}

Вид окна показан на рисунке «Вид окна от лица пользователя, который не задал критерии поиска и фильтрации»

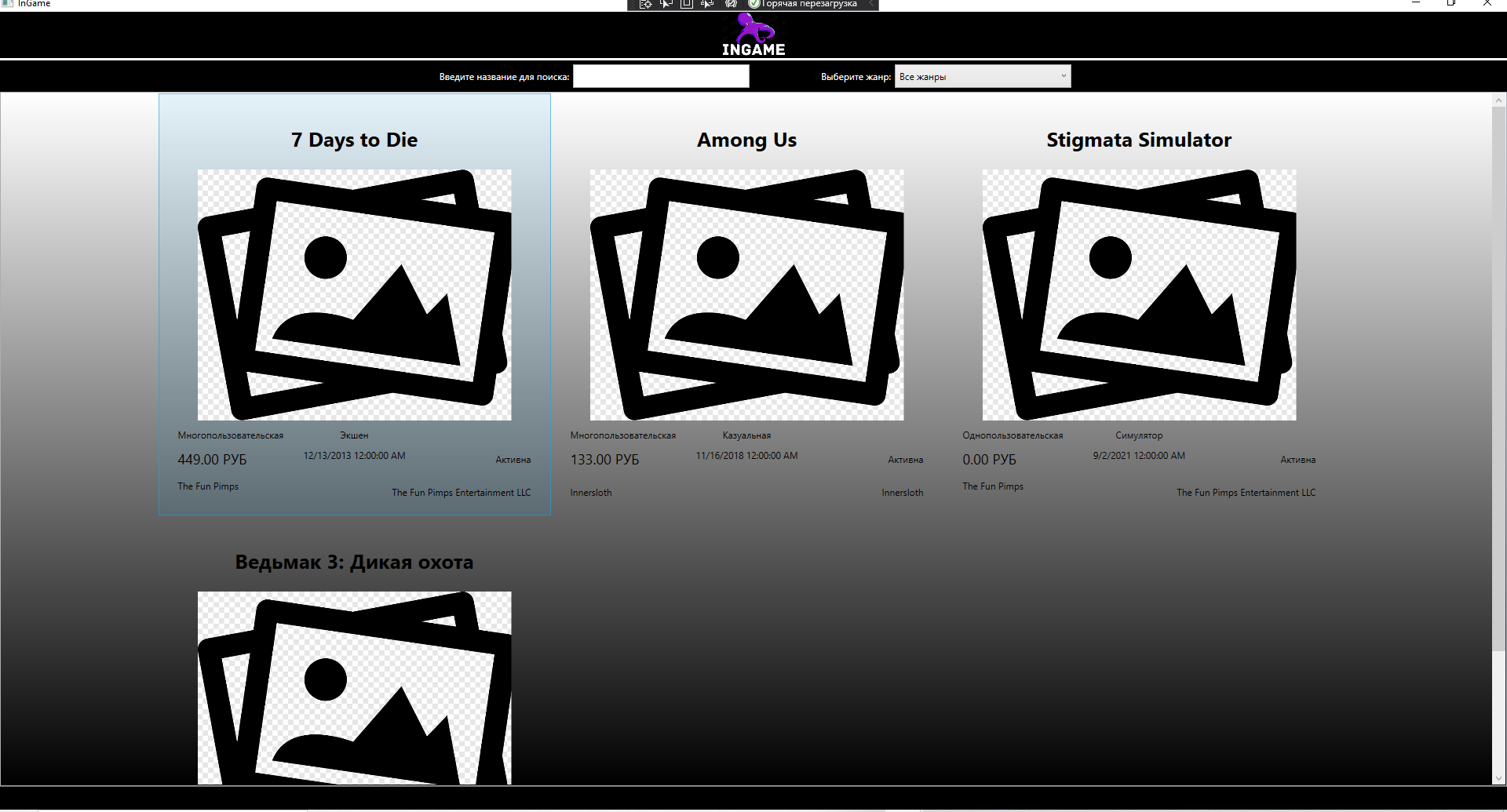


Рисунок 5 - Вид окна от лица пользователя, который не задал критерии поиска и фильтрации

Результат того, что произошло после того, как пользователь настроил критерии поиска и фильтрации, изображен на рисунке «Результат после того, как пользователь задал критерии поиска и фильтрации».

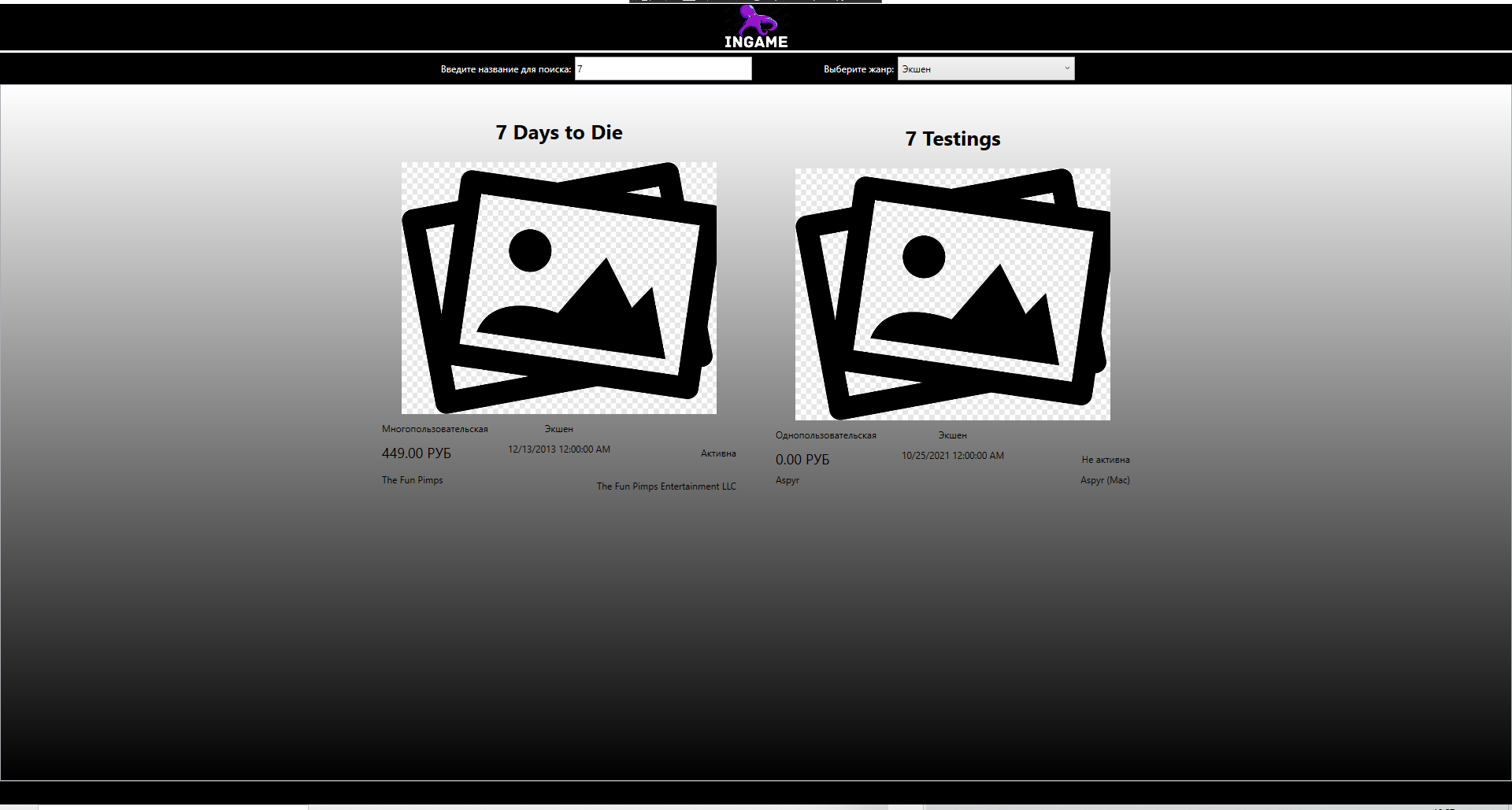


Рисунок 6 - Результат после того, как пользователь задал критерии поиска и фильтрации.

## Тестирование приложения

Таблица 7 – Тестирование приложения

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действие | Ожидаемый результат | Итоговый результат |
| 1 | Ввод верного логина и неверного пароля | Сообщение с предупреждением о некорректности ввода логина или пароля | Сообщение с предупреждением о некорректности ввода логина или пароля |
| 2 | Проверка открытия верного окна в зависимости от статуса аккаунта | При статусе 1 или 0 - вывод окна разработчика  При статусе 2 вывод окна пользователя | При статусе 1 или 0 - вывод окна разработчика  При статусе 2 вывод окна пользователя |
| 3 | Увеличение окна | Окно расширится на полный размер | Окно расширилось на полный размер |
| 4 | Сворачивание окна | Окно свернется | Окно свернулось |

## Защита информационной системы

Для защиты было использована аутенфитиикация через логин и пароль, однако так же реализована возможность регистрации, которая также хорошо защищена. Остальные сегменты защищены от возможных вылетов, ибо приложение прочно связано с базой данных.

## Инструкция пользователю

Пользователю при запуске программы встретится то, что изображено на рисунке «Графическое изображение работы InGame»

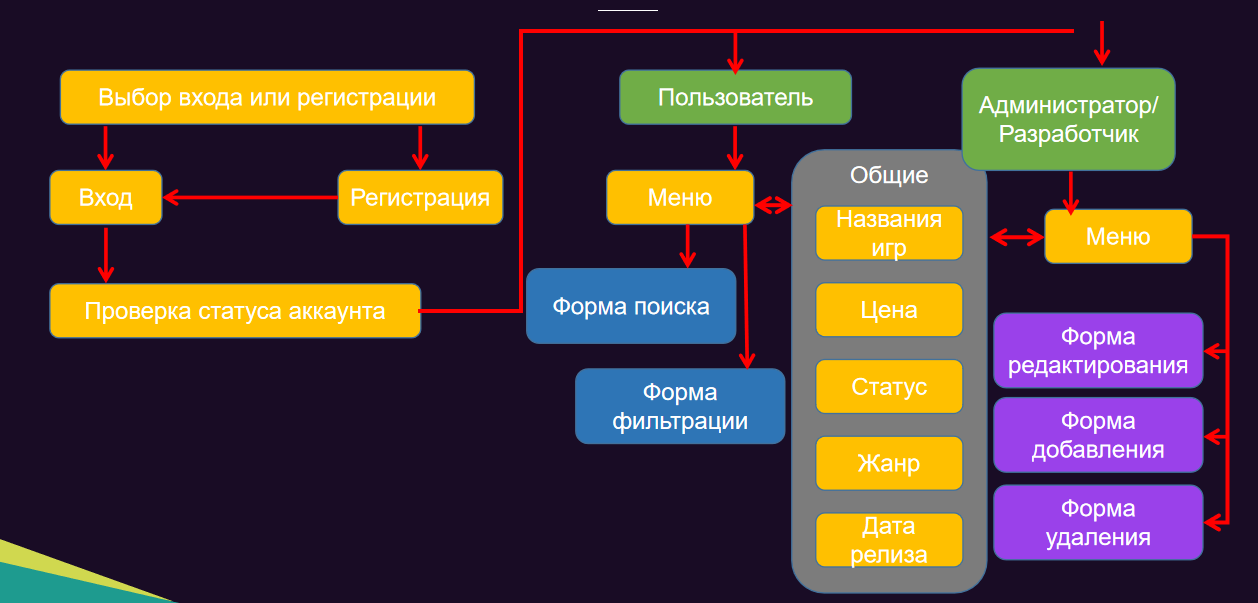


Рисунок 7 - Графическое изображение работы InGame

## Общие сведения об информационной системе

InGame представляет собой библиотеку, которая содержит информацию об играх, разработчиках, издателях, датах релиза, жанрах, ценах, статусах, режимах, информации о пользователях (Логин и Пароль) и статусе аккаунта с возможностью фильтрации и поиска по библиотеке.

## Требования к техническим средствам

Для удобного пользования программой требуется компьютер со средней комплектацией, а именно:

* Материнская плата HUAWEI BOHK-WAX9X-PCB,
* Процессор AMD Ryzen 7 3700U with Radeon Vega Mobile Gfx 2.30 GHz,
* Память установленной оперативная память (RAM) 8,00 ГБ
* Видеокарта AMD Radeon™ RX Vega 10 Graphics

## Требования к программным средствам

Для того, чтобы программа успешно запускалась требуется только операционная система Windows 7 или выше.

## Настройка информационной системы

Для этого достаточно будет запустить exe файл, который будет прилагаться.

# ЗАКЛЮЧЕНИЕ

Курсовой проект посвящен одной из частей современной культуры, а точнее – видеоиграм.

В процессе выполнения курсовой работы:

1. Изучена предметная область;
2. Спроектирована база данных;
3. Разработано приложение InGame, которое представляет собой библиотеку видеоигр;
4. Осуществлена защита посредством языка программирования C# и методов Entity Framework;
5. Освоены методы разработки приложения с использованием функционала WPF, C# и Entity Framework.

Дальнейшее развитие темы курсового проекта предполагает возможность оценивания игр, добавления в избранное, страница профиля, а также добавления тех игр, которые еще не существуют в базе данных.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Microsoft SQL Server – википедия [электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Microsoft\_SQL\_Server (Дата обращения 15.05.2021)
2. Microsoft Visual Studio – википедия [электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Microsoft\_Visual\_Studio (Вход на данный ресурс произведён 15.05.2021)
3. Клиент-сервер – википедия [электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82\_%E2%80%94\_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80 (Вход на данный ресурс произведён 15.05.2021)

# ПРИЛОЖЕНИЕ А

USE [master]

GO

/\*\*\*\*\*\* Object: Database [Games2] Script Date: 25.10.2021 19:19:39 \*\*\*\*\*\*/

CREATE DATABASE [Games2]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'Games2', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\Games2.mdf' , SIZE = 8192KB , MAXSIZE = 102400KB , FILEGROWTH = 65536KB )

LOG ON

( NAME = N'Games2\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\Games2\_log.ldf' , SIZE = 8192KB , MAXSIZE = 102400KB , FILEGROWTH = 65536KB )

GO

ALTER DATABASE [Games2] SET COMPATIBILITY\_LEVEL = 140

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [Games2].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [Games2] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [Games2] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [Games2] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [Games2] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [Games2] SET ARITHABORT OFF

GO

ALTER DATABASE [Games2] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [Games2] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [Games2] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [Games2] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [Games2] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [Games2] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [Games2] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [Games2] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [Games2] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [Games2] SET DISABLE\_BROKER

GO

ALTER DATABASE [Games2] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [Games2] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [Games2] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [Games2] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [Games2] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [Games2] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [Games2] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [Games2] SET RECOVERY SIMPLE

GO

ALTER DATABASE [Games2] SET MULTI\_USER

GO

ALTER DATABASE [Games2] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [Games2] SET DB\_CHAINING OFF

GO

ALTER DATABASE [Games2] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [Games2] SET TARGET\_RECOVERY\_TIME = 60 SECONDS

GO

ALTER DATABASE [Games2] SET DELAYED\_DURABILITY = DISABLED

GO

ALTER DATABASE [Games2] SET QUERY\_STORE = OFF

GO

USE [Games2]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Developer] Script Date: 25.10.2021 19:19:40 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Developer](

[IdDeveloper] [int] IDENTITY(1,1) NOT NULL,

[DeveloperName] [nvarchar](50) NULL,

CONSTRAINT [PK\_Developer] PRIMARY KEY CLUSTERED

(

[IdDeveloper] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Game] Script Date: 25.10.2021 19:19:40 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Game](

[GameID] [int] NOT NULL,

[ID] [int] IDENTITY(1,1) NOT NULL,

[GameName] [nvarchar](50) NOT NULL,

[IDGenre] [int] NOT NULL,

[IDDeveloper] [int] NULL,

[IDPublisher] [int] NULL,

[ReleaseDate] [date] NOT NULL,

[Price] [money] NOT NULL,

[Mode] [nvarchar](50) NOT NULL,

[GameStatus] [nvarchar](30) NOT NULL,

[ImagePreview] [varbinary](max) NULL,

[Rating] [int] NULL,

CONSTRAINT [PK\_Game] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[GameImage] Script Date: 25.10.2021 19:19:40 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[GameImage](

[ImageID] [int] IDENTITY(1,1) NOT NULL,

[GameID] [int] NOT NULL,

[ImageSource] [varbinary](max) NULL,

CONSTRAINT [PK\_GameImage] PRIMARY KEY CLUSTERED

(

[ImageID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Genre] Script Date: 25.10.2021 19:19:40 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Genre](

[IDGenre] [int] IDENTITY(1,1) NOT NULL,

[GenreName] [nvarchar](50) NOT NULL,

CONSTRAINT [PK\_Genre] PRIMARY KEY CLUSTERED

(

[IDGenre] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Publisher] Script Date: 25.10.2021 19:19:40 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Publisher](

[IDPublisher] [int] IDENTITY(1,1) NOT NULL,

[PublisherName] [nvarchar](50) NOT NULL,

CONSTRAINT [PK\_Publisher] PRIMARY KEY CLUSTERED

(

[IDPublisher] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Rating] Script Date: 25.10.2021 19:19:40 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Rating](

[ID] [int] IDENTITY(1,1) NOT NULL,

[GameID] [int] NOT NULL,

[GameEstimation] [int] NOT NULL,

[UserID] [int] NOT NULL,

CONSTRAINT [PK\_Rating] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[User] Script Date: 25.10.2021 19:19:40 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[User](

[UserID] [int] IDENTITY(1,1) NOT NULL,

[Nickname] [nvarchar](30) NOT NULL,

[Email] [nvarchar](50) NOT NULL,

[Password] [nvarchar](530) NOT NULL,

[UserStatus] [int] NOT NULL,

CONSTRAINT [PK\_User] PRIMARY KEY CLUSTERED

(

[UserID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[UserStatus] Script Date: 25.10.2021 19:19:40 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[UserStatus](

[UserID] [int] NOT NULL,

[Status] [int] NOT NULL,

[ID] [int] IDENTITY(1,1) NOT NULL,

CONSTRAINT [PK\_UserStatus] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Game] WITH CHECK ADD CONSTRAINT [FK\_Game\_Developer] FOREIGN KEY([IDDeveloper])

REFERENCES [dbo].[Developer] ([IdDeveloper])

GO

ALTER TABLE [dbo].[Game] CHECK CONSTRAINT [FK\_Game\_Developer]

GO

ALTER TABLE [dbo].[Game] WITH CHECK ADD CONSTRAINT [FK\_Game\_Genre] FOREIGN KEY([IDGenre])

REFERENCES [dbo].[Genre] ([IDGenre])

GO

ALTER TABLE [dbo].[Game] CHECK CONSTRAINT [FK\_Game\_Genre]

GO

ALTER TABLE [dbo].[Game] WITH CHECK ADD CONSTRAINT [FK\_Game\_Publisher] FOREIGN KEY([IDPublisher])

REFERENCES [dbo].[Publisher] ([IDPublisher])

GO

ALTER TABLE [dbo].[Game] CHECK CONSTRAINT [FK\_Game\_Publisher]

GO

ALTER TABLE [dbo].[GameImage] WITH CHECK ADD CONSTRAINT [FK\_GameImage\_Game] FOREIGN KEY([GameID])

REFERENCES [dbo].[Game] ([ID])

GO

ALTER TABLE [dbo].[GameImage] CHECK CONSTRAINT [FK\_GameImage\_Game]

GO

ALTER TABLE [dbo].[Rating] WITH CHECK ADD CONSTRAINT [FK\_Rating\_User] FOREIGN KEY([UserID])

REFERENCES [dbo].[User] ([UserID])

GO

ALTER TABLE [dbo].[Rating] CHECK CONSTRAINT [FK\_Rating\_User]

GO

ALTER TABLE [dbo].[UserStatus] WITH CHECK ADD CONSTRAINT [FK\_UserStatus\_User] FOREIGN KEY([UserID])

REFERENCES [dbo].[User] ([UserID])

GO

ALTER TABLE [dbo].[UserStatus] CHECK CONSTRAINT [FK\_UserStatus\_User]

GO

USE [master]

GO

ALTER DATABASE [Games2] SET READ\_WRITE

GO

# ПРИЛОЖНИЕ Б

**XAML**

**AddEditPage.xaml**

<Page x:Class="Games2.AddEditPage"

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:Games2"

mc:Ignorable="d"

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

Title="AddEditPage">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="110\*"/>

<ColumnDefinition Width="310\*"/>

<ColumnDefinition Width="110\*"/>

<ColumnDefinition Width="269\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="60"/>

<RowDefinition Height="60"/>

<RowDefinition Height="60"/>

<RowDefinition Height="60"/>

<RowDefinition Height="60"/>

<RowDefinition Height="60"/>

<RowDefinition Height="81"/>

<RowDefinition Height="12"/>

</Grid.RowDefinitions>

<TextBlock Text="Название игры" Margin="16,26,7,17" Height="20" Width="87" Foreground="White"/>

<TextBlock Text="Жанр" Grid.Row="1" Margin="16,25,7,18" Height="20" Width="87" Foreground="White"/>

<TextBlock Text="Дата выхода" Grid.Row="3" VerticalAlignment="Center" HorizontalAlignment="Center" Margin="16,7,7,43" Height="16" Width="87" Foreground="White"/>

<TextBlock Text="Цена" Grid.Row="1" Grid.Column="2" Margin="22,25,0,18" Height="20" Width="88" Foreground="White"/>

<TextBlock Text="Режим" Grid.Row="5" Margin="13,14,10,10" Height="22" Width="87" Foreground="Black"/>

<TextBlock Text="Статус" Grid.Row="5" Grid.Column="2" Margin="22,14,1,10" Height="22" Width="87" Foreground="Black"/>

<TextBlock Text="Картинка" Grid.Row="2" Grid.Column="2" Margin="22,28,1,15" Height="22" Width="87" Foreground="White"/>

<TextBlock Text="Разработчик" Grid.Row="3" Grid.Column="2" Margin="22,28,1,15" Height="22" Width="87" Foreground="White" VerticalAlignment="Center" HorizontalAlignment="Center"/>

<TextBlock Text="Издатель" Grid.Row="4" Grid.Column="2" Margin="22,28,1,15" Height="22" Width="87" Foreground="White" VerticalAlignment="Center" HorizontalAlignment="Center"/>

<TextBox Text="{Binding Path=GameName}" Grid.Column="1" HorizontalAlignment="Left" Margin="10,16,0,17" Width="175" MaxLength="100" Height="30"/>

<ComboBox SelectedItem="{Binding Genre}" Grid.Column="1" Grid.Row="1" HorizontalAlignment="Left" Margin="10,15,0,18" Height="30" x:Name="ComboGenre" Width="175" DisplayMemberPath="GenreName"/>

<Calendar SelectedDate="{Binding ReleaseDate}" Grid.Row="2" x:Name="calendar" HorizontalAlignment="Left" FirstDayOfWeek="Monday" Height="auto" Margin="10,0,0,10" Grid.Column="1" Width="255" Grid.RowSpan="3"/>

<TextBox Text="{Binding Path=Price}" Grid.Column="3" Grid.Row="1" HorizontalAlignment="Left" Margin="10,15,0,18" Width="175" Height="30"/>

<ComboBox SelectedItem="{Binding Mode}" Grid.Row="5" Grid.Column="1" HorizontalAlignment="Left" Width="175" Margin="10,10,0,6" x:Name="ComboMode">

<local:Mode Gamemode="True"></local:Mode>

<local:Mode Gamemode="False"></local:Mode>

</ComboBox>

<ComboBox SelectedItem="{Binding GameStatus}" Grid.Row="5" Grid.Column="3" HorizontalAlignment="Left" Width="175" Margin="10,10,0,6" x:Name="ComboStatus">

<local:Status GameStatus="False"></local:Status>

<local:Status GameStatus="True"></local:Status>

</ComboBox>

<Button x:Name="LoadImage" Content="Загрузить кнопку" Width="Auto" Height="Auto" Grid.Row="2" Grid.Column="3" Margin="185,21,61,14" RenderTransformOrigin="2.185,0.105"/>

<TextBlock x:Name="ImageError" Grid.Row="3" Height="16" Margin="55,3,55,47" Width="0"/>

<Button Content="Сохранить" Name="BtnSave" Click="BtnSave\_Click" Margin="201,25,45,25" Grid.ColumnSpan="2" Grid.Row="6" Grid.Column="1" Height="Auto" Width="Auto"/>

<ComboBox SelectedItem="{Binding Developer}" Grid.Column="3" Grid.Row="3" HorizontalAlignment="Left" Margin="10,15,0,18" Height="30" x:Name="ComboDevel" Width="175" DisplayMemberPath="DeveloperName"/>

<ComboBox SelectedItem="{Binding Publisher}" Grid.Column="3" Grid.Row="4" HorizontalAlignment="Left" Margin="10,15,0,18" Height="30" x:Name="ComboPubl" Width="175" DisplayMemberPath="PublisherName"/>

<TextBox Grid.Column="3" Grid.Row="2" HorizontalAlignment="Left" Margin="10,20,0,15" Width="175" Height="30"/>

</Grid>

</Page>

**App.xaml**

<Application x:Class="Games2.App"

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

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

xmlns:local="clr-namespace:Games2"

StartupUri="MainWindow.xaml">

<Application.Resources>

<Style TargetType="Button">

<Setter Property="Margin" Value="5"/>

<Setter Property="Width" Value="175"/>

<Setter Property="Height" Value="30"/>

<Setter Property="Background" Value="Purple"/>

</Style>

<Style TargetType="TextBox">

<Setter Property="Width" Value="175"/>

<Setter Property="Height" Value="30"/>

<Setter Property="Margin" Value="5"/>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="HorizontalAlignment" Value="Stretch"/>

<Setter Property="VerticalContentAlignment" Value="Center"/>

</Style>

<Style TargetType="TextBlock">

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

</Style>

<Style TargetType="ComboBox">

<Setter Property="HorizontalAlignment" Value="Stretch"/>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="VerticalContentAlignment" Value="Center"/>

<Setter Property="Height" Value="30"/>

<Setter Property="Margin" Value="5"/>

</Style>

</Application.Resources>

</Application>

**GamePrewie.xaml**

<Page x:Class="Games2.GamePrewie"

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:Games2"

mc:Ignorable="d"

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

Title="GamePrewie">

<Grid>

<ListView Grid.Row="1" Name="LViewGame">

<ListView.ItemTemplate>

<DataTemplate>

<Grid Margin="20" Width="400">

<Grid.RowDefinitions>

<RowDefinition Height="70"></RowDefinition>

</Grid.RowDefinitions>

<Image Width="400" Grid.Row="1" Stretch="UniformToFill" HorizontalAlignment="Center" Margin="5">

<Image.Source>

<Binding Path="ImagePrewiew">

<Binding.TargetNullValue>

<ImageSource>Resourses/nopicture.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

<TextBlock Text="{Binding GameName}" VerticalAlignment="Center" TextAlignment="Center" TextWrapping="Wrap"

HorizontalAlignment="Center" Margin="5 5" FontSize="26" Grid.Row="0"/>

<TextBlock Text="{Binding Price}" Grid.Row="1" Margin="5 5 5 15" HorizontalAlignment="Center" FontSize="26" FontWeight="Bold"/>

<TextBlock Text="{Binding ReleaseDate}" Grid.Row="2" HorizontalAlignment="Center"/>

<TextBlock Text="{Binding Mode}" Grid.Row="3" HorizontalAlignment="Right" />

<TextBlock Text="{Binding Status}" Grid.Row="4" HorizontalAlignment="Left" />

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</Page>

**GamesPage.xaml**

<Page x:Class="Games2.GamesPage"

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:Games2"

mc:Ignorable="d"

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

Title="GamesPage" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="390\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<DataGrid x:Name="DGridGames" AutoGenerateColumns="False" IsReadOnly="True" SelectionChanged="DGridGames\_SelectionChanged" GridLinesVisibility="Horizontal">

<DataGrid.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0" ColorInterpolationMode="ScRgbLinearInterpolation">

<GradientStop Color="Black" Offset="0.202"/>

<GradientStop Color="White" Offset="1"/>

<GradientStop Color="#FF626262" Offset="0.32"/>

<GradientStop Color="#FF626262" Offset="0.489"/>

<GradientStop Color="#FF919191" Offset="0.684"/>

<GradientStop Color="Black" Offset="0.074"/>

</LinearGradientBrush>

</DataGrid.Background>

<DataGrid.Columns>

<DataGridTextColumn Header="Название игры" Binding="{Binding GameName}" Width="\*"/>

<DataGridTextColumn Header="Жанр" Binding="{Binding Genre.GenreName}" Width="\*"/>

<DataGridTextColumn Header="Дата выхода" Binding="{Binding ReleaseDate}" Width="\*"/>

<DataGridTextColumn Header="Цена" Binding="{Binding Price}" Width="\*"/>

<DataGridTextColumn Header="Режим" Binding="{Binding Mode}" Width="\*"/>

<DataGridTextColumn Header="Статус" Binding="{Binding GameStatus}" Width="\*"/>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать" Name="BtnEdit" Click="BtnEdit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Content="Добавить" Grid.Row="1" HorizontalAlignment="Left" Name="BtnAdd" Click="BtnAdd\_Click" Margin="137,10,0,9.6"/>

<Button Content="Удалить" Grid.Row="1" HorizontalAlignment="Right" Name="BtnDel" Click="BtnDel\_Click" Margin="0,10,137,9.6"/>

</Grid>

</Page>

**GamesPagee.xaml**

<Page x:Class="Games2.GamesPagee"

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:Games2"

mc:Ignorable="d"

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

Title="GamesPagee">

<Grid Background="Black" >

<Grid.RowDefinitions>

<RowDefinition Height="auto"></RowDefinition>

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

</Grid.RowDefinitions>

<WrapPanel Orientation="Horizontal" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal">

<TextBlock Text="Введите название для поиска:" Width="175" TextAlignment="Right" Foreground="White"></TextBlock>

<TextBox Width="225" Name="TBoxSearch" TextChanged="TBoxSearch\_TextChanged"></TextBox>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Выберите жанр:" Width="175" TextAlignment="Right" Foreground="White"></TextBlock>

<ComboBox Width="225" Name="ComboGenre" SelectionChanged="ComboGenre\_SelectionChanged" DisplayMemberPath="GenreName"/>

</StackPanel>

</WrapPanel>

<ListView Grid.Row="1" Name="LViewGames" ScrollViewer.HorizontalScrollBarVisibility="Disabled" HorizontalContentAlignment="Center" >

<ListView.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="Black" Offset="1"/>

<GradientStop Color="White"/>

</LinearGradientBrush>

</ListView.Background>

<ListView.ItemsPanel>

<ItemsPanelTemplate>

<WrapPanel Orientation="Horizontal" HorizontalAlignment="Center"></WrapPanel>

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Grid Margin="20" Width="450">

<Grid.RowDefinitions>

<RowDefinition Height="70"></RowDefinition>

<RowDefinition Height="330"></RowDefinition>

<RowDefinition Height="auto"></RowDefinition>

<RowDefinition Height="auto"></RowDefinition>

<RowDefinition Height="auto"></RowDefinition>

</Grid.RowDefinitions>

<Image Width="400" Grid.Row="1" Stretch="UniformToFill" HorizontalAlignment="Center" Margin="5">

<Image.Source>

<Binding Path="ImagePreview">

<Binding.TargetNullValue>

<ImageSource>C:\Users\Геляяяяяя\source\repos\Games2\Resources\nopicture.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

<TextBlock Text="{Binding GameName}" VerticalAlignment="Center" TextAlignment="Center" Width="390" TextWrapping="Wrap" HorizontalAlignment="Center" Margin="5 5" Grid.Row="0" FontSize="26" FontWeight="Bold"/>

<TextBlock Text="{Binding Genre.GenreName}" HorizontalAlignment="Center" Margin="5 5" Grid.Row="2"/>

<TextBlock Text="{Binding ReleaseDate}" Grid.Row="3" HorizontalAlignment="Center" Margin="5 5 5 15"/>

<TextBlock Text="{Binding Price, StringFormat={}{0:N2} РУБ}" Grid.Row="3" HorizontalAlignment="Left" FontSize="18"/>

<TextBlock Text="{Binding Mode}" Grid.Row="2" HorizontalAlignment="Left"/>

<TextBlock Text="{Binding GameStatus}" Grid.Row="3" HorizontalAlignment="Right"/>

<TextBlock Text="{Binding Developer.DeveloperName}" Grid.Row="4" HorizontalAlignment="Left"/>

<TextBlock Text="{Binding Publisher.PublisherName}" Grid.Row="4" HorizontalAlignment="Right"/>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</Page>

**InGame.xaml**

<Window x:Class="Games2.InGame"

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:Games2"

mc:Ignorable="d"

Title="InGame" Height="600" Width="800">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="60"/>

<RowDefinition/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid Background="Black" Margin="0,0,-0.4,1"></Grid>

<Image Source="Resources/ЛАГАТИП.png" Margin="344,0,342.6,3"></Image>

<Grid Grid.Row="2" Background="Black" Margin="0,0.4,-0.4,-5"/>

<Grid Grid.Row="1" Margin="0,2,0,0">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="Black" Offset="0"/>

<GradientStop Color="White" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Frame NavigationUIVisibility="Hidden" ContentRendered="MainFrame\_ContentRendered" Name="MainFrame"/>

<Button Name="BtnBack" Content="Back" HorizontalAlignment="Right" Margin="0,-49,25,494" Click="BtnBack\_Click" Height="Auto"/>

</Grid>

</Grid>

</Window>

**Login.xaml**

<Page x:Class="Games2.Login"

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:Games2"

mc:Ignorable="d"

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

Title="Login">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="100\*"/>

<RowDefinition Height="25"/>

<RowDefinition Height="30"/>

<RowDefinition Height="25"/>

<RowDefinition Height="30"/>

<RowDefinition Height="70\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="400\*"/>

<ColumnDefinition Width="400\*"/>

</Grid.ColumnDefinitions>

<TextBlock Height="30" Width="278" Grid.ColumnSpan="2" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20">Добро пожаловать в InGame!</TextBlock>

<TextBlock Margin="0 0 100 0" Height="21" Width="100" FontSize="15" Grid.Row="1" VerticalAlignment="Center" HorizontalAlignment="Right">Введите ник:</TextBlock>

<TextBox Name="Nikname" Grid.Row="2" Grid.ColumnSpan="2" HorizontalAlignment="Center" VerticalAlignment="Center" Height="30" Width="300"/>

<TextBlock Margin="150 0 0 0" Height="21" Width="120" FontSize="15" Grid.Row="3" VerticalAlignment="Center" HorizontalAlignment="Center" >Введите пароль:</TextBlock>

<PasswordBox x:Name="Password" Width="300" Grid.Row="4" Grid.ColumnSpan="2" HorizontalAlignment="Center" VerticalAlignment="Center" Height="30"></PasswordBox>

<Button x:Name="Entering" Content="Войти" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="5" Click="Button\_Click"></Button>

<Button x:Name="Reginn" Content="Регистрация" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="5" Click="Regin\_Click\_1"/>

</Grid>

</Page>

**MainWindow.xaml**

<Window x:Class="Games2.MainWindow"

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:Games2"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="60"/>

<RowDefinition/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid Background="Black" Margin="0,0,-0.4,1"></Grid>

<Image Source="Resources/ЛАГАТИП.png" Margin="344,0,342.6,3"></Image>

<TextBlock Foreground="White" HorizontalAlignment="Left">"По всем вопросам обращаться на почту <LineBreak>

</LineBreak> angelina.berezhnova@yandex.ru</TextBlock>

<Grid Grid.Row="2" Background="Black" Margin="0,0.4,-0.4,-5">

<Button Name="BtnBack" Content="Back" HorizontalAlignment="Right" Click="BtnBack\_Click" Height="20" Width="100"/>

</Grid>

<Grid Grid.Row="1" Margin="0,2,0,0">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="Black" Offset="0"/>

<GradientStop Color="White" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Frame NavigationUIVisibility="Hidden" ContentRendered="MainFramee\_ContentRendered" Name="MainFramee"/>

</Grid>

</Grid>

</Window>

**Regin.xaml**

<Page x:Class="Games2.Regin"

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:Games2"

mc:Ignorable="d"

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

Title="Regin">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="400\*"/>

<ColumnDefinition Width="400\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="100\*"/>

<RowDefinition Height="50\*"/>

<RowDefinition Height="50\*"/>

<RowDefinition Height="50\*"/>

<RowDefinition Height="50\*"/>

<RowDefinition Height="100\*"/>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Grid.ColumnSpan="2" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="30" >Давайте познакомимся :)</TextBlock>

<TextBlock Grid.Row="1" Grid.Column="0" VerticalAlignment="Center" HorizontalAlignment="Center" Text="Введите Ваш ник:" FontSize="20"/>

<TextBox x:Name="Nicknamee" Text="{Binding Nickname}" Grid.Row="1" Grid.Column="1" Width="300" VerticalAlignment="Center" HorizontalAlignment="Left"/>

<TextBlock Grid.Row="2" Grid.Column="0" VerticalAlignment="Center" HorizontalAlignment="Center" Text="Введите Ваш EMail:" FontSize="20"/>

<TextBox x:Name="Emaili" Text="{Binding Email}" Grid.Row="2" Grid.Column="1" Width="300" VerticalAlignment="Center" HorizontalAlignment="Left"/>

<TextBlock Grid.Row="3" Grid.Column="0" VerticalAlignment="Center" HorizontalAlignment="Center" Text="Введите Ваш пароль:" FontSize="20"/>

<TextBox x:Name="Pasword" Text="{Binding Password}" Grid.Row="3" Grid.Column="1" Width="300" VerticalAlignment="Center" HorizontalAlignment="Left"/>

<TextBlock Grid.Row="4" Grid.Column="0" VerticalAlignment="Center" HorizontalAlignment="Center" Text="Введите Ваш пароль еще раз:" FontSize="20"/>

<TextBox x:Name="Password2" Grid.Row="4" Grid.Column="1" Width="300" VerticalAlignment="Center" HorizontalAlignment="Left"/>

<Button Grid.Row="5" Grid.ColumnSpan="2" Content="Зарегистрироваться" Name="Regins" Click="Regins\_Click" VerticalAlignment="Center" HorizontalAlignment="Center"/>

</Grid>

</Page>

**C#**

**Функции, которые реализованы во всех элементах кода**

**AddEditPage.xaml.cs**

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.Data.Entity.Validation;

using System.Drawing;

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 Games2

{

public class Status

{

public bool GameStatus { get; set; } = false;

public override string ToString()

{

return GameStatus ? "Активна" : "Не активна";

}

}

public class Mode

{

public bool Gamemode { get; set; } = false;

public override string ToString()

{

return Gamemode ? "Однопользовательская" : "Многопользовательская";

}

}

/// <summary>

/// Логика взаимодействия для AddEditPage.xaml

/// </summary>

public partial class AddEditPage : Page

{

private Game \_currentGame = new Game();

public AddEditPage(Game selectedGame)

{

InitializeComponent();

if (selectedGame != null)

\_currentGame = selectedGame;

DataContext = \_currentGame;

ComboGenre.ItemsSource = Games2Entities.GetContext().Genre.ToList();

ComboDevel.ItemsSource = Games2Entities.GetContext().Developer.ToList();

ComboPubl.ItemsSource = Games2Entities.GetContext().Publisher.ToList();

}

private void calendar\_SelectedDatesChanged(object sender, SelectionChangedEventArgs e)

{

DateTime? SelectedDate = calendar.SelectedDate;

MessageBox.Show(SelectedDate.Value.Date.ToShortDateString());

}

private void BtnSave\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentGame.GameName))

errors.AppendLine("Ошибка 1: Укажите название игры!");

if (\_currentGame.Genre == null)

errors.AppendLine("Ошибка 2: Выберите жанр игры!");

if (\_currentGame.ReleaseDate == null)

errors.AppendLine("Ошибка 3: Выберите дату издания игры!");

if (\_currentGame.Price < 0 || \_currentGame.Price > 10000)

errors.AppendLine("Ошибка 4: Укажите стоимость игры в рублях!");

if (\_currentGame.Mode == null)

errors.AppendLine("Ошибка 5: Выберите режим игры!");

if (\_currentGame.GameStatus == null)

errors.AppendLine("Ошибка 6: Выберите статус игры!");

if (\_currentGame.Developer.DeveloperName == null)

errors.AppendLine("Ошибка 7: Введите разработчика игры!");

if (\_currentGame.Publisher.PublisherName == null)

errors.AppendLine("Ошибка 8: Введите издателя игры!");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString());

return;

}

if (\_currentGame.ID == 0)

Games2Entities.GetContext().Game.Add(\_currentGame);

try

{

Games2Entities.GetContext().SaveChanges();

MessageBox.Show("Информация сохранена!");

Manager.MainFrame.GoBack();

}

catch (DbEntityValidationException ex)

{

List<string> errorMessages = new List<string>();

foreach (DbEntityValidationResult validationResult in ex.EntityValidationErrors)

{

string entityName = validationResult.Entry.Entity.GetType().Name;

foreach (DbValidationError error in validationResult.ValidationErrors)

{

errorMessages.Add(entityName + "." + error.PropertyName + ": " + error.ErrorMessage);

MessageBox.Show(ex.Message.ToString());

MessageBox.Show(entityName.ToString());

}

}

}

}

}

}

**App.xaml.cs**

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Threading.Tasks;

using System.Windows;

namespace Games2

{

/// <summary>

/// Логика взаимодействия для App.xaml

/// </summary>

public partial class App : Application

{

}

}

**GamePrewie.xaml.cs**

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 Games2

{

/// <summary>

/// Логика взаимодействия для GamePrewie.xaml

/// </summary>

public partial class GamePrewie : Page

{

public GamePrewie()

{

InitializeComponent();

var currentGames = Games2Entities.GetContext().Game.ToList();

}

}

}

**GamesPage.xaml.cs**

<Page x:Class="Games2.GamesPage"

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:Games2"

mc:Ignorable="d"

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

Title="GamesPage" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="390\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<DataGrid x:Name="DGridGames" AutoGenerateColumns="False" IsReadOnly="True" SelectionChanged="DGridGames\_SelectionChanged" GridLinesVisibility="Horizontal">

<DataGrid.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0" ColorInterpolationMode="ScRgbLinearInterpolation">

<GradientStop Color="Black" Offset="0.202"/>

<GradientStop Color="White" Offset="1"/>

<GradientStop Color="#FF626262" Offset="0.32"/>

<GradientStop Color="#FF626262" Offset="0.489"/>

<GradientStop Color="#FF919191" Offset="0.684"/>

<GradientStop Color="Black" Offset="0.074"/>

</LinearGradientBrush>

</DataGrid.Background>

<DataGrid.Columns>

<DataGridTextColumn Header="Название игры" Binding="{Binding GameName}" Width="\*"/>

<DataGridTextColumn Header="Жанр" Binding="{Binding Genre.GenreName}" Width="\*"/>

<DataGridTextColumn Header="Дата выхода" Binding="{Binding ReleaseDate}" Width="\*"/>

<DataGridTextColumn Header="Цена" Binding="{Binding Price}" Width="\*"/>

<DataGridTextColumn Header="Режим" Binding="{Binding Mode}" Width="\*"/>

<DataGridTextColumn Header="Статус" Binding="{Binding GameStatus}" Width="\*"/>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать" Name="BtnEdit" Click="BtnEdit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Content="Добавить" Grid.Row="1" HorizontalAlignment="Left" Name="BtnAdd" Click="BtnAdd\_Click" Margin="137,10,0,9.6"/>

<Button Content="Удалить" Grid.Row="1" HorizontalAlignment="Right" Name="BtnDel" Click="BtnDel\_Click" Margin="0,10,137,9.6"/>

</Grid>

</Page>

**GamesPagee.xaml.cs**

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 Games2

{

/// <summary>

/// Логика взаимодействия для GamesPagee.xaml

/// </summary>

public partial class GamesPagee : Page

{

public GamesPagee()

{

InitializeComponent();

var allgenre = Games2Entities.GetContext().Genre.ToList();

allgenre.Insert(0, new Genre

{

GenreName = "Все жанры"

});

ComboGenre.ItemsSource = allgenre;

ComboGenre.SelectedIndex = 0;

//var currentGames = Games2Entities.GetContext().Game.ToList();

//LViewGames.ItemsSource = currentGames;

}

private void UpdateGames()

{

var currentGames = Games2Entities.GetContext().Game.ToList();

if (ComboGenre.SelectedIndex > 0)

currentGames = currentGames.Where(p => p.Genre==ComboGenre.SelectedItem).ToList();

currentGames = currentGames.Where(p => p.GameName.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();

LViewGames.ItemsSource = currentGames.OrderBy(p => p.GameName).ToList();

}

private void TBoxSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

UpdateGames();

}

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

{

UpdateGames();

}

}

}

**InGame.xaml.cs**

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;

namespace Games2

{

/// <summary>

/// Логика взаимодействия для InGame.xaml

/// </summary>

public partial class InGame : Window

{

public InGame()

{

InitializeComponent();

if (Nicknameeeee.Status == 0 || Nicknameeeee.Status == 1)

{

MainFrame.Navigate(new GamesPage());

} else {

MainFrame.Navigate(new GamesPagee());

}

Manager.MainFrame = MainFrame;

}

private void BtnBack\_Click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.GoBack();

}

private void MainFrame\_ContentRendered(object sender, EventArgs e)

{

if (MainFrame.CanGoBack)

{

BtnBack.Visibility = Visibility.Visible;

}

else

{

BtnBack.Visibility = Visibility.Hidden;

}

}

}

}

**Login.xaml.cs**

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 Games2

{

/// <summary>

/// Логика взаимодействия для Login.xaml

/// </summary>

public partial class Login : Page

{

public Login()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

var Users = Games2Entities.GetContext().User.ToList();

bool Prov = false;

foreach (User Nickname in Users)

{

if ((Nikname.Text == Nickname.Nickname) && (Password.Password == Nickname.Password))

{

Nicknameeeee.Prov = true;

Nicknameeeee.Nickname = Nikname.Text;

Nicknameeeee.Status = Nickname.UserStatus;

InGame xaskWindow = new InGame();

xaskWindow.Show();

}

}

if (Prov == false)

{

Manager.MainFramee.Navigate(new Login());

StringBuilder errors = new StringBuilder();

if (Prov == false)

errors.AppendLine("Ошибка 1: Некорректные логин или пароль!!");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString());

return;

}

}

}

private void Regin\_Click\_1(object sender, RoutedEventArgs e)

{

Manager.MainFramee.Navigate(new Regin());

//MainFramee.Navigate(new Regin());

//Manager.MainFramee = MainFramee;

}

}

}

**MainWindow.xaml.cs**

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 Games2

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

MainFramee.Navigate(new Login());

Manager.MainFramee = MainFramee;

if (Nicknameeeee.Prov == true)

{

this.Close();

}

}

private void BtnBack\_Click(object sender, RoutedEventArgs e)

{

Manager.MainFramee.GoBack();

}

private void MainFramee\_ContentRendered(object sender, EventArgs e)

{

if (MainFramee.CanGoBack)

{

BtnBack.Visibility = Visibility.Visible;

}

else

{

BtnBack.Visibility = Visibility.Hidden;

}

}

}

}

**Regin.xaml.cs**

using System;

using System.Collections.Generic;

using System.Data.Entity.Validation;

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 Games2

{

/// <summary>

/// Логика взаимодействия для Regin.xaml

/// </summary>

public partial class Regin : Page

{

private User \_currentUser = new User();

public Regin()

{

InitializeComponent();

DataContext = \_currentUser;

}

private void Regins\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentUser.Nickname))

errors.AppendLine("Ошибка 1: Укажите Ваш никнейм!");

if (string.IsNullOrWhiteSpace(\_currentUser.Email))

errors.AppendLine("Ошибка 2: Укажите Ваш Email!");

if (string.IsNullOrWhiteSpace(\_currentUser.Password))

errors.AppendLine("Ошибка 3: Укажите Ваш пароль!");

if (Password2.Text != \_currentUser.Password)

errors.AppendLine("Ошибка 4: Пароли не совпадают!");

//if (string.IsNullOrWhiteSpace(Password2))

// errors.AppendLine("Ошибка 3: Укажите Ваш пароль!");

\_currentUser.UserStatus = 2;

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString());

return;

}

if (\_currentUser.UserID == 0)

Games2Entities.GetContext().User.Add(\_currentUser);

try

{

Games2Entities.GetContext().SaveChanges(); //Выдает ошибку

MessageBox.Show("Информация сохранена!");

Manager.MainFramee.GoBack();

}

catch (DbEntityValidationException ex)

{

List<string> errorMessages = new List<string>();

foreach (DbEntityValidationResult validationResult in ex.EntityValidationErrors)

{

string entityName = validationResult.Entry.Entity.GetType().Name;

foreach (DbValidationError error in validationResult.ValidationErrors)

{

errorMessages.Add(entityName + "." + error.PropertyName + ": " + error.ErrorMessage);

MessageBox.Show(ex.Message.ToString());

MessageBox.Show(entityName.ToString());

}

}

//Console.WriteLine(ex);

//MessageBox.Show(ex.Message.ToString());

}

}

}

}

**Дополнительные файлы:**

**Manager.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Controls;

namespace Games2

{

class Manager

{

public static Frame MainFrame { get; set; }

public static Frame MainFramee { get; set; }

}

}

**Nicknameeeee.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Games2

{

class Nicknameeeee

{

public static string Nickname;

public static int Status;

public static bool Prov;

}

}

**BaseModel.Context.cs**

//------------------------------------------------------------------------------

// <auto-generated>

// Этот код создан по шаблону.

//

// Изменения, вносимые в этот файл вручную, могут привести к непредвиденной работе приложения.

// Изменения, вносимые в этот файл вручную, будут перезаписаны при повторном создании кода.

// </auto-generated>

//------------------------------------------------------------------------------

namespace Games2

{

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

public partial class Games2Entities : DbContext

{

private static Games2Entities \_context;

public Games2Entities()

: base("name=Games2Entities")

{

}

public static Games2Entities GetContext()

{

if (\_context == null)

{

\_context = new Games2Entities();

}

return \_context;

}

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

throw new UnintentionalCodeFirstException();

}

public virtual DbSet<Developer> Developer { get; set; }

public virtual DbSet<Game> Game { get; set; }

public virtual DbSet<GameImage> GameImage { get; set; }

public virtual DbSet<Genre> Genre { get; set; }

public virtual DbSet<Publisher> Publisher { get; set; }

public virtual DbSet<Rating> Rating { get; set; }

public virtual DbSet<sysdiagrams> sysdiagrams { get; set; }

public virtual DbSet<User> User { get; set; }

public virtual DbSet<UserStatus> UserStatus { get; set; }

}

}