МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ижевский государственный технический университет имени

М. Т. Калашникова»

Институт «Информатика и вычислительная техника»

Кафедра «Программное обеспечение»

Работа защищена с оценкой

«\_\_\_\_\_\_\_\_\_»

Дата \_\_\_\_\_\_\_\_\_\_

Подпись \_\_\_\_\_\_\_/\_\_\_\_\_\_\_

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе на тему:

«Разработка сервиса-агрегатора актуальной информации с игровых платформ»

Выполнил:

студент группы Б17-191-1 Тихонов Е.М.

Руководитель:

к.т.н., доцент Архипов И.О.

Рецензия:

Степень достижения поставленной цели работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

полнота разработки темы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

уровень самостоятельности работы обучающегося\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

недостатки работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc70764689)

[1. РАЗРАБОТКА СИСТЕМЫ «СЕРВИС-АГРЕГАТОР АКТУАЛЬНОЙ ИНФОРМАЦИИ С ИГРОВЫХ ПЛАТФОРМ» 4](#_Toc70764690)

[1.1. Обоснование целесообразности разработки системы 4](#_Toc70764691)

[1.1.1. Назначение объекта автоматизации 4](#_Toc70764692)

[1.1.2. Обоснование цели создания системы 4](#_Toc70764693)

[1.1.3. Обоснование состава автоматизируемых задач 4](#_Toc70764694)

[1.2. Аналитический обзор 4](#_Toc70764695)

[1.3. Основные требования к системе 7](#_Toc70764696)

[1.3.1. Основные цели создания системы и критерии эффективности ее функционирования 7](#_Toc70764697)

[1.3.2. Функциональное назначение системы 7](#_Toc70764698)

[1.3.3. Особенности объекта автоматизации и условия эксплуатации, определяющие основные требования к системе 9](#_Toc70764699)

[1.3.4. Требования к функциональной структуре системы 10](#_Toc70764700)

[1.3.5. Типовые проектные решения и (или) пакеты прикладных программ, применяемые в системе 12](#_Toc70764701)

[1.3.6. Требования к техническому обеспечению 12](#_Toc70764702)

[1.4. Основные технические решения проекта системы 13](#_Toc70764703)

[1.4.1. Описание организации информационной базы 13](#_Toc70764704)

[1.4.2. Описание системы программного обеспечения 16](#_Toc70764705)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18](#_Toc70764706)

# ВВЕДЕНИЕ

В настоящее время наблюдается рост рынка игровой индустрии, в том числе и сегмента игр на ПК. Но, в отличии, от сегмента мобильных устройств и игровых консолей, где весь рынок игр сконцентрирован на одной платформе (например, на Android в Google Play), сегмент ПК сильно разрознен.

Вместе с увеличением рынка растет и количество крупных, и не очень, участников рынка – издателей и дистрибьюторов, большинство из которых владеют собственной платформой для распространения игровых изданий. Также довольно часто встречаются издания, выходящие не на чьей-либо платформе, а распространяющиеся самостоятельно.

Из-за такой диверсифицированности рынка у обычных потребителей, желающих найти себе новую игру, возникает проблема – на поиск игр уходит больше времени, чем им хотелось бы. И для того, чтобы иметь большую выборку по интересующим игрока критериями, ему приходится искать игры на большом количестве платформ и сводить результаты самостоятельно.

Одним из способов решения этой проблемы можно выделить реализацию сервиса, который будет собирать информацию с разных игровых платформ и давать пользователю возможность проводить поиск игр в единой информационной системе.

Исходя из этого можно сформулировать следующую задачу: необходимо разработать сервис-агрегатор информации с игровых платформ, который будет хранить информацию о большинстве игр, выпущенных для ПК и предоставлять пользователю возможность поиска этих игр в одном месте с широким выбором фильтров, а также организовывать списки выбранных игр по разным категориям в личном кабинете.

На момент начала разработки приложения, на рынке не было представлено систем, полностью закрывающих данную проблему.

# РАЗРАБОТКА СИСТЕМЫ «СЕРВИС-АГРЕГАТОР АКТУАЛЬНОЙ ИНФОРМАЦИИ С ИГРОВЫХ ПЛАТФОРМ»

## Обоснование целесообразности разработки системы

### Назначение объекта автоматизации

Объектом автоматизации является процесс поиска игр с разных платформ по общим критериям.

### Обоснование цели создания системы

Целью создания системы является:

* Упрощение выполнения поиска игр по различным критериям (название, жанр, издатель, разработчик, и т.д.). Пользователю предоставляется возможность получения доступа к игровой информации разных платформ в едином информационном пространстве;
* Минимизация необходимости переключения между приложениями, и, в следствии, переключения внимания. Действия выполняются в одном окне, вместо фильтрации списков в приложениях игровых платформах и использования различных поисковых систем для поиска самостоятельных проектов.

### Обоснование состава автоматизируемых задач

Для разрабатываемой системы могут быть определены следующие задачи:

* Разработка модуля «Личный кабинет»
* Разработка модуля администрирования
* Разработка модуля управления данными об играх
* Разработка модулей интеграции с платформами крупных игровых дистрибьюторов

## Аналитический обзор

Игры стали повсеместным источником развлечения — если сложить все часы активной игры и ее просмотра, то получится, что это самое любимое занятие в мире. Масштабы игровой индустрии сопоставимы, например, с киноиндустрией. А по скорости роста за последние пять лет индустрия видеоигр существенно ее опережала.

По степени влияния на потребителей и вовлеченности их в интерактивное окружение, предлагаемое видеоиграми, этот сегмент уже давно выделяется среди других видов развлечений.

По мере развития компьютерных технологий, увеличивался и размер групп разработчиков, так как увеличивающаяся сложность графики и программирования в целом требовала все большего штата специалистов. В настоящее время стоимость разработки может достигать десятков (нескольких сотен) миллионов долларов, даже если разработчики используют связующее ПО и полностью готовые игровые движки. Большинство профессионально выполненных компьютерных игр разрабатываются в течение от одного до трех лет, что создает повышенные требования к бюджету игры. В следствии чего, для снижения издержек некоторые разработчики переключаются на использование других способов распространения игр, например, цифровая дистрибуция.

Особенности индустрии компьютерных игр во многом схожи с другими ответвлениями индустрии развлечений (например, музыкальная индустрия), однако игровую индустрию часто осуждают за малые вознаграждения разработчиков. Это способствует существованию независимых разработок, которые начинаются после того как разработчики покидают свои старые компании и основывают собственные новые компании и проекты. В наиболее значительных случаях такие компании постепенно развиваются в огромные безликие корпорации, принимающие правила ведения бизнеса своих предшественников, поддерживая таким образом упомянутый цикл.

Именно по указанным выше причинам и возникла проблема децентрализованности рынка выходящих на ПК видеоигр, заставляющая пользователя совершать много однотипных действий, отличающих только интерфейсом платформы.

Так же следует упомянуть, что следствием этой проблемы также является разрозненность библиотеки игр пользователя: игра могут хранится только в библиотеке платформы, на которой эта игра вышла. А в случае самостоятельных проектов возможность ведения их библиотеки не представляется возможным в принципе.

Разрабатываемый программный продукт должен помочь в решении указанных проблем.

Кроме разрабатываемой информационной системы, существует также некоторые другие сервисы для поиска игр по критериям и ведения библиотеки. Сравним следующие варианты: «www.backloggery.com», «www.grouvee.com», «lets-play.by», «Chaynikam.info/gamebase» - составим сводную таблицу (таблица 1.1), где колонки – это определенные критерии, строки – данные приложения, а знаки «+» или «–» означают, удовлетворяет или не удовлетворяет данное приложение этому критерию. Выделим следующие критерии: поиск игр по критериям (название, жанр, издатель и т.д.), возможность оценивания игр пользователем, ведение библиотеки игр пользователя, написание отзывов к играм, импорт и экспорт информации о библиотеке пользователя.

Таблица 1.1. Сравнение сервисов поиска игр

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Сервис | Поиск игр по критериям (название, жанр, издатель и т.д.) | Возможность оценивания игр пользователем | Ведение библиотеки игр пользователя | Написание отзывов к играм | Импорт и экспорт информации о библиотеке пользователя |
| Разрабатываемая система | + | + | + | + | + |
| www.backloggery.com | - | + | + | + | - |
| www.grouvee.com | + | + | - | - | - |
| lets-play.by | + | - | - | - | - |
| Chaynikam.info/gamebase | + | - | - | - | - |

На основе таблицы 1.1 можно сделать вывод, что разрабатываемое приложение – сервис-агрегатор актуальной информации с игровых платформ – удовлетворяет всем приведенным критериям, в отличие от других вариантов, а значит является лучшим решением поставленной задачи.

## Основные требования к системе

### Основные цели создания системы и критерии эффективности ее функционирования

Создание сервиса-агрегатора актуальной информации с игровых платформ для упрощения поиска игр по необходимым критериям.

Эффективность работы приложения можно оценить, сопоставив результаты поиска игр в приложении со сводными результатами поиска на игровых платформах (с которыми настроена интеграция) по тем же критериям.

### Функциональное назначение системы

Требования к функциональным особенностям сервиса:

1. Система должна иметь возможность регистрации пользователей с помощью собственной системы регистрации учетных записей
2. Система должна иметь возможность аутентификации пользователей по логину и паролю с помощью собственной системы управления учетными записями
3. Пользователь должен иметь возможность настраивать данные о себе (имя, фамилия, фото, дата рождения, никнейм, и т.д.) в личном кабинете
4. Реализовать разделение ролей пользователей на обычного и администратора, которому, помимо основного функционала, будет доступна возможность управления списком игр и модерирования отзывов об играх
5. Система должна хранить следующую информацию о каждой игре:

- Название

- Описание

- Лого

- Жанр

- Пользовательский рейтинг

- Издатель

- Разработчик

- Дата выхода

- Возрастное ограничение

Каждая категория представляет собой отдельный столбец в таблице.

В зависимости от каждой игры состав полей может отличаться.

1. Реализовать поиск игр по базе по критериям, указанным в п.5 в виде панели фильтрации в меню поиска. Также добавить возможность поиска по названию на главной странице сайта.
2. Реализовать возможность оценки игр пользователями. Особенности:

- Оценивать могут только авторизованные пользователи

- Оценка проходит на странице игры

- Игра оценивается от 1 до 10

- Пользователь может в любой момент поменять свою оценку

Также на странице игры для всех будет отображаться ее средняя оценка пользователями.

1. Реализовать интеграцию с платформами дистрибьюторов игр (которые предоставляют открытое API) для регулярного обновления базы игр. Обновление должно происходить автоматически силами системы по заданному расписанию. Также должна быть возможность ручного запуска администратором вне расписания по соответствующей кнопке в панели администратора.
2. Система должна дать авторизованным пользователям возможность вести собственную библиотеку игр из имеющихся в базе. Для этого в личном кабинете создать списки по статусам “Запланировано”, “Играю”, “Брошено”, “Пройдено”. Пользователь со страницы игры может ее добавить в список, перемещать по спискам и удалить ее из своих списков. Строка каждого списка должна содержать название игры, текущую оценку пользователя и при нажатии открывать страницу игры.
3. Система должна дать пользователю возможность экспортировать информацию о своих списках игр в виде файла в формате json. Для этого необходимо разработать шаблон, в соответствии с которым данные должны подставляться в файл, который будет выгружен для пользователя.
4. Система должна дать пользователю возможность импортировать списки игр других пользователей по соответствующей кнопке в личном кабинете. Импорт происходит в виде загрузки Json-файла, который соответствует шаблону (см. П.10). Дальше данные из файла заносятся в списки текущего пользователя.
5. Система должна дать авторизованным пользователям возможность оставления отзыва об игре и просмотр отзывов других пользователей на странице игры. После написания отзыва он попадает на модерацию администратору. Администратор может либо принять отзыв, и он появится в списке отзывов к игре, либо отклонить, и отзыв будет удален;
6. Система должна содержать новостную ленту в виде колонки на главной странице, ссылки которой будут вести на страницу с новостью;
7. Система должна отображать список самых популярных игр среди пользователей сервиса на главной странице. Популярность считается из количества добавлений игры пользователями в свои списки;
8. Система должна дать администратору возможность вручную добавлять новые игры в базу. Для этого необходимо разработать специальное меню, в котором администратор сможет ввести все необходимые данные об игры, загрузить картинки и отправить игру в базу.

### Особенности объекта автоматизации и условия эксплуатации, определяющие основные требования к системе

Компьютер, на котором будет установлено приложение, должно иметь доступ к сети Интернет, на нем должен быть установлен пакет «.NET Core SDK » версии 3.1 и выше, и СУБД «Microsoft SQL Server» версии 2012 и выше.

Компьютер, на котором будет использоваться приложение, должен иметь доступ в Интернет и установленный браузер для подключения к сервису.

На рисунке 1.1 представлена диаграмма вариантов использования.



Рис. 1.1. Диаграмма вариантов использования

### Требования к функциональной структуре системы

В основе сервиса лежит модульная структура.

Модуль «Личный кабинет» предназначен для регистрации и авторизации, управления профилями и библиотеками, экспорта и импорта библиотек пользователей.

Модуль администрирования предназначен для ручного управления базой игр, в том числе и добавления игр в ручную с помощью заполнения специальной формы на сервисе, модерирования отзывов пользователей, а также управления системными процессами, такими как внеочередной запуск процессов интеграции и др.

Модуль управления данными об играх предназначен для взаимодействия пользователя с базой игр – поиск игр по критериям, оценивание игр и написание отзывов к ним.

Подсистема интеграции с платформами игровых дистрибьюторов состоит из модуля взаимодействия с API внешних игровых платформ, которые получают информацию с внешних источников и заносят ее в БД, а также сервиса для запуска процессов интеграции по расписанию в автоматическом режиме.

Подсистема управления БД предназначена для взаимодействия с БД игровой информации и БД учетных записей пользователей.

На рисунке 1.2 представлена структурная схема приложения.

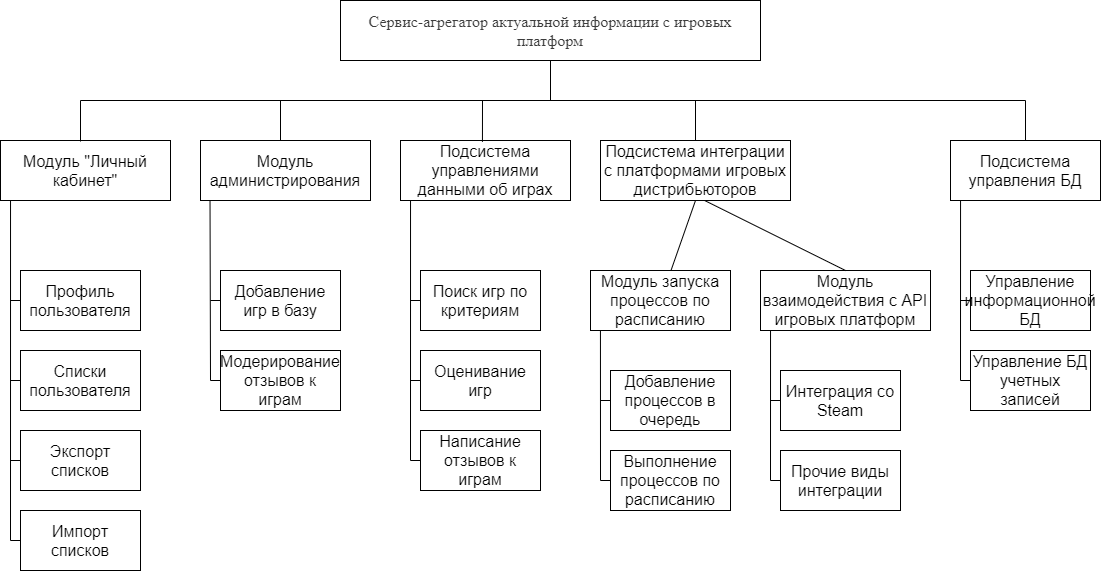


Рис 1.2. Структурная схема приложения

На рисунке 1.3 представлена схема работы системы

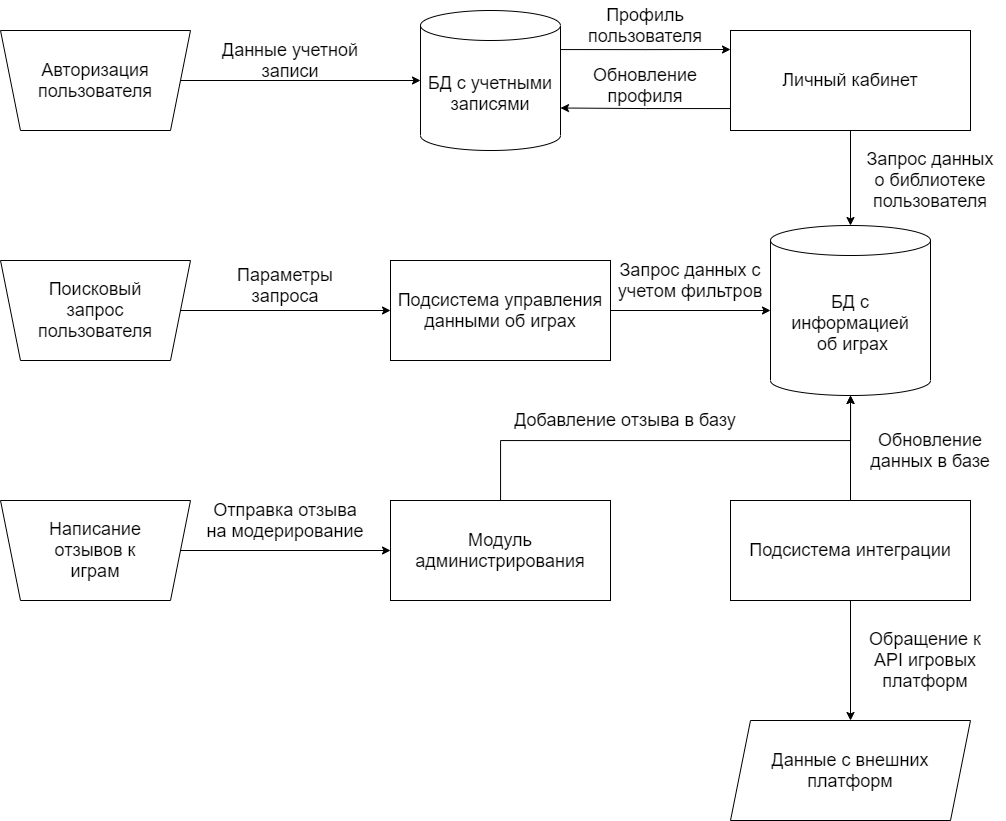


Рис 1.3. Схема работы системы

### Типовые проектные решения и (или) пакеты прикладных программ, применяемые в системе

Для разработки сервиса была выбрана технология ASP .Net Core 3.1 и MVC, а в качестве IDE – Visual Studio 2019. Разработка велась на языке C# 8 с использованием фреймворков AspNetCore.Identity.EntityFrameworkCore 3.1.8, AspNetCore.Mvc.NewtonsoftJson 3.1.9, Microsoft.EntityFrameworkCore 3.1.8, System.Linq.Dynamic.Core 1.2.5, в качестве СУБД выбрана Microsoft SQL Server Express 2019.

### Требования к техническому обеспечению

Требования к компьютеру для сервера и СУБД:

Процессор – 6 ядер, частота 3.1 Ггц (Intel Core i5-8400/ AMD Ryzen 5 3500).

Диск – HDD 30+ ГБ.

Оперативная память – 16 ГБ.

Сеть – канал связи со скоростью от 20 Мбит/сек.

Требования к компьютеру пользователя:

Процессор - Intel Pentium 4 / Athlon 64 или более поздней версии с поддержкой SSE2.

Диск – не менее 500 МБ.

Оперативная память – 512 МБ.

Сеть - канал связи со скоростью от 20 Мбит/сек.

## Основные технические решения проекта системы

### Описание организации информационной базы

Все данные программы будут храниться в виде баз данных в MSSQL Server

Данные о пользователям и данные об объектах системы будут хранится в разных базах.

База данных объектов системы:

|  |  |  |
| --- | --- | --- |
| Имя БД | Назначение | Состав столбцов |
| Catalogs | Хранит все виды каталогов пользователей | Id (PK, int, Not NULL),  Name (nvarchar(max), NULL),  Description (nvarchar(max), NULL) |
| Games | Хранит информацию об играх | Id (PK, int, Not NULL),  Name (nvarchar(max), NULL),  Description (nvarchar(max), NULL),  Logo (varbinary(max), NULL)  Genre (int, Not NULL),  Rating (float, Not NULL),  Publisher (nvarchar(max), NULL),  Developer (nvarchar(max), NULL),  ReleaseDate (datetime(7), Not NULL),  AgeRating (nvarchar(max), NULL) |
| GameSummaries | Хранит информации о каждом добавлении любой игры любым пользователем в любой из своих списков. Для пользователя представляет строку в его списке игр | Id (PK, int, Not NULL),  GameName (nvarchar(max), NULL),  Genre (int, Not NULL),  Rate (float, Not NULL),  GameId (FK, int Not NULL),  CatalogId (FK, int Not NULL),  UserId(nvarchar(450), NULL) |
| GameComments | Хранит отзывы об играх | Id (PK, int, Not NULL),  GameId (FK, int, Not NULL),  UserId (nvarchar(456), NULL),  Text (nvarchar(max), NULL),  Date (datetime2(7), Not NULL) |

База данных учетных записей:

|  |  |  |
| --- | --- | --- |
| Имя БД | Назначение | Состав столбцов |
| AspNetRoles | Хранит информацию о существующих в системе ролях пользователей | Id (PK, navrchar(450), NULL),  Name (nvarchar(256), NULL) |
| AspNetUsers | Хранит учетные записи пользователей | Id (PK, navrchar(450), NULL),  UserName (nvarchar(256), NULL),  PasswordHash (navrchar(max), NULL),  Login (navrchar(max), NULL),  FullName (navrchar(max), NULL),  BirthDate (datetime2(7), Not NULL) |
| AspNetUserRoles | Хранит связи пользователей и их ролей | UserId (PK, FK, navrchar(450), NULL),  RoleId (PK, FK, navrchar(450), NULL), |

Помимо указанных таблиц и столбцов будут хранится и метаданные системы, необходимые только для внутренних вычислений.

Для хранения конфигурационных данных используется файл appsettings.json. Данные в нем хранятся в формате «”ключ”: “значение”». В нем будут хранится строки подключения к SQL серверу, параметры запуска синхронизаций и пр.

ER-диаграмма сущностей представлена на рисунке 1.4.

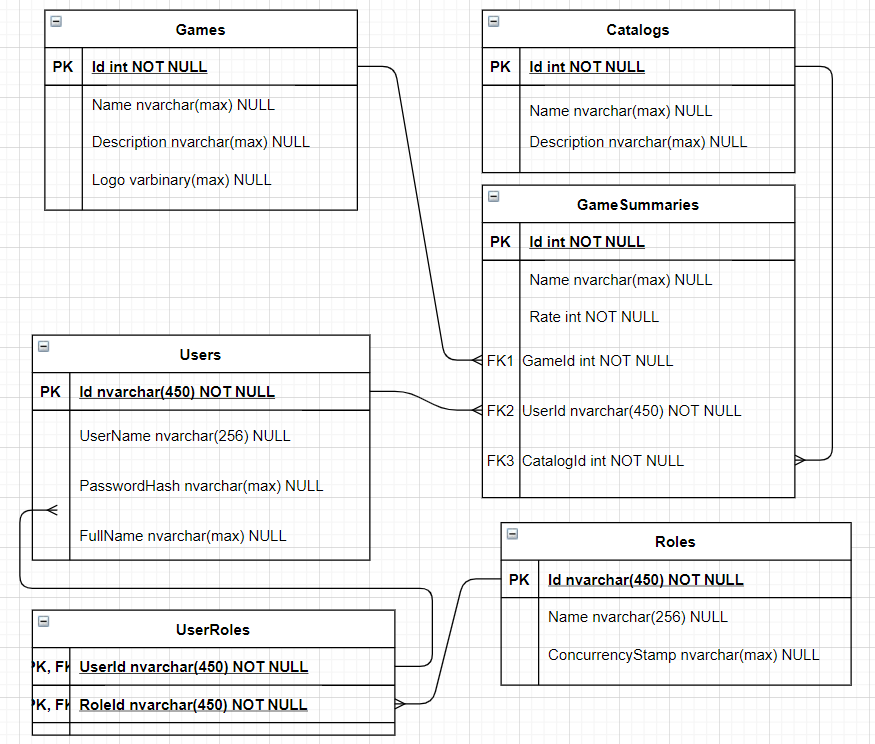


Рис. 1.4. ER-диаграмма

### Описание системы программного обеспечения

Компьютер для сервера:

Требуемая ОС – Windows 7 SP1/ Windows 8.1 / Windows 10 с версии 1607 и выше / Windows Server 2012 R2 / Windows Server 2016 / Windows Server 2019

Требуемая СУБД - Microsoft SQL Server 2012 / 2014 / 2016 / 2019

Требуемая версия компонентов IIS – 7.5 и выше

Требуемая версия .Net Core SDK – 3.1 и выше

Устройство пользователя:

Требуемая ОС – любая ОС, поддерживающая любой из следующих интернет-браузеров:

Google Chrome - версия 90 и выше,

Firefox – версия 71 и выше,

Opera – Версия 15 и выше,

Microsoft Edge – версия 20 и выше

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Сенилов М.А., Архипов И.О., Соболева В.П. Методические указания по выполнению выпускной работы бакалавра. Направление 230100.62 «Информатика и вычислительная техника» для студентов кафедры «Программное обеспечение». – Ижевск: Издательство ИжГТУ, 2010г. – 19с;
2. ASP .NET Core Web Browser Support [Электронный ресурс] // URL: https://docs.telerik.com/aspnet-core/compatibility/browser-support/ (дата обращения: 28.04.2021)
3. Документация по ASP .NET | Microsoft Docs [Электронный ресурс] // URL: https://docs.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-5.0/ (дата обращения: 28.04.2021)
4. Глобальный обзор игровой индустрии: тренды, инсайты и прогнозы на 2021 год [Электронный ресурс] // URL: https://adindex.ru/news/researches/2019/01/25/230750.phtml (дата обращения: 28.04.2021)
5. Обзор Entity Framework Core [Электронный ресурс] // URL: https://docs.microsoft.com/ru-ru/ef/core/ (дата обращения: 29.04.2021)