**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное автономное образовательное   
учреждение высшего образования   
«Санкт-Петербургский политехнический университет Петра Великого»**

**ИНСТИТУТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

|  |  |
| --- | --- |
|  | **Допустить к защите**  Заместитель директора по  учебно-методической работе  \_\_\_\_\_\_\_\_\_\_\_ \_\_Е. Г. Конакина\_\_\_\_\_  (Подпись) (И.О.Ф.)  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |
|  |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Тема Разработка информационной системы «Игровая библиотека»

специальность 09.02.07 группа 42919/3

Студент (ка) Шевелев Д. А.

(подпись) (ФИО)

Руководитель Журавская А. М.

(подпись) (ФИО)

Санкт-Петербург

2023

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc136454159)

[1 Общая часть 5](#_Toc136454160)

[1.1 Анализ предметной области 5](#_Toc136454161)

[1.2 Постановка задач 7](#_Toc136454164)

[1.3 Анализ методов решения 7](#_Toc136454165)

[1.4 Выбор методов и решений 10](#_Toc136454166)

[1.5 Обоснование и выбор технологий разработки 10](#_Toc136454167)

[2 Специальная часть 14](#_Toc136454171)

[2.1 Проектирование системы 14](#_Toc136454172)

[2.2 Проектирование базы данных 17](#_Toc136454176)

[2.3 Разработка пользовательского интерфейса 21](#_Toc136454179)

[2.4 Описание разработанных модулей 22](#_Toc136454180)

[2.5 Код программы 23](#_Toc136454182)

[2.6 Тестирование 23](#_Toc136454183)

[2.7 Демонстрация разработанного приложения 25](#_Toc136454185)

[3. Экономическая часть 36](#_Toc136454186)

[3.1 Область применения программного продукта и его преимущества перед аналогичным программным продуктом. 36](#_Toc136454187)

[3.2 Трудоемкость разработки программного продукта, квалификация исполнителя (лей) и его(их) оклад. 42](#_Toc136454188)

[3.3 Расчет затрат на разработку информационных технологий 44](#_Toc136454189)

[3.4 Определение цены программного продукта, который разработан одной организаций по заказу другой. 49](#_Toc136454196)

[4 Техника безопасности и охрана труда 52](#_Toc136454197)

[4.1 Анализ условий труда программиста, работающего в ООО «Омега» 52](#_Toc136454198)

[4.2 Расчёт искусственного освещения в помещении 54](#_Toc136454199)

[4.3 Электробезопасность на предприятии 56](#_Toc136454200)

[4.4 Пожарная безопасность на предприятии 59](#_Toc136454201)

[ЗАКЛЮЧЕНИЕ 61](#_Toc136454202)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 62](#_Toc136454203)

[ПРИЛОЖЕНИЕ А (обязательное) Техническое задание 64](#_Toc136454204)

[ПРИЛОЖЕНИЕ Б (обязательное) Листинг исходного кода приложения 70](#_Toc136454219)

# ВВЕДЕНИЕ

В рамках выполнения дипломного проектирования передо мной была поставлена задача разработать базу данных и окно авторизации для информационной системы «Игровая библиотека». Создание БД включало в себя реализацию сопутствующих документов и диаграмм. При разработке окна авторизации в системе необходимо было продумать обеспечение безопасности программными средствами, такими как блокировка входа в аккаунт при неверно введенных данных и капче.

После разработки окна авторизации была поставлена задача ясно продумать концепт основного функционального интерфейса информационной системы и реализовать его, проведя подробный анализ современных методов решения поставленных предприятием задач, на основе которых необходимо было остановиться на наиболее релевантном способе подхода к существующей проблеме, чтобы наиболее эффективным образом подойти к ее разрешению.

По окончании основной стадии разработки была очевидна необходимость в достойной испытательной программе для созданного программного обеспечения, чтобы обеспечить максимально бесшовное и гладкое внедрение информационной системы в устоявшуюся рабочую структуру предприятия.

# 1 Общая часть

## 1.1 Анализ предметной области

### 1.1.1 Описание предприятия Omega

Компания Omega не только внедряет готовые IТ-продукты и сервисы, представленные на рынке, и разрабатывает собственные, но и выступает кроссплатформенным интегратором и строит сложные интеллектуальные решения.

Прорывные разработки данной компании в области искусственного интеллекта, бизнес-аналитики и анализа больших данных, виртуальной и дополненной реальности, робототехники способствуют реализации управленческих решений, органично интегрируются в производственный и образовательный процесс.

На каждом этапе специалисты Омеги стремятся создать цифровую экосистему организации, города, региона и обеспечить адаптивность среды для каждого пользователя.

### 1.1.2 Описание предметной области

Информационная система — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы, которые обеспечивают и распространяют информацию

Информационная система «Игровая библиотека» предназначена для людей, желающих обмениваться своим игровым опытом с пользователями со всего мира. Выступающая в качестве начального этапа проектирования большого программного продукта, данная ИС создана как модуль, демонстрирующий, каким образом пользователи будут взаимодействовать с системой и друг с другом.

Индустрия компьютерных игр зародилась в середине 1970-х годов как движение энтузиастов и за несколько десятилетий выросла из небольшого рынка в мейнстрим с годовой прибылью в 9,5 миллиардов долларов в США в 2007 году и 11,7 миллиардов в 2008 году (согласно ежегодным отчётам ESA). На рынке работают как крупные игроки, так и небольшие фирмы и стартапы, а также независимые разработчики и сообщества (напр. Kickstarter и др.).

Современные персональные компьютеры дали множество новшеств игровой индустрии. К числу самых значимых относят звуковые и графические карты, CD- и DVD-приводы, Unix и центральные процессоры.

Звуковые карты изначально были разработаны для интегрирования качественного цифрового звука в компьютерные игры, и только потом звуковое оборудование было усовершенствовано под нужды меломанов.

Графические карты, которые на заре компьютерной эпохи эволюционировали в направлении увеличения количества поддерживаемых цветов, позже стали развиваться для аппаратной поддержки графических интерфейсов пользователя (англ. GUI) и игр. Для GUI требовалось увеличение разрешения экрана, а для игр — ускорение трёхмерной графики.

Изначально CD и DVD были разработаны как недорогой и достаточно надежный способ хранения и распространения любых данных. Впоследствии, когда эти технологии стали применяться в компьютерных играх, началось их развитие в сторону увеличения скорости чтения данных.

«Геймерское» сообщество – это группа людей, объединенная общим интересом – видеоиграми. Данная индустрия сегодня развивается колоссальными темпами, и, тем самым, привлекает в сообщество миллионы человек год за годом.

На рынке игровой индустрии присутствует достаточное количество магазинов, занимающихся размещением и дистрибуцией игровых программных продуктов, но также присутствует дефицит приложений, предоставляющих возможности социализации для игроков по всему миру.

## 1.2 Постановка задач

«Игровая библиотека», получившая коммерческое наименование «GameBox» это некая «социальная сеть» для геймеров, желающих делиться своим игровым опытом и перенимать опыт чужой. На основании данной концепции можно сформировать следующие задачи разработки:

* Создать базу данных, необходимую для грамотного хранения каталогов и пользовательских данных;
* Спроектировать функциональный и удобный пользовательский интерфейс, позволяющий пользователями с разным уровнем допуска к системе комфортно работать с ней;
* Спроектировать алгоритмы и вспомогательные проекты, необходимые для устойчивой функциональности основного приложения.

## 1.3 Анализ методов решения

После исследования предметной области были выделены следующие методы реализации поставленных задач:

* Самый распространенный способ создания системы подобного назначения – это веб-приложение. Сайт, или веб-сайт, также веб-узел, — одна или несколько логически связанных между собой веб-страниц; также место расположения контента сервера. Обычно сайт в Интернете представляет собой массив связанных данных, имеющий уникальный адрес и воспринимаемый пользователями как единое целое.Не требует установки, все обновления происходят на сервере, доставляются пользователям сразу — достаточно просто перезагрузить страницу или выйти, а потом снова зайти в аккаунт. Но иногда для его работы нужно установить дополнительные библиотеки или использовать защищенные сетевые протоколы.

Работа веб-приложения зависит не только от того, насколько грамотно оно разработано и характеристик пользовательского устройства, но также от скорости интернет-соединения, работоспособности удаленного сервера. Доступно из любой точки мира, с любого устройства, а пользовательские файлы всегда будут под рукой. Но только если есть интернет-соединение или реализована возможность работы офлайн и загрузки-выгрузки данных.

Приложение, разработанное с использованием современных протоколов и средств защиты, способно полноценно обеспечивать сохранность данных. Однако на некоторые моменты разработчики не могут повлиять: браузер, облачный сервер, канал связи — могут повысить уровень безопасности за счет дополнительных средств проверки, но также снизить его за счет своих уязвимостей. Несомненный плюс для пользователей: такое ПО проще контролировать. Ограничения среды снижают вероятность, что оно скрыто получит доступ к файлам или запустит какой-либо процесс.

* Менее популярный, но более удобный способ – это создание десктопного приложения. Десктопное приложение нужно устанавливать на компьютере, обновлять каждый раз, как выходит новая версия. Несмотря на то, что чаще всего процесс автоматизирован — все равно это занимает время пользователей и ресурсы устройств. Дополнительно придется отслеживать версии на каждом компьютере, смартфоне и планшете. Десктопное приложение работает автономно, поэтому главное — качество кода и стабильность оборудования, на котором этот код выполняется. Но если связь с сервером необходима — то возникают те же проблемы, что у «конкурента».

Доступно всегда — но только с устройства, на котором оно установлено. Чтобы работать с разных устройств, его придется установить на каждом, а также придумать, где хранить файлы, чтобы всегда иметь к ним доступ. Десктопное приложение зависит от операционной системы, процессора, видеокарты, ряда других параметров. Приходится учитывать нюансы каждой среды (в том числе при «отлове» ошибок), писать код с учетом возможных вариантов, нанимать отдельных разработчиков или даже целые команды для версий под разные ОС.

Десктопное приложение настраивается более гибко, а значит — теоретически при его разработке можно предусмотреть все потенциальные уязвимости. На практике — вряд ли. Впрочем, сделать его полностью безопасным все же можно. Но только если устройство, на котором оно установлено, не будет никуда подключаться, даже к защищенной локальной сети. В противном случае — риск все равно будет.

Однозначно сказать, что безопаснее — сложно (если вообще возможно). На это влияют много факторов, прежде всего — человеческий. А ведь именно в защите от человеческого фактора, в различных его проявлениях, заключается смысл всех мер безопасности.

* Мобильное приложение, взаимодействующее с базой данных.

Мобильное приложение – это программный пакет, функционал и дизайн которого «заточен» под возможности мобильных платформ. Перечислим несколько основных плюсов приложения:

1. Интерфейс программы создан конкретно под работу на мобильном устройстве через сенсорный экран или кнопки;
2. Удобная и понятная для пользователей гаджетов навигация, мобильное меню;
3. Лучшее взаимодействие с пользователем через сообщения, пуш-уведомления, напоминания. Приложение может выполнять функции даже в фоновом режиме, чего нельзя сказать о сайте. Для работы с программой не нужно открывать браузер, а многие приложения поддерживают ряд функций и при отключенном интернете;
4. Хранение персональных данных пользователя. Эта функция расширяет возможности персонализации приложений. Например, вызывает такси на дом (прописка), записывает на прием к врачу по медицинскому полису и другие преимущества;
5. Более гибкая обратная связь с компанией, сервисом;
6. Можно задействовать больше ресурсов. Например, подключить геолокацию и вызывать машину в любую точку города;
7. Приложения могут учитывать биологические ритмы человека и оповещать его о необходимости следовать режиму.

## 1.4 Выбор методов и решений

На основе вышеперечисленных методов была выбрана следующая стратегия разработки – десктопное приложение, связанное с базой данных.

## 1.5 Обоснование и выбор технологий разработки

### 1.5.1 Выбор и описание СУБД

СУБД, на которой реализована база данных – Microsoft SQL Server. 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.

Microsoft SQL Server также поддерживает Open Database Connectivity (ODBC) — интерфейс взаимодействия приложений с СУБД. Версия SQL Server 2005 обеспечивает возможность подключения пользователей через веб-сервисы, использующие протокол SOAP. Это позволяет клиентским программам, не предназначенным для Windows, кроссплатформенно соединяться с SQL Server. Microsoft также выпустила сертифицированный драйвер JDBC, позволяющий приложениям под управлением Java (таким как BEA и IBM WebSphere) соединяться с Microsoft SQL Server 2000 и 2005.

SQL Server поддерживает зеркалирование и кластеризацию баз данных. Кластер сервера SQL — это совокупность одинаково конфигурированных серверов; такая схема помогает распределить рабочую нагрузку между несколькими серверами. Все сервера имеют одно виртуальное имя, и данные распределяются по IP-адресам машин кластера в течение рабочего цикла. Также в случае отказа или сбоя на одном из серверов кластера доступен автоматический перенос нагрузки на другой сервер.

В SQL Server 2005 встроена поддержка .NET Framework. Благодаря этому, хранимые процедуры БД могут быть написаны на любом языке платформы .NET, используя полный набор библиотек, доступных для .NET Framework, включая Common Type System (система обращения с типами данных в Microsoft .NET Framework). Однако, в отличие от других процессов, .NET Framework, будучи базисной системой для SQL Server 2005, выделяет дополнительную память и выстраивает средства управления SQL Server вместо того, чтобы использовать встроенные средства Windows. Это повышает производительность в сравнении с общими алгоритмами Windows, так как алгоритмы распределения ресурсов специально настроены для использования в структурах SQL Server.

### 1.5.2 Выбор и описание языка программирования

Данный программный продукт реализован на языке программирования C#, с использованием интерфейса программирования Windows Forms, являющегося частью Microsoft .NET Framework.

С# популярен за счет своей «простоты». Простоты для современных программистов и больших команд разработчиков, чтобы те могли в сжатые сроки создавать функциональные и производительные приложения. Этому способствуют нетипичные конструкции языка и специфичный синтаксис, помогающий максимально органично реализовать намеченные функции.

Популярность языка – еще одно значимое преимущество. Большое количество поклонников C# способствуют его развитию. Также это благоприятно влияет на рост числа вакансий, связанных с разработкой на языке Microsoft. Программисты, хорошо знакомые с С#, востребованы в индустрии, несмотря на их большое и постоянно увеличивающееся количество.

Понятный синтаксис C# заметно упрощает не только разработку как таковую, но и другие важные аспекты совместной работы, например, чтение чужого кода. Это упрощает процесс рефакторинга и исправления ошибок при работе над приложениями в больших командах.

Язык C# практически универсален. Можно использовать его для создания любого ПО: продвинутых бизнес-приложений, видеоигр, функциональных веб-приложений, приложений для Windows, macOS, мобильных программ для iOS и Android.

Безопасность ваших программ и операционных систем обеспечивается благодаря мощным утилитам на базе C#. Колоссальное количество вирусов, на ежедневной основе атакующих компьютеры пользователей, блокируется инструментами, созданными с помощью языка Microsoft. Аналогичная ситуация наблюдается в крупном бизнесе – мировые корпорации защищаются от хакерских атак с помощью ПО, написанного на С#.

Visual C# — это реализация языка C# корпорацией Майкрософт. Поддержка Visual C# в Visual Studio обеспечивается с помощью полнофункционального редактора кода, компилятора, шаблонов проектов, конструкторов, мастеров кода, мощного и удобного отладчика и многих других средств. Библиотека классов .NET Framework предоставляет доступ ко многим службам операционной системы и другим полезным, правильным классам, что существенно ускоряет цикл разработки.

### 1.5.3 Выбор и описание среды разработки

Для удобной разработки программного продукта была выбрана среда программирования Microsoft Visual Studio.

Microsoft Visual Studio — это узкоспециализированная линейка программных продуктов от компании Microsoft, которая включает в себя интегрированную среду разработки программного обеспечения и множество других средств для разработчиков.

Visual Studio ориентированно на разработку приложений, причем в Visual Studio можно создавать как консольные приложения, так и приложения с поддержкой графического интерфейса, и с поддержкой технологии Windows Forms.

Помимо этого, в Visual Studio можно разрабатывать веб-сайты, веб-приложения, веб-службы, при этом разработанные приложения могут быть для любой платформы, которую поддерживают Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Microsoft Silverlight.

Редактор исходного кода Visual Studio с поддержкой технологии IntelliSence позволяет очень удобно отредактировать исходный код и при этом не потерять нить изменений.

Мощнейший отладчик кода может работать как с машинным, так и с исходным кодами. Также Visual Studio предоставляет множество других инструментов для работы: многофункциональный редактор форм для создания графического интерфейса приложения, мощный веб-редактор, инструмент дизайна классов и инструмент дизайна схем баз данных.

Visual Studio обладает возможностью использовать множество плагинов для расширения функциональных возможностей системы практически на каждом уровне работы, включая подключение систем контроля версий исходного кода, таких как Perforce, Subversion и Visual SourceSafe, также можно добавить огромное множество новых инструментов для работы с визуальным проектированием программных продуктов.

Дизайнер форм Visual Studio незаменим при разработке программ с графическим интерфейсом, помогая спроектировать внешний вид будущего приложения и работу каждого элемента интерфейса.

Наконец, Visual Studio предоставляет комплекс инструментов для автоматизации тестирования приложений в части проверки работы интерфейсов, модульного и нагрузочного тестирования.

# 2 Специальная часть

## 2.1 Проектирование системы

### 2.1.1 Разработка диаграммы вариантов использования

При проектировании информационной системы была разработана диаграмма вариантов использования для определения области применения создаваемого программного продукта (см. Рисунок 1).

Изображение выглядит как текст, диаграмма, зарисовка, рисунок

Автоматически созданное описание

Рисунок 1 – Диаграмма вариантов использования

### 2.1.2 Разработка диаграммы активности

При проектировании информационной системы была разработана диаграмма активности, изображенная на рисунке 2.

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 2 – Диаграмма активности

### 2.1.3 Разработка диаграммы последовательности

При проектировании информационной системы была разработана диаграмма последовательности, определяющая модель взаимодействия администратора с системой (см. Рисунок 3).

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 3 – Диаграмма последовательности

## 2.2 Проектирование базы данных

### 2.2.1 Составление ER-диаграммы

С помощью анализа данных были выявлены зависимости между ними и спроектирована ER-диаграмма, отображающая выявленные сущности данных и отношения между ними. Исходя из описания системы, диаграммы потоков данных и алгоритма работы модуля, база данных должна содержать:

* Сущность(таблицу), отображающую категории игровых продуктов;
* Сущность(таблицу), отображающую разработчиков игровых продуктов;
* Сущность(таблицу), содержащую обзоры пользователей на игры;
* Сущность(таблицу), содержащую страницы игровых продуктов;
* Сущность(таблицу), содержащую добавления игровых продуктов в коллекции пользователей;
* Сущность(таблицу), содержащую роли пользователей;
* Сущность(таблицу), содержащую подписки пользователей;
* Сущность(таблицу), содержащую общие данные о пользователях;

Эти сущности и отношения между ними проиллюстрированы на ER-

диаграмме на рисунке 4.

Изображение выглядит как текст, диаграмма, Параллельный, План

Автоматически созданное описание

Рисунок 4 – ER-диаграмма

### 2.2.2 Составление словаря данных

Для грамотной разработки была поставлена задача сформировать словарь данных для каждой сущности БД. Данные словари представлены в таблицах 1–8.

Таблица 1 – Словарь данных в т. «Game»

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип данных** | **Описание** |
| ID (PK) | int | Идентификатор игры |
| Name | varchar(50) | Название игры |
| DeveloperID (FK) | int | Идентификатор разработчика |
| Image | image | Обложка |
| AgeRating | int | Возрастное ограничение |
| CategoryID (FK) | int | Идентификатор категории |
| Price | money | Цена |
| UserScore | float | Оценка пользователей |
| PressScore | float | Оценка прессы |
| Description | varchar(500) | Описание |
| SystemRequirements | varchar(500) | Системные характеристики |
| ReleaseDate | date | Дата выхода |
| WindowsAvailable | bool | Доступность на Windows |
| LinuxAvailable | bool | Доступность на Linux |
| MacAvailable | bool | Доступность на Mac |
| Trending | bool | Нахождение в тренде |

Таблица 2 – Словарь данных в т. «Developer»

| **Название** | **Тип данных** | **Описание** |
| --- | --- | --- |
| ID (PK) | int | Идентификатор студии |
| Name | varchar(50) | Имя студии |
| ManagerEmail | varchar(50) | Контакты менеджера |
| Office | varchar(50) | Головной офис |

Таблица 3 – Словарь данных в т. «Category»

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип данных** | **Описание** |
| ID (PK) | int | Идентификатор категории |
| Name | varchar(50) | Название категории |

Таблица 4 – Словарь данных в т. «Following»

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип данных** | **Описание** |
| ID (PK) | int | Идентификатор подписки |
| HostID (FK) | varchar(50) | Идентификатор хоста подписки |
| FollowerID (FK) | varchar(50) | Идентификатор подписчика |

Таблица 5 – Словарь данных в т. «Review»

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип данных** | **Описание** |
| ID (PK) | int | Идентификатор отзыва |
| UserID (FK) | varchar(50) | Идентификатор профиля |
| GameID (FK) | int | Идентификатор игры |
| Date | date | Дата оставления отзыва |
| Text | varchar(300) | Текст отзыва |
| Score | float | Оценка |

Таблица 6 – Словарь данных в т. «Role»

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип данных** | **Описание** |
| ID (PK) | int | Идентификатор роли |
| Name | varchar(50) | Название роли |
| Description | varchar(150) | Описание роли |

Таблица 7 – Словарь данных в т. «User»

| **Название** | **Тип данных** | **Описание** |
| --- | --- | --- |
| Username (PK) | varchar(50) | Имя пользователя |
| Login | varchar(50) | Логин пользователя |
| Password | varchar(50) | Пароль пользователя |
| RoleID (FK) | int | Идентификатор роли |
| Name | varchar(50) | Настоящее имя |
| Photo | image | Фото пользователя |
| AboutMe | varchar(200) | «Обо мне» |

Таблица 8 – Словарь данных в т. «Purchase»

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип данных** | **Описание** |
| ID (PK) | int | Идентификатор добавления в коллекцию |
| UserID (FK) | varchar(50) | Идентификатор пользователя |
| GameID (FK) | int | Идентификатор игры |
| Date | date | Дата добавления |

## 2.3 Разработка пользовательского интерфейса

Интерфейс программного продукта подстраивается под уровень допуска пользователя к информационной системе. В зависимости от уровня видоизменяются формы приложения, убираются или добавляются новые элементы взаимодействия пользователя с программой.

Скриншоты функционирующего интерфейса программы при работе администратора с ИС размещены в Приложении А.

## 2.4 Описание разработанных модулей

### 2.4.1 Описание модуля хэширования паролей

Хеширование — это преобразование информации с помощью особых математических формул. В результате возникает хеш — отображение данных в виде короткой строки, в идеале — уникальной для каждого набора информации. Размер строки может быть одинаковым для информации разного объема.

Хеш — это не закодированная исходная информация. Это скорее уникальная метка, которая генерируется для каждого набора данных индивидуально.

Математические алгоритмы, по которым хешируется информация, называются хеш-функциями. Их название тоже иногда сокращают как «хеш». Хеш-функций существует очень много, они различаются методами вычислений, назначением, надежностью и другими параметрами.

Для хэширования паролей пользователей, в ИС «GameBox» была разработана хэш-функция, изображенная на рисунке 5.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 5 – Хэш-функция

## 2.5 Код программы

Текст программы выполнен в соответствии с ГОСТ 19.101–77 (СТСЭВ 1626–79) и ГОСТ 19.401–78 (СТСЭВ 3746–82), представляет собой запись программы на исходном языке программирования с необходимыми комментариями. Часть исходного кода представлена в Приложении Б.

## 2.6 Тестирование

### 2.6.1 Тест-кейсы

Перечень тест-кейсов приведен в таблице 9.

Таблица 9 – Перечень тест-кейсов

| **№** | **Заголовок** | **Шаги воспроизведения** | **Данные** | **Ожидаемый результат** | **Действительный результат** | **Статус тест-кейса** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Проверка возможности авторизации пользователя | 1. В форме авторизации ввести тестовые данные; 2. Нажать кнопку «Войти». | Логин: «DanikOnDaTrap»  Пароль: shevelev | * Открылось главное окно; * Доступны вкладки «Каталог», «Рекомендуем», «Мой профиль». | * Открылось главное окно; * Доступны вкладки «Каталог», «Рекомендуем», «Мой профиль». | Пройден |
| 2 | Проверка возможности заведения учетной записи | 1. В форме авторизации нажать на кнопку «Завести учетную запись» 2. Ввести тестовые данные 3. Нажать на кнопку «Создать учетную запись» | Логин: «test»  Пароль: test  Имя пользователя: «test» | * Форма регистрации закрылась; * Ошибка не появлялась. | * Форма регистрации закрылась; * Ошибка не появлялась. | Пройден |
| 3 | Проверка возможности добавления страницы игры | 1. В форме авторизации ввести тестовые данные; 2. Нажать кнопку «Войти». 3. Нажать на кнопку «Добавить игру» 4. Ввести произвольные данные, отвечающие требованиям системы, в поля игровой карточки 5. Нажать на кнопку «Добавить» | Логин: «DanikOnDaTrap»  Пароль: shevelev | * Форма добавления игры закрылась; * Появилось сообщение об успешном добавлении игры в каталог; * Добавленная карточка отобразилась в каталоге. | * Форма добавления игры закрылась; * Появилось сообщение об успешном добавлении игры в каталог; * Добавленная карточка отобразилась в каталоге. | Пройден |
| 4 | Проверка возможности добавления игры в личный каталог | 1. В форме авторизации ввести тестовые данные; 2. Нажать кнопку «Войти»; 3. Выбрать любую карточку, ранее не добавленную в личную коллекцию, из каталога и нажать на нее; 4. В появившемся окне нажать «Добавить в коллекцию»; 5. Закрыть окно игры; 6. Перейти во вкладку «Мой профиль». | Логин: «DanikOnDaTrap»  Пароль: shevelev | * Добавленная карточка отобразилась в личной коллекции последней в списке; * Сообщение об ошибке не возникало. | * Добавленная карточка отобразилась в личной коллекции последней в списке; * Сообщение об ошибке не возникало. | Пройден |
| 5 | Проверка возможности модерации отзывов | 1. В форме авторизации ввести тестовые данные; 2. Нажать кнопку «Войти»; 3. Выбрать любую карточку из каталога, под которой написан комментарий; 4. Нажать на комментарий левой кнопкой мыши; 5. В диалоговом окне выбрать «Yes». | Логин: «DanikOnDaTrap»  Пароль: shevelev | * Сообщение об ошибке не возникало; * Диалоговое окно закрылось; * Отзыв пропал из списка. | * Сообщение об ошибке не возникало; * Диалоговое окно закрылось; * Отзыв пропал из списка. | Пройден |

## 2.7 Демонстрация разработанного приложения

При запуске приложения пользователя встречает окно авторизации в информационной системе, выполненное в устойчивой стилизации всего проекта и содержащее поля для ввода персональных авторизационных данных. Окно авторизации изображено на рисунке 6.

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 6 – Окно авторизации

Внизу формы авторизации располагается кнопка «Завести учетную запись», нажав на нее, пользователь, не имеющий аккаунта в системе, может его завести, в отдельно открытом окне регистрации (см. Рисунок 7).

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 7 – Окно авторизации

Окно регистрации позволяет пользователю выбрать логин, пароль и имя своей будущей учетной записи. После заполнения всех полей и подтверждения регистрации, аккаунт вносится в базу данных, а текущее окно закрывается, возвращая пользователя на этап авторизации.

В окне авторизации предусмотрена защита, на случай попытки получения несанкционированного доступа к информационной системе. При введении неверных данных авторизации, пользователю для повторной попытки входа необходимо будет ввести капчу.

Капча (captcha) — это специальный защитный код, который показывается на некоторых сайтах в виде всплывающего окна или картинки. Пользователю предлагается решить простую задачу, подтвердить статус, ввести слова или цифры, ответить на вопрос.

Вид окна авторизации при неправильном вводе данных изображен на рисунке 8.

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 8 – Капча в окне авторизации

Если пользователь совершит повторную ошибку при введении данных в поля, то вход в систему будет временно заблокирован, а пользователь получит уведомление, изображенное на рисунке 9.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 9 – Уведомление о блокировке

В случае успешной авторизации появляется главное окно программы, и пользователя встречает общий каталог игровых продуктов, добавленных в систему (см. Рисунок 10).

Изображение выглядит как текст, снимок экрана, графический дизайн, мультфильм

Автоматически созданное описание

Рисунок 10 – Главное окно программы

Пользователь с уровнем допуска «Администратор» имеет кнопку «Добавить игру», другие пользователи не имеют данной возможности. Справа внизу окна располагаются имя пользователя, вошедшего в систему, и его фотография профиля, при нажатии на которую, пользователь переходит во вкладку «Мой профиль».

Вверху главного окна собраны все вкладки программы, справа вверху расположены типизированные элементы взаимодействия со списками, они унифицированы, обладают анимацией наведения и встречаются практически во всех окнах приложения.

Основное внимание пользователя отводится на библиотеку программных продуктов, занимающую большую часть окна на вкладке «Каталог». Здесь отображены все игры, внесенные в БД вручную или же через форму добавления, доступную администратору. Форма добавления игры в каталог изображена на рисунке 11.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 11 – Форма добавления

В поля формы для удобства администратора, автоматически подгружаются данные, так что при указании, например, жанра добавляемой игры, не обязательно знать соответствующий идентификатор, сохраненный в БД, программа сама конвертирует сначала в текст, отобразив в интерактивном элементе, а затем, при подтверждении добавления игры, обратно в идентификатор, готовый к хранению в соответствующей таблице.

В форме добавления реализован функционал безопасного доступа к системе, по средствам которого, при введении недостаточного количества информации о продукте и/или ее некорректности, будет выведено сообщение об ошибке (см. Рисунок 12), база данных не будет повреждена.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 12 – Сообщение об ошибке

Находясь в разных вкладках и окнах программы пользователю может быть необходимо узнать побольше об игре, на этот случай в программе реализовано окно просмотра подробной информации о продукте, изображенное на рисунке 13.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Веб-сайт

Автоматически созданное описание

Рисунок 13 – Окно просмотра информации о игре

Кроме прочей полезной информации пользователь может видеть все отзывы, написанные о конкретном игровом продукте, пользователь с уровнем допуска «Модератор» или «Администратор» имеет возможность удаления нежелательных отзывов путем клика мышью. Система удостовериться в намерениях модератора вызвав всплывающее окно подтверждения, изображенное на рисунке 14.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 14 – Окно модерации

Любой пользователь может самостоятельно написать рецензию, на любой продукт, находящийся в общем каталоге, для этого необходимо нажать на кнопку «Добавить отзыв». Будет вызвано окно составления отзыва, изображенное на рисунке 15.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 15 – Окно добавления отзыва

Также в системе реализован функционал подбора лучших игр на основе пользовательского рейтинга, для просмотра этих игровых продуктов предусмотрена вкладка «Рекомендуем», расположенная в главном окне программы и изображенная на рисунке 16.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 16 – Вкладка «Рекомендуем»

Еще одна вкладка, расположенная в главном окне программы – это «Мой профиль». При ее выборе пользователь попадает на свою личную страницу, доступную для кастомизации и редактирования, пользователь видит свою личную коллекцию игровых продуктов и список людей, на которых оформлена подписка.

Вкладка «Мой профиль» изображена на рисунке 17.

Изображение выглядит как текст, снимок экрана, человек, Веб-сайт

Автоматически созданное описание

Рисунок 17 – Вкладка «Мой профиль»

В приложении доступен просмотр профилей других пользователей, при клике на их аватары. Также реализован графический отклик на наведение указателя мыши на фотографии пользователей из списка, нажатие открывает окно просмотра страницы профиля, изображенное на рисунке 18.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 18 – Окно просмотра профиля

# 3. Экономическая часть

## 3.1 Область применения программного продукта и его преимущества перед аналогичным программным продуктом.

### 3.1.1 Steam

Steam — онлайн-сервис цифрового распространения компьютерных игр и программ, разработанный и поддерживаемый компанией Valve. Steam выполняет роль средства технической защиты авторских прав, платформы для многопользовательских игр и потокового вещания, а также социальной сети для игроков. Программный клиент Steam также обеспечивает установку и регулярное обновление игр, облачные сохранения игр, текстовую и голосовую связь между игроками.

На начало 2021 года через Steam предлагалось к продаже свыше 50 тысяч игр, и это количество увеличивалось ежегодно на 8—10 тысяч игр. В 2020 году число активных пользователей сервиса, использовавших его хотя бы один раз в месяц, превышало 120,4 миллиона.

C 2016 года была утверждена премия в области компьютерных игр Steam Awards, в рамках которой отбираются лауреаты посредством голосования среди пользователей службы цифровой дистрибуции.

Steam Community — 12 сентября 2007 года было выпущено обновление для Steam (официально названное Steam Community), призванное облегчить общение между пользователями сервиса. Steam Community позволяет создавать свои персональные web-странички в системе Steam, создавать группы и вступать в них. В каждой группе есть чат, число участников в котором отображается на странице группы. Статус каждого участника отображается на его страничке: «В сети», «Отключён от сети», «Отошёл», «Занят», «В игре» (с указанием названия игры) и «Сплю» (недоступный для самостоятельного выбора статус, выставляющийся автоматически после двухчасового бездействия клавиатуры и мыши). Также на личной страничке отображается статистика времени игры за последние две недели, отдельно для каждой игры. Время, проведённое за No-Steam игрой, в статистике не отображается. Чат в Steam был значительно переработан к выходу Steam Community. Была добавлена возможность голосового общения, а возможное число участников возросло до нескольких десятков (ранее было возможным общение только один-на-один). Главным обновлением Steam-чата является возможность использовать его в любой игре, даже если она не продаётся в Steam: если в меню Steam разрешена соответствующая опция, определённой комбинацией клавиш (по умолчанию Shift+TAB) открывается полупрозрачное меню Steam Overlay — интерфейс Steam Community поверх изображения игры.

### 3.1.2 Epic Games Store

Epic Games Store (сокращённо EGS) — онлайн-сервис цифрового распространения компьютерных игр, разработанный и управляемый американской компанией Epic Games.

Магазин доступен в качестве веб-сайта и через приложение Epic Games Launcher. В данный момент у него нет таких функций, как поддержка гарнитуры виртуальной реальности.

Собственные игры Epic Games будут доступны исключительно через данный магазин, и компания планирует финансировать релизы только через данную площадку, используя гарантии дохода для разработчиков, которые выбирают данные условия. При этом Epic платит разницу, если игра не оправдывает ожиданий по доходам.

Для популяризации собственного магазина Epic Games устроила акцию, в которой предлагает по одной или несколько бесплатных игр в неделю с 2018 года.

В магазине планируется ввести систему облачного сохранения, но Epic Games не планирует добавлять какие-либо «игровые функции», вроде карточек Steam, предназначенные для стимулирования продаж.

Там, где это возможно, Epic Games планирует расширить свою программу «Поддержка создателя», запущенную в Fortnite Battle Royale, на другие игры, предлагаемые в магазине. Игроки смогут поддержать стримера или создателя контента, выбранного компанией на основе представленных приложений. Поддерживаемые авторы смогут получать доход от микротранзакций, совершенными через Epic Games Store от игроков, которые их поддерживают, стимулируя создателей контента. В Fortnite создатели получили около 5 % от стоимости микротранзакции.

До Epic Games Store основным онлайн дистрибьютором был Steam от Valve, хотя существовали и другие конкуренты, вроде GOG.com, Origin или Uplay. Valve берет комиссию в 30% от всех игр, что сопоставимо с комиссией других сетей цифровой дистрибуции, такими как App Store. В августе 2017 года Тим Суини от лица Epic Games сказал, что комиссия в 30% больше не является разумной, и что Valve всё ещё сможет получать прибыль, если сократит её до 8%.

В начале декабря 2018 года Epic Games объявила, что откроет цифровой магазин, чтобы бросить вызов Steam. Epic планирует конкурировать с Valve за долю прибыли разработчиков игр: 88% от Epic против 70% от Valve. Epic также отказалась бы от платы, взимаемой за лицензирование движка Unreal Engine для создания игры, если игра была опубликована в Epic Game Store. По подсчётам Epic Games, магазин все равно будет получать прибыль. Epic также заявила, что не будет налагать ограничения на управление цифровыми правами на игры, продаваемые через её платформу.

Магазин открылся несколько дней спустя, 6 декабря 2018 года, в рамках церемонии награждения The Game Awards 2018, с несколькими играми и коротким списком предстоящих проектов. Ожидается, что магазин откроется для MacOS и Windows, а затем перейдет на Android и другие платформы. Epic стремится выпустить магазин для устройств Android, минуя Google Play Store, где комиссия будет составлять 12 % вместо 30 % у Google. Монополия Apple Inc. на iOS в настоящее время делает невозможным выпуск Epic Games Store в обход официального магазина App Store (в связи с чем проходили судебные разбирательства с 13 августа 2020 года до 10 сентября 2021 года). Аналитики считают, что, если Google уменьшит комиссию в Google Play с 30 %, Apple тоже придется её уменьшить.

Магазин Epic Games был анонсирован через несколько дней после того, как Valve сообщила об изменении модели распределения доходов в Steam, которое сократило комиссию с 30 % до 25 %, когда игра заработала более 10 миллионов долларов США и до 20 % после 50 миллионов долларов США. Несколько разработчиков инди-игр выразили обеспокоенность тем, что это изменение призвано помочь сохранить крупных разработчиков и издателей ААА, и мало что сделало для поддержки более мелких разработчиков. Таким образом, когда было объявлено о создании магазина Epic Games Store, несколько журналистов посчитали его потенциально разрушительным для текущей модели Steam. Некоторые разработчики и издатели объявили о планах эксклюзивно выпустить в EGS игры, которые они планировали продавать через Steam, или иметь эксклюзивное время на витрине магазина Epic до появления в других сервисах.

### 3.1.3 GOG

GOG.com (ранее Good Old Games, с англ. — «старые добрые игры») — сервис цифровой дистрибуции компьютерных игр и фильмов без DRM.

Продукты можно приобретать и скачивать онлайн, и все они предоставляются без защиты. Ценовая политика изначально подразумевала две категории игр — за $5 и за $10. Но с началом продажи более новых игр, а позднее и поддержкой региональных цен, ценовых категорий появилось намного больше. На отдельные игры или подборки регулярно проводятся скидки, как на выходные и праздники, так и посреди недели.

Пользователям не требуется скачивать каких-либо клиентов или программ, для того чтобы скачивать или обновлять игры. Но присутствует опциональный менеджер загрузок, который позволяет автоматизировать закачку и проверку правильности скачанных установщиков игр. Все игры можно хранить на любых носителях и скачивать заново неограниченное количество раз. Скачанные игры никак не привязаны к пользовательскому профилю на GOG.com, но юридически всё ещё попадают под пользовательское соглашение.

Для того чтобы игры нормально работали на новейших версиях операционных систем и аппаратного обеспечения, некоторые игры обновлены и предоставляются с открытыми для использования программами эмуляции и улучшения совместимости, вроде ScummVM и DOSBox. Иногда, в игры уже встроены дополнительные обновления, созданные не оригинальными разработчиками — например Outcast или Dungeon Keeper 2.

Вместе с самой игрой пользователь также получает бонусные материалы, которые прилагаются к игре и список которых присутствует на странице продукта. Обычно в такие материалы входит саундтрек игры, обои, аватары и руководства. Также, на GOG работает полноценная поддержка пользователей на нескольких языках, и есть программа 30-дневной гарантии возврата денег за неработающие продукты. А для пользователей, у которых региональная цена выше цены в США, существует программа «справедливой цены», по которой пользователи получают «переплаченную» сумму обратно в виде кредита на последующие покупки.

Поскольку GOG.com, как и многие другие сервисы цифровой дистрибуции, не делает продажи публичными, узнать точную позицию сервиса, по сравнению с другими практически невозможно. Но иногда отдельные разработчики и издатели делают продажи своей игры публичными, из чего можно узнавать некоторые подробности.

В статье 11 ноября 2011 PC Gamer назвал онлайн-продажи Ведьмака 2. По их данным: Direct2Drive, Impulse[en] и Gamersgate[en] вместе продали 10 тыс. копий (4 %), GOG продал 40 тыс. копий (16 %), а Steam за тот же период продал 200 тыс. копий (80 %).

20 февраля 2013 разработчик Defender’s Quest Ларс Дусе назвал продажи своей игры за первые три месяца на 6 разных сервисах цифровой дистрибуции, включая 4 крупных сервиса и 2 способа покупки и скачивания игры напрямую от разработчика. По результатам вышло, что GOG.com был вторым крупным сервисом по продажам, уступая только Steam.

### 3.1.4 Discord

Discord (с англ. — «раздор») — кроссплатформенная проприетарная система мгновенного обмена сообщениями (мессенджер) с поддержкой VoIP и видеоконференций, предназначенная для использования различными сообществами по интересам.

Система наиболее популярна у геймеров и учащихся. Разработчиком является компания Discord Inc. (ранее — Hammer & Chisel) из Сан-Франциско. Настольное клиентское приложение на базе фреймворка Electron реализовано для Windows, macOS и Linux, мобильное приложение — для Android и iOS, также существует веб-клиент. Серверы мессенджера размещены в 11 центрах обработки данных в разных частях мира. По состоянию на 13 мая 2019 года в мессенджере было зарегистрировано более 250 млн участников при среднем ежемесячном онлайне в 56 млн пользователей.

Приложение способно организовывать голосовые конференции с настройкой канала связи и работать по принципу push-to-talk, создавать публичные и приватные чаты для обмена текстовыми сообщениями. Программа имеет браузерную версию, отличием от настольной версии является то, что режим push-to-talk работает, только если в браузере активна вкладка с приложением.

При включении режима «стример» скрывается вся личная информация, отключаются звуки и уведомления на рабочий стол. Включается режим автоматически (при запуске программ для трансляций, например, Open Broadcaster Software), также режим можно включить или отключить вручную.

### 3.1.5 Подведение итогов

Ближайшим аналогом информационной системы «Игровая библиотека» можно назвать платформу «Steam». Однако, несмотря на схожесть в некоторых аспектах, разработанная ИС, получившая собственное наименование «GameBox» нацелена не на дистрибуцию игрового ПО с последующим созданием сообщества для заинтересованных пользователей, а именно на предоставление игрокам по всему миру возможности коммуникации друг с другом, с акцентом на отзывах, рекомендациях и личном опыте игроков. «GameBox», в отличие от конкурентов, является некой социальной сетью для людей, увлекающихся игровой индустрией, что позволяет гораздо более гибко удовлетворять социальные потребности пользователей.

## 3.2 Трудоемкость разработки программного продукта, квалификация исполнителя (лей) и его(их) оклад.

Трудоемкость разработки можно определить в таблице 9.1, т. е. строка «всего» показывает общую трудоемкость разработки.

Таблица 9.1 – Трудоемкость разработки программного продукта

| **Наименование этапа** | **Условное обозначение** | **Трудоемкость выполнения этапа, час** |
| --- | --- | --- |
| Описание задания | Оз | 6 |
| Написание технического задания | Нтз | 9 |
| Анализ методов решений | Амр | 15 |
| Разработка функционала приложения | Рф | 90 |
| Разработка интерфейса приложения | Ри | 152 |
| Отладка и тестирование приложения | Оит | 65 |
| Оформление документации | Оф | 15 |
| Всего (общая трудоемкость разработки) | | 280 |

Разработчики программного продукта представлены в таблице 9.2.

Таблица 9.2 – Разработчики программного продукта

|  |  |  |  |
| --- | --- | --- | --- |
| **Исполнители** | **Оклад, руб.** | **Часовая тарифная ставка, руб.** | **Количество сотрудников** |
| Разработчик-программист | 70000 | 397,72 | 1 |

Часовая тарифная ставка определяется исходя из месячного оклада, количества рабочих дней в месяце и продолжительности рабочего дня по формуле:

ЧТС = , руб./час. (1)

где ЧТС – часовая тарифная ставка, руб./час.;

Ом – оклад исполнителя в месяц, руб. /мес.;

Д – количество рабочих дней в месяце (для расчета принимаем (Д=22 раб. дня);

Тс -продолжительность рабочего дня (для расчета Тс=8час.).

ЧТС = = 397,72 руб./час.

## 3.3 Расчет затрат на разработку информационных технологий

Под информационными технологиями понимаются экономические информационные системы (ЭИС), программные продукты (ПП), информационные базы данных и т. д.

Расчет полных затрат на разработку проектного решения (программного продукта) осуществляется по формуле:

Зрп = Зот + Зст + Зэвм + Зсп + Зрм + Знр, руб. (2)

где Зот – затраты на оплату труда разработчика (разработчиков), руб.;

Зст – страховые взносы по оплате труда во внебюджетные фонды, руб.;

Зэвм – затраты, связанные с содержанием вычислительной техники, руб.;

Зсп – затраты на специальное программное обеспечение, руб.;

Зрм – затраты на расходные материалы, необходимые при разработке программного продукта, руб.;

Знр – затраты по накладным расходам, приходящиеся на разработку программного продукта, руб.

Исходные данные, связанные с разработкой программного продукта приведены в таблице 9.3.

Таблица 9.3 – Исходные данные

| **Наименование** | **Обозначение** | **Значение** |
| --- | --- | --- |
| Оклад разработчика | Ор | 70 000 рублей |
| Время разработки | Трп | 2 месяца |
| Машинное время разработки | Тмч | 1,5 месяца |
| Коэффициент дополнительной заработной платы | Кд | 0,12 |
| Коэффициент страховых взносов | Кст | 0,3 |
| Количество условных ед. используемой техники | Q | 1 шт. |
| Стоимость машино-часа эксплуатации оборудования | Cм-ч | 10 руб./час |
| Коэффициент готовности техники | Кгт | 0,95 |
| Число рабочих дней в месяце | ЧРД | 22 дня |
| Продолжительность смены | Тсм | 8 |
| Коэффициент сменности | Ксм | 1 |
| Коэффициент транспортных расходов | Кт | 0,15 |
| Коэффициент накладных расходов | Кнр | 0,54 |

**3.3.1 Затраты на оплату труда разработчиков (Зот), руб.**

Размер фонда оплаты труда разработчиков (Зот) рассчитывается по

формуле:

Зот = Ор × Трп (1+Кд), руб. (3)

где Ор – месячный оклад разработчика проектного решения, руб./мес.;

Трп – время разработки проектного решения разработчиком, мес. включает в себя машинное время работы над проектом (Тмрп);

Кд – коэффициент дополнительной заработной платы разработчика;

Зот = 70 000 × 2 (1+0,12) = 156 800 рублей.

**3.3.2 Затраты по страховым вносам (Зст), руб.**

Сумма страховых взносов определяется по формуле:

Зст = Кст × Зот, руб (4)

где Кст – коэффициент страховых взносов или процентная ставка, установленная государством, для расчета отчислений во внебюджетные фонды.

Зст = 0,3 × 156 800 = 47 040 рублей

**3.3.3 Затраты по содержанию ЭВМ (Зэвм), руб.**

Затраты, связанные с эксплуатацией и содержанием ЭВМ, определяются по формуле:

Зэвм = Тмрп × Кгт × Q× Cм-ч, руб. (5)

гдеТмрп – машинное время на разработку проектного решения, час;

Кгт – коэффициент использования техники;

Q – количество условных единиц используемой техники;

Cм-ч – стоимость машино-часа эксплуатации оборудования, руб./час.

Так как машинное время может измеряться в месяцах, а себестоимость машино-часа за один час, то машинное время необходимо перевести в часы.

Величина стоимости машинного часа(**Cм-ч**) принимается по данным организации. Перевод рабочего времени в часы осуществляется по формуле:

Тмрп = Тмч × Чрд × Тсм × Ксм, час. (6)

где Тмч – рабочее время в месяцах;

Чрд – число рабочих дней в месяце;

Тсм – продолжительность рабочей смены;

Ксм – количество рабочих смен.

Зэвм = (1,5 × 22 × 8 ×1) × 0,95 × 1 × 10 = 2 508 рублей

**3.4.4 Затраты на специальные программные продукты (Зсп), руб.**

Затраты на специальные программные продукты, необходимые для разработки проектного решения определяются по формуле:

Зсп= , руб. (7)

где *Цρ*– цена *ρ*-го специального программного средства, руб.

n **–** количество видов программных продуктов, шт.

**3.3.5 Затраты на расходные материалы (Зрм), руб.**

Затраты на расходные материалы необходимые для разработки проектного решения определяются по формуле:

Зсп = , руб. (8)

где -количество материала, шт.;

цена вида материала, руб.;

n – количество видов материалов.

Затраты на расходные материалы удобнее определять в таблице (см. таблицу 9.4).

Таблица 9.4 Затраты на расходные материалы

| **Наименование** | **Цена (без НДС)** | **Количество, шт.** | **Стоимость, руб.** |
| --- | --- | --- | --- |
| Электричество | 4,82 руб./КВт\*ч | 1170 КВт\*ч | 5639,44 |
| Бумага | 330 руб. | 2 упаковки | 660 |
| Ручка | 50 руб. | 2 шт. | 100 |
| Заправка МФУ | 4200 руб | 1 раз | 4200 |
| Итого: | | | 10 599,44 |

**3.3.6 Затраты по накладным расходам (Знр)**

Накладные расходы – это затраты организации, которые напрямую не связаны с основным производством, не входят в стоимость материалов и оплату труда.

Затраты по накладным расходамопределяются по формуле:

Знр = Кнр × Зот, руб. (9)

где Кнр – коэффициент накладных расходов (или процент) принимается для расчета по данным предприятия;

Зот – затраты по оплате труда, руб., см.формулу (2).

Зрп – полные затраты на разработку программного продукта определяются по формуле (1) и сводятся в таблицу.

Знр = 0,54 × 156 800 = 84 672 руб.

Таблица 9.5 – Смета затрат на разработку программного продукта

| **Перечень затрат** | **Условное обозначение** | **Значение, руб.** |
| --- | --- | --- |
| Оплата труда | Зот | 156 800 |
| Страховые взносы | Зст | 47 040 |
| Содержание ЭВМ | Зэвм | 2 508 |
| Специальные программы | Зсп | 0 |
| Расходные материалы | Зрм | 10 599,44 |
| Накладные расходы | Знр | 84 672 |
| Итого затрат на разработку | Зрп | 301 619,44 |

## 3.4 Определение цены программного продукта, который разработан одной организаций по заказу другой.

Цена программного продукта, который разработан одной организаций по заказу другой и не предназначен для тиражирования, определяется по формуле:

Цпп = Зрп+Пп +НДС, руб.                                            (10)

  где Цпп – цена программного продукта, руб.;

Зрп – затраты на разработку проектного решения;

Пп – планируемая прибыль (принимается ≥ 25% от суммы затрат на

разработку программного продукта, т. е. от Зрп) руб.;

НДС– налог на добавленную стоимость (ставка НДС = 20%), руб.

Цпп = 301 619,44 + 90 485,832 + 78 421 = 470 526,272 руб.

Планируемая прибыль также может быть рассчитана исходя из нормативной рентабельности программного продукта, определяемая организацией и рассчитывается по формуле:

      Пп = Зрп × Rп, руб.                                                  (11)

где Rп – коэффициент рентабельности, определенный предприятием.

Пп = 301 619,44 × 0,3 = 90 485,832 руб.

Налог на добавленную стоимость определяется по ставке, установленной государством от суммы затрат на разработку программного продукта и величины плановой прибыли по формуле:

НДС = (Зрп + Пп) × Кндс, руб.                                   (12)

гдеКндс **–**коэффициент НДС (Кндс = ставка НДС%/100%).

НДС = (301 619,44+90 485,832) × 0,18= 70 578,95 руб.

В результате полученного положительного финансового результата (прибыли) предприятие должно начислить налог на прибыль (ставка налога-20% от прибыли) по формуле:

НП = Пп × Кнп, руб.                                 (13)

НП = 90 485,832 × 0,2 = 18 097,17 руб.

Прибыль, оставшаяся на предприятии после уплаты налога на прибыль, определяется по формуле:

                     Пч = Пп - НП, руб.                                                            (14)

Пч = 90 485,832–18 097,17 = 72 388,66 руб.

В целом поступления в бюджет от реализации программного продукта составят:

                   БП = НП + НДС, руб.                                                   (15)

БП = 18 097,17 + 70 578,95 = 88 676,12 руб.

Исходя из расчетов экономической части, затраты на разработку составляют 301 619,44 рубля, цена программного продукта составляет 470 526,272 рубля, а прибыль после уплаты налога составляет 72 388,66 рублей.

ООО «Омега» готов приобрести продукт за данную цену и реализовать его в своей системе для предоставления пользователям комфорта в социализации вокруг игровых программных продуктов.

# 4 Техника безопасности и охрана труда

## 4.1 Анализ условий труда программиста, работающего в ООО «Омега»

Рабочее место программиста – это часть помещения, где он проводит большую часть рабочего времени. Совокупность показателей технических и программных средств определяет качество и производительность работы, наряду с габаритными показателями мебели и её удобством.

Программист имеет собственное рабочее место, оборудованное ПЭВМ. Рабочий стул (кресло) должен быть подъёмно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также – расстоянию спинки до переднего края сиденья.

Рабочая поверхность стола сделана из дерева. На столе размещены ЭВМ, монитор и принтер. Площадь поверхности стола предполагает размещение монитора, устройств ввода/вывода и рабочей зоны с местом расположения оперативной документации.

Помещение для работы представляет собой кабинет длиной 3,8 м, шириной 4,5 м и высотой потолков 2,5 м. Общая площадь выделенного пространства - 17,1 м2.

Для снижения нагрузки на технические средства, влекущей за собой перегрев компьютера и ноутбука, используются программные средства, позволяющие снизить нагрузку на технические средства за счёт отказа в использование ненужных служб.

Для создания и поддержания независимо от наружных условий оптимальных значений температуры, влажности, чистоты воздуха, в холодное время используется водяное отопление, в тёплое время года применяется кондиционирование воздуха.

При идентификации вредных производственных факторов было установлено, что опасными могут являться:

* электромагнитные поля и излучения от технических средств;
* шум и вибрация, источником которого является оргтехника;
* статическое электричество;
* статические перегрузки, вызываемые неподвижной работой за компьютером;
* запылённость рабочей зоны, так влажная уборка рабочей поверхности проводиться не ежедневно, а только по мере загрязнения;
* умственное перенапряжение, так как в процессе деятельности программист анализирует и перерабатывает огромное количество административных сетей, разрабатывает программные продукты, поддерживает технику в рабочем состоянии;
* перенапряжение органов зрения.

В целях выявления вредных и (или) опасных факторов производственной среды и трудового процесса, оценке уровня их воздействия на работника с учетом отклонения их фактических значений от установленных уполномоченным Правительством Российской Федерации федеральным органом исполнительной власти гигиенических нормативов условий труда и применения средств индивидуальной и коллективной защиты работников проводится специальная оценка условий на рабочих местах. Оценке подлежат все имеющиеся в организации рабочие места.

Согласно санитарным правилам СП 2.2.3670-20 "Санитарно-эпидемиологические требования к условиям труда", дисплей должен располагаться на расстоянии 60-70 см, но не менее 50 см от глаз. Между боковыми поверхностями мониторов должно быть не менее 1,2 м. При использовании жидкокристаллических дисплеев на каждый компьютер должно приходиться не менее 4,5 м2 площади. На дисплей ПК не должен попадать прямой солнечный свет во избежание бликов и повышенной нагрузки на зрение.

Поэтому в кабинете, где работает программист, стоит только один компьютер, а комплектующая оргтехника расположена достаточно далеко непосредственно от самого специалиста. Это обусловлено тем, что все копировальные аппараты, принтеры и факс в периоды своей работы выделяют не только вредные токсичные газы, но и шумы и вибрацию. Данные факторы оказывают гораздо меньшее негативное влияние, нежели излучение от компьютера, но также являются вредными и могут нанести вред здоровью сотрудника.

Для снижения напряжения с глаз, а также для снятия мышечного напряжения во время рабочего дня проводится проветривание в течение 10–15 минут каждые 1,5 часа.

Исходя из анализа рабочего места программиста ООО «Омега» можно утверждать, что оно полностью соответствует СП 2.2.3670-20 "Санитарно-эпидемиологические требования к условиям труда".

## 4.2 Расчёт искусственного освещения в помещении

Через зрительный анализатор человек получает около 80% из общего объема информации. Качество поступающей, в основном, зависит от освещения в помещении. Если оно неудовлетворительное, это приводит к утомлению организма в целом. Кроме того, нерациональное освещение может явиться причиной травматизма: плохо освещенные опасные зоны, слепящие источники света и блики от них, резкие тени ухудшают видимость настолько, что вызывают полную потерю ориентировки работающих, снижают производительность труда и увеличивают брак продукции. Поэтому необходимо достаточное искусственное освещение для рабочего места.

Условия работы для расчета показателя освещенности рабочего места программиста являются:

* помещение размером 17,1 квадратных метров;
* высота помещения 2,50 метров, длина 3,8 метра, ширина 4,5 метра;
* светильники PTF/R UNI LED (33 Вт) со светодиодными лампами в количестве 3 шт., новые;
* крупногабаритная мебель отсутствует.

Основным критерием, по которому определяется необходимое количество осветительных приборов, является нормируемый уровень освещенности.

Базовым методом проектирования осветительной установки является метод коэффициентов использования, позволяющий вручную проводить все вычислительные процедуры при решении относительно простых светотехнических задач.

По этому методу необходимое количество светильников в осветительной установке определяется с помощью следующей формулы:



где N – количество светильников, шт;

Е – нормируемая освещенность, лк, определяется по СП 52.13330.2016 «Естественное и искусственное освещение»;

S – площадь помещения, м2;

Кз – коэффициент запаса (см. Рисунок 6);

U – коэффициент использования.

Фл – световой поток одной лампы в светильнике, лм;

n – количество ламп в светильнике.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 18 – Коэффициент запаса

Коэффициент использования характеризует эффективность использования светового прибора в помещении. Для его определения необходимо знать индекс помещения φ и коэффициент отражения стен, пола и потолка.

Индекс помещения:



где  – индекс помещения;

S – площадь помещения, м2;

h1 – высота подвеса светильников над рабочей поверхностью, м;

h2 – высота рабочей поверхности, м;

а – длина помещения, м;

b – ширина помещения, м

.

Делая вывод, можно утверждать, что полученное количество светильников совпадает с числом светильников, установленных в помещении, что соответствует нормативным показателям искусственного освещения согласно СП 52.13330.2016 «Естественное и искусственное освещение».

## 4.3 Электробезопасность на предприятии

При поступлении на работу сотрудник должен пройти инструктаж по технике безопасности и электробезопасности. Работника знакомят с основными правилами по технике безопасности, предлагают внимательно прочитать действующие на предприятии инструкции, поясняя при этом отдельные правила и требования.

Инструктаж по технике безопасности при выполнении конкретной работы проводит руководитель соответствующего производственного участка, показывая инструктируемому правильные безопасные приемы работы.

Повторный инструктаж проводится для рабочих независимо от их квалификации, стажа и опыта работы, не реже 1 раза в 6 месяцев по программе инструктажа на рабочем месте.

При нарушении работающими правил и инструкций по технике безопасности, технологической и производственной дисциплины, а также в случаях изменения технологического процесса или вида работы проводят дополнительные инструктажи.

Все инструктажи оформляются записями в специальном журнале с указанием номеров или шифров инструкций. Журнал о проведении инструктажа хранится у руководителя подразделения.

Работник на рабочем месте должен соблюдать общие меры электробезопасности. При этом запрещается:

* прикасаться к задней панели системного блока при включенном питании;
* допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и др. устройств;
* производить самостоятельно вскрытие и ремонт оборудования;
* пользоваться неисправными электроприборами и электропроводкой;
* ремонтировать электроприборы самостоятельно;
* подвешивать электропровода на гвоздях, металлических и деревянных предметах, перекручивать провод, закладывать провод и шнуры на водопроводные трубы и батареи отопления, вешать что-либо на провода, вытягивать за шнур вилку из розетки;
* прикасаться одновременно к персональному компьютеру и к устройствам, имеющим соединение с землей (радиаторы отопления, водопроводные краны, трубы и т. п.);
* пользоваться самодельными электронагревательными приборами и электроприборами с открытой спиралью;
* наступать на переносимые электрические провода, лежащие на полу.

При работе с электроприборами и оргтехникой (персональные компьютеры, принтеры, сканеры, копировальные аппараты, факсы, бытовые электроприборы, приборы освещения):

* Автоматические выключатели и электрические предохранители должны быть всегда исправны.
* Изоляция электропроводки, электроприборов, выключателей, штепсельных розеток, ламповых патронов и светильников, а также шнуров, с помощью которых включаются в электросеть электроприборы, должны быть в исправном состоянии.
* Электроприборы необходимо хранить в сухом месте, избегать резких колебаний температуры, вибрации, сотрясений.
* Для подогрева воды пользоваться сертифицированными электроприборами с закрытой спиралью и устройством автоматического отключения, с применением несгораемых подставок.

Учитывая проводимые мероприятия посвященные электробезопасности на предприятии ООО «Омега», можно утверждать о отличной подготовке сотрудников и новоприбывших студентов-практикантов к ведению профессиональной деятельности на данном рабочем месте, а также, о минимизации рисков, связанных с опасностью электрооборудования

## 4.4 Пожарная безопасность на предприятии

Инструктаж по пожарной безопасности проводится по программе, разработанной ответственным по охране труда организации, с учетом требований стандартов, правил, норм и инструкций о мерах пожарной безопасности. Продолжительность инструктажа устанавливается в соответствии с утвержденной программой. Инструктаж по пожарной безопасности, как правило, проводится совместно с инструктажем по технике безопасности и в те же сроки.

Инструктаж по пожарной безопасности проходят все работники организации, независимо от их образования, стажа работы по данной профессии или должности, временные работники, командированные, обучающиеся и студенты, прибывшие на практику.

О проведении инструктажа по пожарной безопасности работник, проводивший инструктаж, делает запись в журнале учета проведения инструктажей по пожарной безопасности, с обязательной подписью инструктируемого и инструктирующего. При регистрации внепланового инструктажа по пожарной безопасности указывают причину его проведения.

Все работники организаций должны допускаться к работе только после прохождения противопожарного инструктажа, а при изменении специфики работы проходить дополнительное обучение предупреждению и тушению возможных пожаров в порядке, установленном руководителем.

На проведение первичного противопожарного инструктажа необходимо отводить не менее 1 ч. Инструктируемые должны ознакомиться:

* С действующими на объекте правилами пожарной безопасности и инструкциями;
* С производственными участками, наиболее опасными в пожарном отношении, где запрещается курить, применять открытый огонь;
* С возможными причинами возникновения пожара и мерами его предупреждения;
* С практическими действиями в случае возникновения пожара - вызов пожарной помощи, использование первичных средств пожаротушения, место расположения ближайшего телефона и ознакомление с правилами поведения в случае возникновения пожара, эвакуации людей и материальных ценностей.

При первичном инструктаже инструктирующий обязан рассказать о производственных установках с повышенной пожарной опасностью, мерах предотвращения пожаров и загораний, указать место курения, ознакомить вновь поступившего с имеющимися на объекте средствами пожаротушения, показать ближайший телефон и объяснить правила поведения в случае возникновения пожара.

Проведение противопожарного инструктажа в обязательном порядке должно сопровождаться практическим показом способов использования имеющихся на объекте средств пожаротушения (противогазы, респираторы, огнетушители и т. д.).

Делая вывод, очевидным является важность соблюдения всех требований пожарной безопасности на предприятии. Компания ООО «Омега» отлично справляется с поставленной задачей по минимизации производственного ущерба персоналу, путем проведения обязательных инструктажей работников, включая студентов, проходящих практическую подготовку.

Основной вывод включает в себя понимание о том, что соблюдение рассмотренных в данном разделе мероприятий позволяет снизить утомляемость и травматизм, обеспечить комфортные условия трудовой деятельности, а также использовать рабочее время на производстве ООО «Омега» максимально эффективно.

# ЗАКЛЮЧЕНИЕ

В рамках выполнения выпускной квалификационной работы было разработано программное обеспечение «GameBox», грамотное проектирование информационной системы позволило избежать накладок на этапе реализации задуманных решений и этапе испытаний созданного программного продукта.

Благодаря программе дипломного проектирования были закреплены ценные навыки разработки информационных систем и навыки взаимодействия с системой контроля версий, облегчающей разработку сложных многомодульных программных продуктов.

В заключении можно дать уверенный ответ, что все необходимые цели выпускной квалификационной работы были достигнуты.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.105-78 Общие требования к программным документам

2. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом

3. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению

4. ГОСТ 19.301-78 Программа и методика испытаний. Требования к содержанию и оформлению

5. ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению

6. ГОСТ 19.402-78 Описание программы. Требования к содержанию и оформлению

7. ГОСТ 19.505-79 Руководство оператора. Требования к содержанию и оформлению

8. ГОСТ 19.701-90 Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

9. Роберт Мартин. Чистый Код / Роберт Мартин.. : Питер, 2019. – 457 с.

10. Роберт Мартин. Чистая Архитектура / Роберт Мартин. : Питер, 2019. – 350 с.

11. Роберт Мартин. Идеальный программист / Роберт Мартин. : Питер, 2019. – 214 с.

12. Подходы к разработке геймифицированных приложений [Электронный ресурс] / Научный журнал КубГАУ, №154 : Попок Л.Е; Карзенкова О.С. – Электрон. дан. -М.: КубГАУ, 2019. – Режим доступа: https://cyberleninka.ru/article/n/podhody-k-razrabotke-geymifitsirovannyh-prilozheniy, свободный. – Загл. с экрана.

13. Обзор современных средств для разработки мобильных приложений / Новые информационные технологии в автоматизированных системах, №17 : Аксенов К. В. – Электрон. дан. -М: НИУ ВШЭ, 2014. – Режим доступа: https://cyberleninka.ru/article/n/obzor-sovremennyh-sredstv-dlya-razrabotki-mobilnyh-prilozheniy, свободный. – Загл. с экрана.

14. Bernhard, Rumpe A Framework for Realtime Online Auctions / Rumpe Bernhard, Wimmel Guido. — Текст : непосредственный // Managing Information Technology in a Global Economy. Proceedings of IRMA International Conference. — 2001 : ,Idea Group Publishing, 2001. — С. 13.

15. Paul, Weinberg SQL The Complete Reference / Weinberg Paul, Groff James, Oppel Andrew. — 3-е изд. — New York : The McGraw-Hill Companies, 2010. — 882 c. — Текст : непосредственный.

16. Плещев, В. В. Сравнительный анализ программных средств разработки приложений и баз данных и индивидуализация учебного процесса их изучения / В. В. Плещев. — Текст : непосредственный // Известия УрГЭУ. — 2003. — № 7 — С. 69.

17. Александреску, А. Язык программирования D / А. Александреску. — М. : Символ, 2013. — 536 с.

18. Ашарина, И. В. Основы программирования на языках С и С++: Курс лекций для высших учебных заведений / И. В. Ашарина. — М. : Гор. Линия –Телеком, 2012. — 208 c.

19. Баженова, И. Ю. Языки программирования : учебник для студентов учреждений высш. проф. образования / И. Ю. Баженова ; под ред. В. А. Сухомлин. — М. : ИЦ Академия, 2012. — 368 c.

20. Руководство по WPF / [Электронный ресурс] // GitHub: [сайт]. — URL: https://github.com/dotnet/wpf?ysclid=l9twxpqpj761972497 (дата обращения: 29.10.2022).

21. Работа с БД SQL SERVER / [Электронный ресурс] // MICROSOFT.COM : [сайт]. — URL : https://learn.microsoft.com/ru-ru/sql/relational-databases/databases/view-a-list-of-databases-on-an-instance-of-sql-server?view=azuresqldb-mi-current (дата обращения: 29.05.2023).

## ПРИЛОЖЕНИЕ А

(обязательное)

Техническое задание на дипломное проектирование

студента группы 42919/3 Шевелева Даниила

* 1. Введение
     1. Наименование программного продукта

Полное наименование программного продукта: «Игровая библиотека «GameBox».

* + 1. Краткая характеристика области применения

Система предназначена для удовлетворения социальных потребностей геймерского сообщества.

* 1. Основания для разработки
     1. Основания для проведения разработки

Разрабатывается по заказу предприятия, проводившего преддипломную практику, ООО «Омега».

* + 1. Наименование темы разработки

Наименование темы разработки следующее: «Разработка ИС «Игровая библиотека».

* 1. Назначение разработки
     1. Функциональное назначение

Функциональным назначением системы является предоставление пользователю возможности просмотра общего каталога игровых продуктов, формирования собственной коллекции и взаимодействия с другими пользователями проекта по средствам оставления отзывов и оформления подписок.

* + 1. Эксплуатационное назначение

Программный продукт должен эксплуатироваться игроками, заинтересованными в социализации вокруг игровых продуктов.

* + 1. Требования к программному продукту
       1. Требования к функциональным характеристикам

Программа должна выполнять следующие функции:

* Возможность авторизации и регистрации учётной записи пользователя;
* Возможность просмотра общего каталога игровых продуктов;
* Возможность просмотра каждого игрового продукта отдельно, с подробным описанием;
* Возможность просмотра отзывов к играм;
* Возможность оставления отзывов к играм;
* Возможность добавления игры в личную коллекцию;
* Возможность поиска пользователей по их идентификатору;
* Возможность оформления подписки на пользователя;
* Возможность просмотра профиля пользователя;
* Возможность оформления отписки от пользователя;
* Возможность просмотра всех отзывов пользователя;
* Возможность просмотра личной коллекции пользователя;
* Возможность добавления игры в общий каталог, для пользователей с соответствующим уровнем допуска к информационной системе;
* Возможность модерации отзывов, для пользователей с соответствующим уровнем допуска к информационной системе;
* Возможность кастомизации личного профиля;
* Возможность просмотра списка рекомендуемых игровых продуктов, сформированного автоматически на основе рейтинга игры, внесенной в каталог.
  + - 1. Требования к ограничениям

Система должна осуществлять следующие ограничения:

* На этапе авторизации пользователя необходимо ограничить количество попыток введения данных до временного отключения возможности использования системы;
* Фото профиля должно быть загружено в формате jpeg или png.
  + - 1. Требования к надежности

Требования к надежности включают в себя следующие:

* Информационная система должна осуществлять автоматический контроль вводимых данных, при неправильных действиях программа предупредит об ошибке.
* Программа должна иметь обработку исключительных ситуаций с возможностью продолжения сеанса работы.
  + - 1. Условия эксплуатации

Программа будет прекрасно работать от плюс 5 до плюс 35 °C при относительной влажности 90 % и атмосферном давлении 462 мм.рт.ст., поскольку такие условия приблизительно соответствуют условиям эксплуатации современных компьютеров непромышленного исполнения.

* + - 1. Требования к составу и параметрам технических средств

Требования к составу технических средств для эксплуатации программы на компьютере, следующие:

* Процессор Intel Pentium 4 или более поздней версии с поддержкой SSE3;
* Оперативная память объемом 2 Гигабайт или более;
* Клавиатура;
* Указательное устройство (компьютерная мышь, трекбол или сенсорная панель).
  + - 1. Требования к информационной и программной совместимости

Для полноценного функционирования данной системы необходимо наличие операционной системы Windows.

* + - 1. Требования к маркировке и упаковке

Программа должна поставляться в виде приложения с установщиком.

* + - 1. Требования к транспортированию и хранению

Программа распространяется в электронном виде. Требования к транспортировке и хранению не предъявляются.

* + - 1. Специальные требования

Специальные требования включают в себя следующие:

* Необходимо сделать интерфейс удобным и интуитивно понятным для новых пользователей;
* Необходимо реализовать капчу в окне авторизации пользователя;
* Необходимо реализовать десктопное приложение;
* Необходимо реализовать форму подгрузку данных из БД в форме добавления игры в каталог.
  1. Технико-экономические показатели

Технико-экономические показатели не рассчитываются.

* 1. Стадии и этапы разработки

Стадии и этапы разработки изложены в таблице А.1.

Таблица А.1 – Стадии и этапы разработки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Содержание стадии | Содержание этапа | Срок 2022 г. | | Форма  отчетности |
| начало | конец |
| Техническое задание | Составление технического задания | 19.04 | 20.04 | Техническое задание |
| Рабочий проект | Проектирование программы | 21.04 | 25.04 | Схема работы системы и спецификации компонентов |
| Составление программы | 26.04 | 10.05 | Программная документация |
| Приёмо-сдаточные испытания | 29.05 | 30.05 | Протокол испытаний |
| Приёмка | Защита проекта | 19.06 | 19.06 | Оценка за дипломный проект |

* 1. Порядок контроля и приемки

Контроль должен осуществляться руководителем дипломного проекта в соответствии с пунктом А.7.

* 1. Порядок приемки

Приемка должна осуществляться с участием дипломной комиссии после проведения приёмо-сдаточных испытаний. В результате защиты дипломного проекта должна быть выставлена оценка за дипломный проект.

# ПРИЛОЖЕНИЕ Б

(обязательное)

**Листинг исходного кода приложения**

**Код формы авторизации:**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

namespace GameLibrary

{

public partial class AuthorizationForm : Form

{

SqlConnection cn;

DataTable table;

private string textCaptcha = String.Empty;

bool captchaRequired = false;

int secblocked = 0;

public AuthorizationForm()

{

InitializeComponent();

textBoxPassword.UseSystemPasswordChar = true;

ConnectionBuild();

}

private void ConnectionBuild()

{

try

{

DataBaseAPI obj = new DataBaseAPI("GameLibDB", "SHEVELEVPC\\SQLEXPRESS");

obj.OpenConnection();

cn = obj.GetConnection();

}

catch

{

MessageBox.Show("Соединение не установлено", "Ошибка");

this.Close();

}

}

private void pictureBoxShow\_Click(object sender, EventArgs e)

{

textBoxPassword.UseSystemPasswordChar = !textBoxPassword.UseSystemPasswordChar;

}

private void buttonConnect\_Click(object sender, EventArgs e)

{

string querySQL = $"SELECT Username, RoleID FROM [USER] WHERE Login = '{textBoxLogin.Text}' AND Password = '{Hashing.hashPassword(textBoxPassword.Text)}'"; // Авторизация

SqlDataAdapter adapter = new SqlDataAdapter();

SqlCommand cmd = new SqlCommand(querySQL, cn);

table = new DataTable();

adapter.SelectCommand = cmd;

adapter.Fill(table);

if (captchaRequired == false)

{

if (table.Rows.Count == 0)

{

MessageBox.Show("Данные введены неверно", "Ошибка!");

textBoxLogin.Text = "";

textBoxPassword.Text = "";

CaptchaFieldUPD();

pictureBox2.Image = this.CreateImage(pictureBox2.Width, pictureBox2.Height);

}

else

{

MWOpen();

}

}

else

{

if (table.Rows.Count != 0 && textBoxCaptcha.Text == textCaptcha)

{

CaptchaFieldUPD();

MWOpen();

}

else

{

pictureBox2.Image = this.CreateImage(pictureBox2.Width, pictureBox2.Height);

MessageBox.Show("Введены некорректные данные, возможность авторизации заблокирована на 10 секунд", "Ошибка!");

buttonConnect.Enabled = false;

timer1.Start();

}

}

}

private void CaptchaFieldUPD()

{

captchaRequired = !captchaRequired;

panelCaptcha.Visible = !panelCaptcha.Visible;

}

private void MWOpen()

{

MainWindow obj = new MainWindow(table.Rows[0].Field<string>("Username"), table.Rows[0].Field<int>("RoleID"), cn);

obj.ShowDialog();

}

private void labelRegistration\_Click(object sender, EventArgs e)

{

RegistrationForm obj = new RegistrationForm(this.cn);

obj.ShowDialog();

}

private Bitmap CreateImage(int Width, int Height)

{

Random rnd = new Random();

Bitmap result = new Bitmap(Width, Height);

int Xpos = rnd.Next(5, Width - 120);

int Ypos = rnd.Next(20, Height - 30);

Brush[] colors = { Brushes.Black,

Brushes.Red,

Brushes.RoyalBlue,

Brushes.Green };

Graphics g = Graphics.FromImage((Image)result);

g.Clear(Color.Gray);

textCaptcha = String.Empty;

string ALF = "1234567890QWERTYUIOPASDFGHJKLZXCVBNM";

for (int i = 0; i < 5; ++i)

textCaptcha += ALF[rnd.Next(ALF.Length)];

g.DrawString(textCaptcha,

new Font("Arial", 20),

colors[rnd.Next(colors.Length)],

new PointF(Xpos, Ypos));

g.DrawLine(Pens.Black,

new Point(0, 0),

new Point(Width - 1, Height - 1));

g.DrawLine(Pens.Black,

new Point(0, Height - 1),

new Point(Width - 1, 0));

for (int i = 0; i < Width; ++i)

for (int j = 0; j < Height; ++j)

if (rnd.Next() % 20 == 0)

result.SetPixel(i, j, Color.White);

return result;

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (secblocked >= 10)

{

buttonConnect.Enabled = true;

timer1.Stop();

}

else

{

secblocked++;

}

}

private void labelRegistration\_MouseEnter(object sender, EventArgs e)

{

labelRegistration.Font = new Font(labelRegistration.Font, FontStyle.Underline);

}

private void labelRegistration\_MouseLeave(object sender, EventArgs e)

{

labelRegistration.Font = new Font(labelRegistration.Font, FontStyle.Regular);

}

}

}

**Код главного окна:**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.IO;

using System.Runtime.InteropServices;

using System.Windows.Forms;

namespace GameLibrary

{

public partial class MainWindow : Form

{

string Username;

int accessLevel = 0;

int MGincrementation = 0;

int FollowingIncrementation = 0;

int currentPage = 0;

int currentPageMG = 0;

int currentPageFollowing = 0;

int minGameID = 0;

int maxGameID = 18;

bool EditingInProgress = false;

SqlConnection connection;

DataTable table;

DataTable FullProductTable;

DataTable TrendingTable;

Label[] gameLabelsName;

Label[] gameLabelsPrice;

Label[] LabelTrending;

Label[] LabelTrendingDesc;

Label[] labelsMyGames;

Label[] labelsFollowing;

PictureBox[] picBoxes;

PictureBox[] picBoxesMyGames;

PictureBox[] PBTrending;

PictureBox[] PBFollows;

PictureBox[] PlatformFirst;

PictureBox[] PlatformSecond;

PictureBox[] PlatformThird;

PictureBox[][] Platforms;

Panel[] Panels;

Panel[] PanelsTrends;

Panel[] panelsMyGames;

Panel[] PanelsFollow;

public MainWindow(string UserID, int Role, SqlConnection sqlCon)

{

InitializeComponent();

this.Username = UserID;

connection = sqlCon;

accessLevel = Role;

Panels = new[] { panel1, panel2, panel3, panel4, panel5, panel6, panel7, panel8, panel9, panel10, panel11, panel12, panel13, panel14, panel15, panel16, panel17, panel18, panel19, panel23, panel24, panel25, panel26, panel27 };

gameLabelsName = new[] { label2, label4, label6, label8, label10, label12, label14, label16, label18, label20, label22, label24, label26, label28, label30, label32, label34, label36};

gameLabelsPrice = new[] { label1, label3, label5, label7, label9, label11, label13, label15, label17, label19, label21, label23, label25, label27, label29, label31, label33, label35,};

picBoxes = new[] { pictureBox1, pictureBox2, pictureBox3, pictureBox4, pictureBox5, pictureBox6, pictureBox7, pictureBox8, pictureBox9, pictureBox10, pictureBox11, pictureBox12, pictureBox13, pictureBox14, pictureBox15, pictureBox16, pictureBox17, pictureBox18 };

PBTrending = new[] { pictureBoxTrend1, pictureBoxTrend2, pictureBoxTrend3 };

LabelTrending = new[] { labelTrend1, labelTrend2, labelTrend3 };

LabelTrendingDesc = new[] { labelTrendDesc1, labelTrendDesc2, labelTrendDesc3 };

PlatformFirst = new[] { pictureBoxPlat1, pictureBoxPlat2, pictureBoxPlat3 };

PlatformSecond = new[] { pictureBoxPlat4, pictureBoxPlat5, pictureBoxPlat6 };

PlatformThird = new[] { pictureBoxPlat7, pictureBoxPlat8, pictureBoxPlat9 };

Platforms = new[] { PlatformFirst, PlatformSecond, PlatformThird };

PanelsTrends = new[] { panel20, panel21, panel22 };

labelsMyGames = new[] { labelMP1, labelMP2, labelMP3, labelMP4, labelMP5, labelMP6 };

picBoxesMyGames = new[] { pictureBoxMP1, pictureBoxMP2, pictureBoxMP3, pictureBoxMP4, pictureBoxMP5, pictureBoxMP6 };

panelsMyGames = new[] { panel19, panel23, panel24, panel25, panel26, panel27 };

PBFollows = new[] { pictureBoxFollows1, pictureBoxFollows2, pictureBoxFollows3, pictureBoxFollows4, pictureBoxFollows5, pictureBoxFollows6 };

PanelsFollow = new[] { panelF1, panelF2, panelF3, panelF4, panelF5, panelF6 };

labelsFollowing = new[] { label40, label41, label42, label43, label44, label45 };

pictureBoxPrevMG.Image.RotateFlip(RotateFlipType.Rotate90FlipNone);

pictureBoxNextMG.Image.RotateFlip(RotateFlipType.Rotate90FlipNone);

TextBoxWatermarkExtensionMethod.SetWatermark(textBoxSearch, "Имя пользователя");

GetProfile();

SetMyGames();

SetFollows();

CatalogLoad();

SetEllipsis();

GetFullTable();

GetTrends();

SetMyProflePage();

RoundElements();

if (accessLevel == 1)

buttonAddGame.Visible = true;

}

private void GetProfile()

{

GetTableByQuery($"SELECT \* FROM [User] WHERE Username = '{Username}'");

labelUserName.Text = table.Rows[0].Field<string>("Username");

pictureBoxProfile.Image = ConvertByteArrayToImage(table.Rows[0].Field<byte[]>("Photo"));

}

private void RoundElements()

{

SetRoundedShape(panelControl, 35);

SetRoundedShape(panel29, 35);

SetRoundedShape(panelMyGames, 20);

SetRoundedShape(pictureBoxMP, 20);

SetRoundedShape(panelMyFollows, 20);

SetRoundedShape(panel28, 15);

SetRoundedShape(panel35, 15);

for (int i = 0; i < Panels.Length; i++)

{

SetRoundedShape(Panels[i], 10);

}

for (int i = 0; i < PanelsTrends.Length; i++)

{

SetRoundedShape(PanelsTrends[i], 20);

}

for (int i = 0; i < PBTrending.Length; i++)

{

SetRoundedShape(PBTrending[i], 15);

}

for (int i = 0; i < 6; i++)

{

SetRoundedShape(PanelsFollow[i], 15);

SetRoundedShape(PBFollows[i], 15);

}

}

private void SetEllipsis()

{

for (int i = 0; i < gameLabelsName.Length; i++)

{

gameLabelsName[i].AutoSize = false;

gameLabelsName[i].Size = new Size(203, 21);

gameLabelsName[i].TextAlign = ContentAlignment.MiddleLeft;

gameLabelsName[i].AutoEllipsis = true;

}

}

private void GetFullTable()

{

SqlDataAdapter adapter = new SqlDataAdapter();

SqlCommand cmd = new SqlCommand("SELECT \* FROM Game", connection);

FullProductTable = new DataTable();

adapter.SelectCommand = cmd;

adapter.Fill(FullProductTable);

}

private void GetTrendingTable()

{

SqlDataAdapter adapter = new SqlDataAdapter();

SqlCommand cmd = new SqlCommand("SELECT \* FROM Game ORDER BY UserScore DESC", connection);

TrendingTable = new DataTable();

adapter.SelectCommand = cmd;

adapter.Fill(TrendingTable);

}

private void GetTrends()

{

GetTrendingTable();

for (int i = 0; i < PBTrending.Length; i++)

{

PBTrending[i].Image = ConvertByteArrayToImage(TrendingTable.Rows[i].Field<byte[]>("Image"));

LabelTrending[i].Text = TrendingTable.Rows[i].Field<string>("Name");

LabelTrendingDesc[i].Text = TrendingTable.Rows[i].Field<string>("Description");

if (TrendingTable.Rows[i].Field<bool>("WindowsAvailable"))

{

Platforms[i][0].Visible = true;

}

if (TrendingTable.Rows[i].Field<bool>("LinuxAvailable"))

{

Platforms[i][1].Visible = true;

}

if (TrendingTable.Rows[i].Field<bool>("MacAvailable"))

{

if (TrendingTable.Rows[i].Field<bool>("LinuxAvailable") == false)

{

Platforms[i][2].Location = Platforms[i][1].Location;

}

Platforms[i][2].Visible = true;

}

}

}

byte[] ConvertImageToBytes(Image img)

{

using (MemoryStream ms = new MemoryStream())

{

img.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg);

return ms.ToArray();

}

}

public Image ConvertByteArrayToImage(byte[] data)

{

try

{

using (MemoryStream ms = new MemoryStream(data))

{

return

Image.FromStream(ms);

}

}

catch

{

return null;

}

}

private void GetTableByQuery(string sqlQ)

{

SqlDataAdapter adapter = new SqlDataAdapter();

SqlCommand cmd = new SqlCommand(sqlQ, connection);

table = new DataTable();

adapter.SelectCommand = cmd;

adapter.Fill(table);

}

private void CatalogLoad()

{

string querySQL = $"SELECT \* FROM Game WHERE ID >= {minGameID} AND ID < {maxGameID}";

GetTableByQuery(querySQL);

for (int i = 0; i < table.Rows.Count; i++)

{

try

{

Panels[i].Visible = true;

picBoxes[i].Image = ConvertByteArrayToImage(table.Rows[i].Field<byte[]>("Image"));

gameLabelsName[i].Text = table.Rows[i].Field<string>("Name");

if (table.Rows[i].Field<Decimal>("Price") == 0)

gameLabelsPrice[i].Text = "Бесплатно";

else

gameLabelsPrice[i].Text = Math.Round(table.Rows[i].Field<Decimal>("Price"), 2).ToString() + " RUB";

}

catch

{

MessageBox.Show("Внимание!", "Ошибка при загрузке каталога");

}

}

}

private void CatalogClear()

{

for (int i = 0; i < 18; i++)

{

Panels[i].Visible = false;

}

}

private void MyGamesClear()

{

for (int i = 0; i < 6; i++)

{

panelsMyGames[i].Visible = false;

}

}

private void MyFollowingClear()

{

for (int i = 0; i < 6; i++)

{

PanelsFollow[i].Visible = false;

}

}

private void pictureBox1\_MouseEnter(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

selectedPB.Location = new Point(selectedPB.Location.X - 2, selectedPB.Location.Y - 2);

selectedPB.Size = new Size(selectedPB.Width + 5, selectedPB.Height + 5);

}

private void pictureBox1\_MouseLeave(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

selectedPB.Location = new Point(selectedPB.Location.X + 2, selectedPB.Location.Y + 2);

selectedPB.Size = new Size(selectedPB.Width - 5, selectedPB.Height - 5);

SetMyGames();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

GamePageForm obj = new GamePageForm(Int32.Parse(selectedPB.Name.Substring(10))-1+(currentPage\*18), Username, accessLevel, connection);

obj.ShowDialog();

SetMyGames();

SetMyProflePage();

}

private void pictureBoxRight\_Click(object sender, EventArgs e)

{

if (pictureBox18.Image != null)

{

CatalogClear();

currentPage++;

labelCurrentPage.Text = (currentPage + 1).ToString();

UpdateRange(18);

CatalogLoad();

}

}

private void pictureBoxLeft\_Click(object sender, EventArgs e)

{

if (currentPage > 0)

{

CatalogClear();

currentPage--;

labelCurrentPage.Text = (currentPage + 1).ToString();

UpdateRange(-18);

CatalogLoad();

}

}

private void UpdateRange(int value)

{

minGameID += value;

maxGameID += value;

}

private void pictureBoxTrend1\_Click(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

GamePageForm obj = new GamePageForm(TrendingTable.Rows[Convert.ToInt32(selectedPB.Name.Substring(selectedPB.Name.Length - 1)) - 1].Field<int>("ID"), Username, accessLevel,connection);

obj.ShowDialog();

SetMyGames();

SetMyProflePage();

}

private void pictureBoxTrend1\_MouseEnter(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

selectedPB.Size = new Size(selectedPB.Width + 5, selectedPB.Height + 5);

}

private void pictureBoxTrend1\_MouseLeave(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

selectedPB.Size = new Size(selectedPB.Width - 5, selectedPB.Height - 5);

}

static void SetRoundedShape(Control control, int radius)

{

System.Drawing.Drawing2D.GraphicsPath path = new System.Drawing.Drawing2D.GraphicsPath();

path.AddLine(radius, 0, control.Width - radius, 0);

path.AddArc(control.Width - radius, 0, radius, radius, 270, 90);

path.AddLine(control.Width, radius, control.Width, control.Height - radius);

path.AddArc(control.Width - radius, control.Height - radius, radius, radius, 0, 90);

path.AddLine(control.Width - radius, control.Height, radius, control.Height);

path.AddArc(0, control.Height - radius, radius, radius, 90, 90);

path.AddLine(0, control.Height - radius, 0, radius);

path.AddArc(0, 0, radius, radius, 180, 90);

control.Region = new Region(path);

}

private void SetMyProflePage()

{

labelNameMP.Text = Username;

GetTableByQuery($"SELECT [Name] AS Игра ,[Text] AS Комментарий, Score AS Оценка, [Date] AS Дата FROM Review INNER JOIN Game ON GameID = Game.ID WHERE UserID = '{Username}'");

dataGridView1.DataSource = table;

GetTableByQuery($"SELECT \* FROM [User] WHERE Username = '{Username}'");

textBoxAboutMe.Text = table.Rows[0].Field<string>("AboutMe");

textBoxActualName.Text = table.Rows[0].Field<string>("Name");

try

{

pictureBoxMP.Image = ConvertByteArrayToImage(table.Rows[0].Field<byte[]>("Photo"));

}

catch

{

MessageBox.Show("Ошибка при загрузке аватара", "Внимание!");

}

CustomDataGrid();

}

private void CustomDataGrid()

{

dataGridView1.Font = new Font("Century Gothic", 12F);

dataGridView1.RowHeadersVisible = false;

dataGridView1.RowHeadersVisible = false;

dataGridView1.BorderStyle = BorderStyle.None;

dataGridView1.EnableHeadersVisualStyles = false;

dataGridView1.ColumnHeadersDefaultCellStyle.BackColor = Color.Gray;

dataGridView1.RowTemplate.MinimumHeight = 50;

dataGridView1.Columns[0].Width = 200;

dataGridView1.Columns[1].Width = 405;

dataGridView1.Columns[2].Width = 80;

dataGridView1.Columns[3].Width = 90;

dataGridView1.ReadOnly = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.DefaultCellStyle.SelectionBackColor = Color.Silver;

dataGridView1.DefaultCellStyle.SelectionForeColor = Color.Black;

}

private void pictureBoxMP\_Click(object sender, EventArgs e)

{

using (OpenFileDialog ofd = new OpenFileDialog() { Filter = "Image Files (\*.png \*.jpg \*.jpeg) |\*.png; \*.jpg; \*.jpeg", Multiselect = false })

{

if (ofd.ShowDialog() == DialogResult.OK)

{

pictureBoxMP.Image = System.Drawing.Image.FromFile(ofd.FileName); //

using (SqlCommand cmd = new SqlCommand("Update [User] Set Photo = @image Where Username = @currentID", connection))

{

cmd.CommandType = CommandType.Text;

cmd.Parameters.AddWithValue("@image", ConvertImageToBytes(pictureBoxMP.Image));

cmd.Parameters.AddWithValue("@currentID", Username);

cmd.ExecuteNonQuery();

}

}

}

}

private void buttonAbotMe\_Click(object sender, EventArgs e)

{

if (!EditingInProgress)

{

EditingInProgress = true;

textBoxAboutMe.ReadOnly = false;

textBoxActualName.ReadOnly = false;

buttonAbotMe.Text = "Завершить редактирование";

}

else

{

using (SqlCommand cmd = new SqlCommand("Update [User] Set Name = @name, AboutMe = @about Where Username = @currentID", connection))

{

cmd.CommandType = CommandType.Text;

cmd.Parameters.AddWithValue("@name", textBoxActualName.Text);

cmd.Parameters.AddWithValue("@about", textBoxAboutMe.Text);

cmd.Parameters.AddWithValue("@currentID", Username);

cmd.ExecuteNonQuery();

}

EditingInProgress = false;

textBoxAboutMe.ReadOnly = true;

textBoxActualName.ReadOnly = true;

buttonAbotMe.Text = "Редактировать данные профиля";

}

}

private bool SetMyGames()

{

MyGamesClear();

GetTableByQuery($"SELECT \* FROM Purchase INNER JOIN Game ON GameID = Game.ID WHERE UserID = '{Username}'");

for (int i = 0; i < 6; i++)

{

try

{

picBoxesMyGames[i].Image = ConvertByteArrayToImage(table.Rows[i + MGincrementation].Field<byte[]>("Image"));

labelsMyGames[i].Text = table.Rows[i + MGincrementation].Field<string>("Name");

panelsMyGames[i].Visible = true;

}

catch

{

return

false;

}

}

return

true;

}

private void pictureBoxNextMG\_Click(object sender, EventArgs e)

{

if (panel25.Visible == true)

{

MGincrementation += 6;

currentPageMG += 1;

labelMGCount.Text = (currentPageMG + 1).ToString();

SetMyGames();

}

}

private void pictureBoxPrevMG\_Click(object sender, EventArgs e)

{

if (currentPageMG != 0)

{

MGincrementation -= 6;

currentPageMG -= 1;

labelMGCount.Text = (currentPageMG + 1).ToString();

SetMyGames();

}

}

private bool SetFollows()

{

MyFollowingClear();

GetTableByQuery($"SELECT HostID, Photo FROM Following INNER JOIN [User] ON Username = HostID WHERE FollowerID = '{Username}'");

for (int i = 0; i < 6; i++)

{

try

{

PBFollows[i].Image = ConvertByteArrayToImage(table.Rows[i + FollowingIncrementation].Field<byte[]>("Photo"));

labelsFollowing[i].Text = table.Rows[i + FollowingIncrementation].Field<string>("HostID");

PanelsFollow[i].Visible = true;

}

catch

{

return

false;

}

}

return

true;

}

private void pictureBoxFollowingPrev\_Click(object sender, EventArgs e)

{

if (currentPageFollowing != 0)

{

FollowingIncrementation -= 6;

currentPageFollowing -= 1;

labelMyFollowsCount.Text = (currentPageFollowing + 1).ToString();

SetFollows();

}

}

private void pictureBoxFollowingNext\_Click(object sender, EventArgs e)

{

if (panelF6.Visible == true)

{

FollowingIncrementation += 6;

currentPageFollowing += 1;

labelMyFollowsCount.Text = (currentPageFollowing + 1).ToString();

SetFollows();

}

}

private void pictureBoxFollows1\_MouseEnter(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

selectedPB.Size = new Size(selectedPB.Width - 1, selectedPB.Height - 1);

}

private void pictureBoxFollows1\_MouseLeave(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

selectedPB.Size = new Size(selectedPB.Width + 1, selectedPB.Height + 1);

}

private void pictureBoxFollows1\_Click(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

ProfileForm obj = new ProfileForm(labelsFollowing[Convert.ToInt32(selectedPB.Name.Substring(17)) - 1].Text, Username, accessLevel,connection);

obj.ShowDialog();

SetFollows();

}

private void pictureBoxMP\_MouseEnter(object sender, EventArgs e)

{

pictureBoxMP.Size = new Size(pictureBoxMP.Width - 1, pictureBoxMP.Height - 1);

}

private void pictureBoxMP\_MouseLeave(object sender, EventArgs e)

{

pictureBoxMP.Size = new Size(pictureBoxMP.Width + 1, pictureBoxMP.Height + 1);

}

private void pictureBoxMP1\_Click(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

GetTableByQuery($"SELECT \* FROM Game WHERE Name = '{labelsMyGames[Convert.ToInt32(selectedPB.Name.Substring(12)) - 1].Text}'");

GamePageForm obj = new GamePageForm(table.Rows[0].Field<int>("ID"), Username, accessLevel, connection);

obj.ShowDialog();

SetMyGames();

SetMyProflePage();

}

private void pictureBoxMP1\_MouseEnter(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

selectedPB.Size = new Size(selectedPB.Width - 1, selectedPB.Height - 1);

}

private void pictureBoxMP1\_MouseLeave(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

selectedPB.Size = new Size(selectedPB.Width + 1, selectedPB.Height + 1);

}

private void buttonAddGame\_Click(object sender, EventArgs e)

{

GameAddingForm obj = new GameAddingForm(connection);

obj.ShowDialog();

}

private void pictureBoxProfile\_Click(object sender, EventArgs e)

{

tabControl1.SelectedIndex = 2;

}

private void pictureBoxLeft\_MouseEnter(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

selectedPB.Size = new Size(selectedPB.Width - 1, selectedPB.Height - 1);

}

private void pictureBoxLeft\_MouseLeave(object sender, EventArgs e)

{

PictureBox selectedPB = sender as PictureBox;

selectedPB.Size = new Size(selectedPB.Width + 1, selectedPB.Height + 1);

}

private void pictureBoxSearch\_Click(object sender, EventArgs e)

{

GetTableByQuery($"SELECT \* FROM [User] WHERE Username = '{textBoxSearch.Text}'");

if (table.Rows.Count != 0)

{

using (SqlCommand cmd = new SqlCommand($"INSERT Following VALUES ('{table.Rows[0].Field<string>("Username")}','{this.Username}') ", connection))

{

cmd.CommandType = CommandType.Text;

cmd.ExecuteNonQuery();

}

MessageBox.Show($"Вы подписались на {table.Rows[0].Field<string>("Username")}", "");

SetFollows();

}

else

{

MessageBox.Show("Пользователя с таким именем не существует!", "Ошибка!");

}

}

}

}