

Министерство образования и науки Российской Федерации

Муромский институт (филиал)

федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет

Имени Александра Григорьевича и Николая Григорьевича
Столетовых»

(МИ ВлГУ)

Факультет _____ ИТР _____

Кафедра: _____ ПИН _____

Курсовая работа

По: _____ Системы управления базами данных _____

_____ Тема: Информационная система «Steam» _____

Руководитель

(Оценка)

_____ Колпаков А.А. _____

(фамилия, инициалы)

(подпись)

(дата)

Члены комиссии

Студент _____ ПИН-122 _____

(группа)

(Подпись Ф.И.О.)

_____ Тимофеева К.В. _____

(фамилия, инициалы)

(Подпись Ф.И.О.)

(подпись)

(дата)

Муром 2024

Целью курсовой работы является разработка приложения, для систематизации данных, упрощения работы с ними и доступа к ним. В ходе работы были использованы теоретические методы в виде изучения научной литературы по созданию концептуальной модели, составлению баз данных и работы с ними в среде PostgreSQL, а также приведение базы данных к определенной нормальной форме.

Работа содержит анализ технического задания, разработку и создание базы данных, руководство для пользователя и программиста, описание и выводы после тестирования приложения, а также итоги работы, в виде готового приложения. Присутствует список используемой литературы.

Приложение написано только на русском языке. Эффективность приложения заключается в минимальном количестве ввода информации от пользователя, что сократит количество ошибок ввода информации. Также приложение не требует специальных системных ресурсов, таких как установка специальных драйверов.

The purpose of the course work is to develop an application to systematize data, simplify working with it and access it. In the course of the work, theoretical methods were used in the form of studying the scientific literature on creating a conceptual model, compiling databases and working with them in the PostgreSQL environment, as well as bringing the database to a certain normal form.

The work contains an analysis of the technical task, development and creation of a database, a guide for the user and programmer, a description and conclusions after testing the application, as well as the results of the work in the form of a ready-made application. There is a list of used literature.

The app is written only in Russian. The effectiveness of the application is to minimize the amount of information input from the user, which will reduce the number of input errors. Also, the application does not require special system resources, such as installing special drivers.

Содержание

Введение.....	5
1 Анализ технического задания	6
2 Разработка моделей данных.....	7
2.1 Описание предметной области	7
2.2 Описание структуры таблиц	8
2.3 Нормализация отношений.....	14
3 Разработка и реализация АИС	15
3.1 Интерфейс программного обеспечения.....	15
3.2 Руководство программиста	17
3.3 Руководство пользователя.....	18
4 Тестирование АИС.....	19
5 Заключение	24
6 Список литературы	25
Приложение А	26
Приложение Б	29
Приложение В.....	32

					МИВУ 09.03.04-14.000ПЗ					
Изм.	Лист	№ докум.	Подп.	Дата	Курсовая работа по теме: Информационная система «Steam»	Лит.		Лист	Листов	
Студент		Тимофеева К.В.					У		4	32
Руков.		Колпаков А.А.				МИ ВлГУ ПИН - 122				
Конс										
Н.контр.										
Утв.										

Введение

Тематика данной курсовой работы связана с разработкой приложения, для систематизации данных и упрощения работы с ними с помощью создание базы данных в среде программирования Visual Studio и PostgreSQL.

Проблема систематизации данных не утратила своего значения, поскольку работать с большим количеством информации в виде данных о каком-то определенном предмете удобнее, когда вся информация представлена в виде таблиц, где с помощью запросов и команд можно будет работать в этой таблице и вывести и найти нужную на данный момент информацию. Поэтому создание такого приложения позволит пользователю без больших знаний искать и добавлять информацию автоматически, что значительно облегчит работу пользователю.

Базы данных – организованная структура, которая предназначена для хранения информации. Базы данных могут применяться, как и в учебных заведениях, так и в банках, магазинах, на любых предприятиях. Базы данных – это информация, представленная в электронном виде, где можно осуществить быстрый доступ к информации, а также осуществлять такие действия как объединение, удаление и изменение информации, что позволяет сэкономить время и силы.

Целью курсовой работы является создание легкого и понятного в использовании приложения, результатом которого служит создание базы данных и работа с ней через приложение-клиент.

Для осуществления обозначенной цели служат следующие задачи:

- Изучение учебной, научной, справочной литературы по теме исследования
- Написание определенного алгоритма для создания приложения
- Разработка приложения
- Тестирование приложения, при введении случайных данных, даже некорректных

1 Анализ технического задания

Согласно техническому заданию на курсовую работу необходимо разработать базу данных и приложение по теме информационная система «Steam». Данное приложение предназначено для легкого редактирования базы данных и более удобного поиска и вывода нужной информации. Необходимо разработать удобный интерфейс, позволяющий исключить ошибки при введении информации от пользователя.

Основные возможности программы: занесение необходимой информации в программу, проверка адекватности внесенных данных, вывод результатов на экран, простой и удобный интерфейс для работы пользователя

Требования к программе: стабильность. Программа должна быть устойчива, даже при введении некорректных данных от пользователя.

Исходя из данных требований к приложению, база данных должна содержать несколько таблиц и данные таблицы должны быть связаны для целостности системы. Для разработки базы данных в рамках поставленной задачи можно использовать PostgreSQL, поскольку она является свободной объектно-реляционной СУБД. PostgreSQL использует для своих баз данных реляционную модель, поддерживая стандартный язык запросов SQL.

Для разработки программы, работающей с базой данных, нужно использовать объектно – ориентированный подход. Поэтому разработка клиентской части приложения будет осуществляться на языке C#, потому что он носит динамически типизированный характер, что облегчает разработчикам поиск ошибок в коде. А также C# обеспечивает строгую типизацию, что помогает избежать многих ошибок на этапе компиляции.

2 Разработка моделей данных

2. 1 Описание предметной области

Система Steam содержит игры, которые пользователь может приобрести. Каждая игра подразделена на жанры, имеет свое описание, имя разработчика, требования к игре и стоимость. Для того чтобы что-то купить, пользователю необходимо пройти авторизацию, придумать свое виртуальное имя и пароль. После покупки игры пользователь может её оценить, оставив рейтинг и, возможно, какое-то краткое впечатление о ней.

После авторизации пользователь может изменить свой учетный профиль, добавив аватар

Для разработки базы данных требуется осуществить следующие задачи:

- 1) Разработать логическое проектирование БД;
- 2) Осуществить физическое проектирование БД;

Основной задачей для первого пункта будет создание ER-модели, определение объектов взаимодействия – сущности, описание их свойств и определение отношений между объектами.

В данной диаграмме будет выделено семь сущностей.

Первая сущность называется «Пользователь». Эта сущность хранит информацию о данных пользователя, таких как ник и имя пользователя, местоположение, аватар пользователя.

Второй сущностью является «Игры». Данная сущность содержит информацию об имени разработчика, названии игры, которую пользователь хочет приобрести, цене игры и жанр, к которому относится игра.

Третья сущность «Отзывы игр». В этой сущности находится информация об имени пользователя, который оценивает игру, название игры, рейтинг игры и сам отзыв – краткое впечатление об игре от пользователя.

Четвертая сущность «Счет». В этой сущности находится информация об идентификаторе счета, сам счет и сумма на счету.

Изм.	Лист	№ докум.	Подпись	Дата

Пятой сущностью является «Разработчик». В этой сущности находится информация об разработчике.

Шестая сущность «Требования». Здесь находятся все требования к какой-либо игре.

Седьмая сущность «Купленные игры». Здесь хранятся все данные о купленных играх пользователя

В результате определения сущностей, их атрибутов и установление связей, была разработана диаграмма, изображенная на рисунке 1.

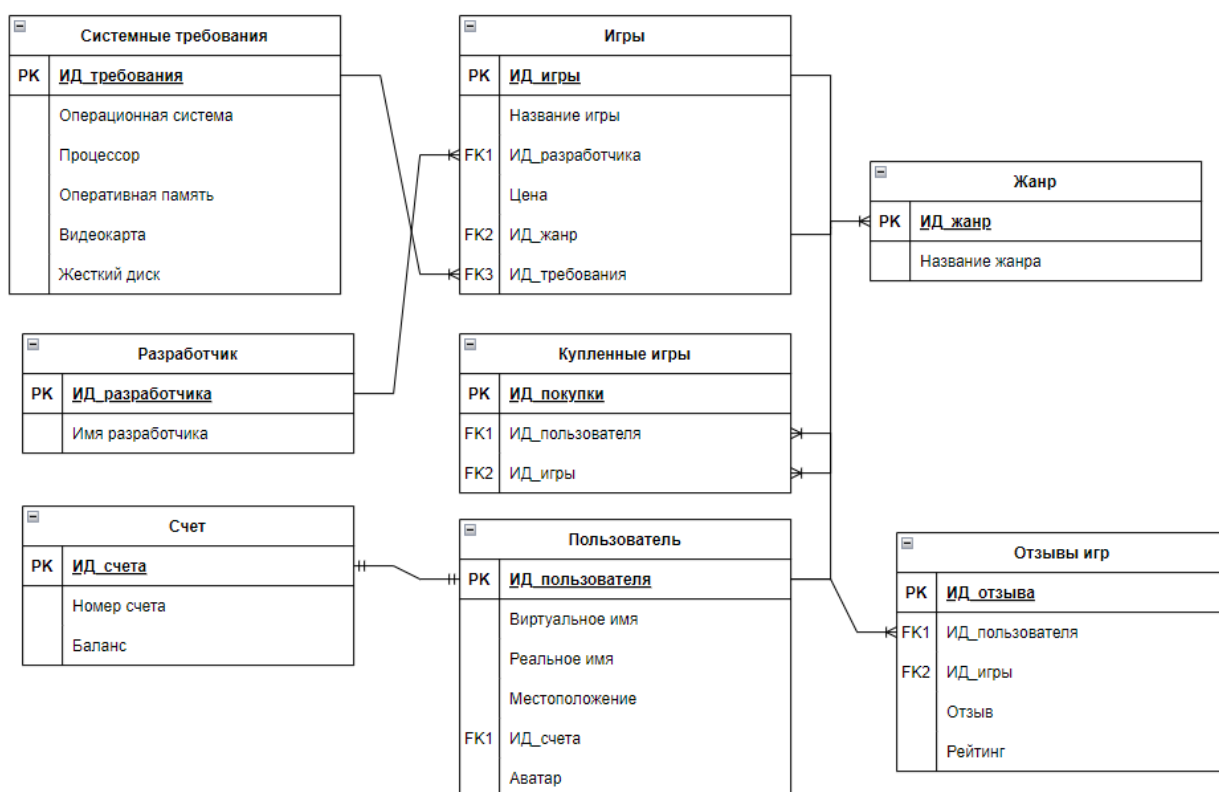


Рисунок 1– ER-модель базы данных «Steam»

2.2 Описание структуры таблиц

Для создания физической модели данных будут использованы средства PostgreSQL. На этом этапе описаны типы данных, которые используются в базе данных, имена таблиц и т.д. А также описание отношений и доменов.

Таблица «Пользователь» будет содержать следующие поля:

Изм.	Лист	№ докум.	Подпись	Дата

ИД_пользователя (ID_USER) = { натуральное число, не пустое }

Виртуальное имя (VIRTUAL_NAME) = {символьный массив длиной 30 символов, не пустое };

Реальное имя (REAL_NAME) = {символьный массив длиной 30 символов, не пустое };

Местоположение (LOCATION_USER) = { символьный массив длиной 30 символов };

ИД_счета (ID_ACC) = { натуральное число, не пустое };

Аватар (AVATAR) = { массив двоичных данных };

Описание атрибутов:

Атрибут « ИД_пользователя» содержит уникальный идентификатор пользователя

Атрибут « Виртуальное имя» содержит виртуальное имя пользователя

Атрибут «Реальное имя» содержит настоящее имя пользователя

Атрибут «Местоположение» содержит местонахождение(страна) пользователя

Атрибут «ИД_счета» содержит идентификатор счета

Атрибут «Аватар» содержит картинку, которая отображается в профиле пользователя.

В данной таблице идет следующие разграничение целостности данных:
Поле ID_USER является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Таблица «Игры» будет содержать следующие поля:

Домены:

ИД_игры (ID_GAME) = { натуральное число, не пустое };

Название (GAME) = { символьный массив длиной 30 символов, не пустое};

ИД_разработчика (ID_DEVELOPER) = { натуральное число, не пустое};

Цена (PRICE =){ натуральное число, не пустое };

ИД_Жанр (ID_GAME_GENRE) = { натуральное число, не пустое};

Изм.	Лист	№ докум.	Подпись	Дата

ID_Требования (ID_REQUIREMENTS) = { натуральное число, не пустое };

Описание атрибутов:

Атрибут «ИД_игры» содержит уникальный идентификатор игры

Атрибут «Игра» содержит название игры

Атрибут «ИД_разработчик» содержит идентификатор разработчика

Атрибут «Цена» содержит стоимость игры

Атрибут «ИД_Жанр» содержит идентификатор жанра игры

Атрибут «ИД_требования» содержит идентификатор требования игры

В данной таблице идет следующие разграничение целостности данных:
Поле ID_GAME является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Внешними ключами являются поля ID_DEVELOPER, ID_GAME_GENRE, ID_REQUIREMENTS которые ссылаются на таблицы «Разработчик», «Жанр» и «Системные требования» соответственно.

Таблица «Отзывы игр» будет содержать следующие поля:

ID_Обзора (ID_GAME_REVIEW) = натуральное число, не пустое };

ID_Пользователя (ID_USER) = натуральное число, не пустое };

ID_игры (ID_GAME)= { натуральное число, не пустое };

Отзыв (REVIEW) = { символьный массив длиной 100 символов };

Рейтинг (RATING)= { натуральное число, не пустое };

Описание атрибутов:

Атрибут «ИД_Обзора» содержит уникальный идентификатор обзора

Атрибут «ИД_Пользователя» содержит идентификатор обзора

Атрибут «ИД_игры» содержит идентификатор игры

Атрибут «Отзыв» содержит впечатление пользователя от игры

Атрибут «Рейтинг» содержит оценку игры

В данной таблице идет следующие разграничение целостности данных:
Поле ID_GAME_REVIEW является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Изм.	Лист	№ докум.	Подпись	Дата

Внешними ключами являются поля ID_USER и ID_GAME которые ссылаются на таблицы «Пользователь» и «Игра» соответственно.

Таблица «Жанр» будет содержать следующие поля:

ИД жанра (ID_GENRE_GAMES) =

Жанр (GENRE) = { символьный массив длиной 30 символов, не пустое};

Описание атрибутов:

Атрибут «ИД_жанр» содержит уникальный идентификатор жанра

Атрибут «Жанр» содержит информацию о жанре игры

В данной таблице идет следующие разграничение целостности данных:
Поле ID_GENRE_GAMES является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Таблица «Библиотека» будет содержать следующие поля:

ИД_купленной игры (ID_BUY_GAME) = { натуральное число, не пустое};

ИД_пользователя (ID_user) = { натуральное число, не пустое };

ИД_Игры (ID_game) = натуральное число, не пустое };

Описание атрибутов:

Атрибут «ИД купленной игры» содержит уникальный идентификатор купленной игры

Атрибут «ИД_пользователя» натуральное число, не пустое

Атрибут «ИД_Игры» натуральное число, не пустое

В данной таблице идет следующие разграничение целостности данных:
Поле ID_GENRE_GAMES является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Внешними ключами являются поля ID_USER и ID_GAME которые ссылаются на таблицы «Пользователь» и «Игра» соответственно.

Таблица «Требования» будет содержать следующие поля:

ИД_требования (ID_REQUIREMENTS) = { натуральное число, не пустое }

Операционная система (OS) = {символьный массив длиной 30 символов, не пустое };

Процессор (CPU) = {символьный массив длиной 30 символов, не пустое };

Оперативная память (RAM) = { натуральное число, не пустое };

Видеокарта (VIDEO_CARD) = { символьный массив длиной 30 символов, не пустое };

Жесткий диск (MEMORY) = { натуральное число, не пустое };

Описание атрибутов:

Атрибут «ИД_требования» содержит уникальный идентификатор требования

Атрибут «Операционная система» содержит информацию о требуемой системе

Атрибут «Процессор» содержит информацию о процессоре

Атрибут «Оперативная память» содержит информацию о количестве требуемой памяти

Атрибут «Видеокарта» содержит информацию о видеокарте

Атрибут «Жесткий диск» содержит информацию о количестве требуемой памяти

В данной таблице идет следующие разграничение целостности данных:
Поле ID_REQUIREMENTS является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

Таблица «Счет» будет содержать следующие поля:

ИД счета (ID_ACC) = { натуральное число, не пустое }

Счет (ACC) = { символьный массив длиной 20 символов, не пустое};

Баланс (BALANCE) = { натуральное число, не пустое }

Описание атрибутов:

Атрибут «ИД счета» содержит уникальный идентификатор жанра

Атрибут «Счет» содержит номер счета

Атрибут «Баланс» содержит количество средств на счету

Изм.	Лист	№ докум.	Подпись	Дата

В данной таблице идет следующие разграничение целостности данных:
Поле ID_ACC является первичным ключом в таблице, значение является уникальным и не содержит null-значение.

По следующей структуре была составлена физическая модель данных, изображенная на рисунке 2

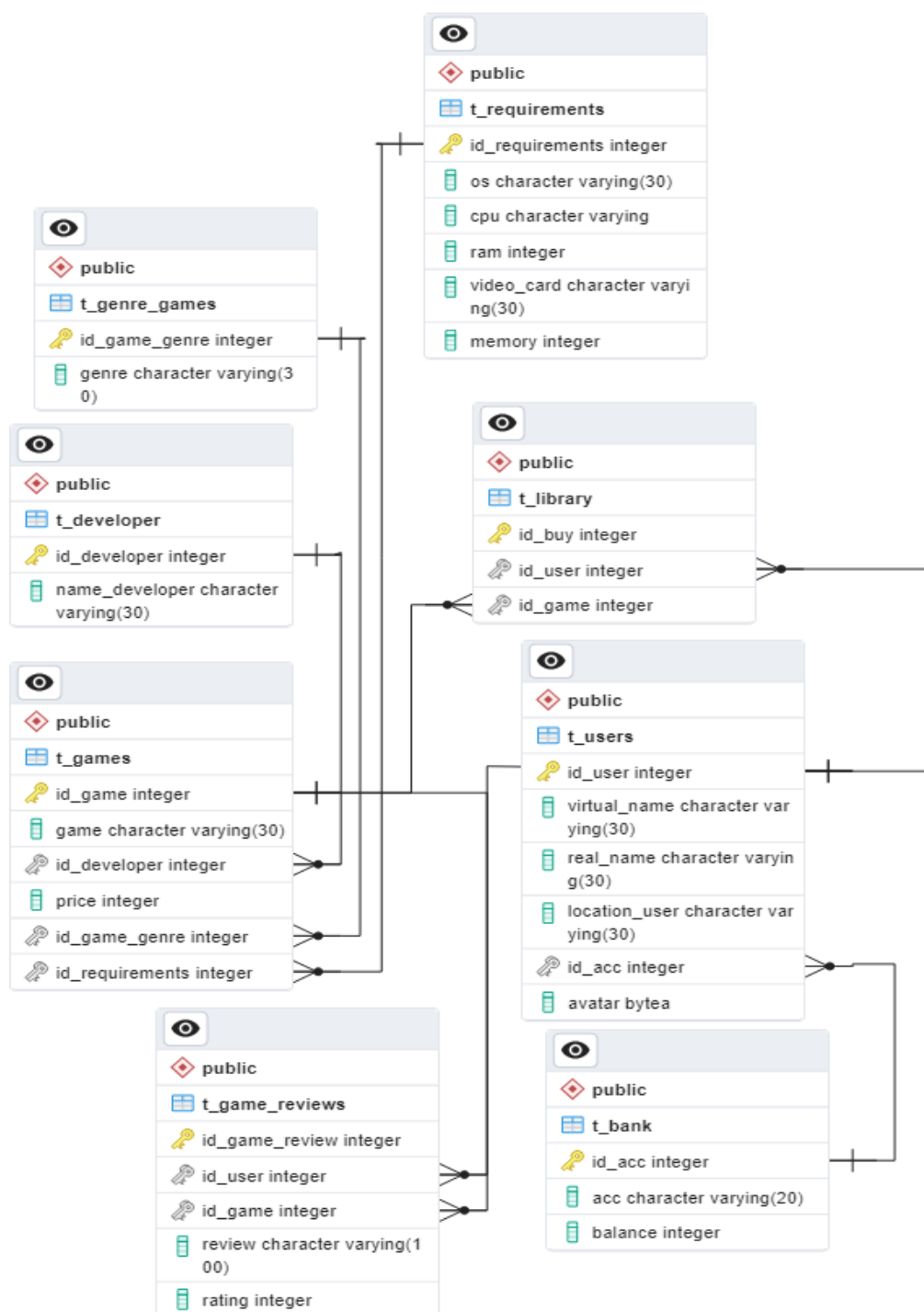


Рисунок 2– Физическая модель базы данных «Steam»

2.3 Нормализация отношений.

При приведении таблиц к нормальным формам, необходимо знать основные правила каждой формы.

1 нормальная форма: необходимость неделимости значения в каждом поле строки. Каждая строка должна быть уникально и не зависеть от других строк. Должен быть определен первичный ключ

2 нормальная форма: условие этой формы - является отсутствие зависимости не ключевых полей от части составного ключа. Все таблицы должны находиться в первой нормальной форме

3 нормальная форма: условием третьей нормальной формы является то, что не первичные атрибуты не должны быть зависимы друг от друга.

На примере таблицы «Пользователь» докажем нахождения таблиц в нормальных формах:

Отношение Пользователь(ИД_пользователя, виртуальное имя, реальное имя, местоположение, ИД_счета, аватар) находится в первой нормальной форме, т.к. все атрибуты данного отношения не являются составными. Местоположение не является составным атрибутом, т.к. записывается только страна пользователя. Ключом отношения будет являться атрибут ИД_пользователя, т.к. он является уникальным для каждого человека. Также данное отношение находится во второй нормальной форме, т.к. все атрибуты полностью зависят от первичного ключа, ключ не является составным. Отношение находится в третьей нормальной форме, т.к. каждый атрибут данной таблицы не зависит друг от друга.

Все эти условия соблюдаются и для других таблиц.

3 Разработка и реализация АИС

3.1 Интерфейс программного обеспечения

В данном пункте приведена информация о реализации приложения для информационной системы «Steam», разрабатываемой в данной курсовой работе.

При разработке приложения для АИС «Steam» были использованы следующие компоненты: Button, Label, DataGridView, Combobox, GroupBox, CheckBox.

Компонент Button - кнопка, с помощью которой пользователь взаимодействует с окнами и редактирует информацию.

Компонент Label представляет описание некоторых объектов, для помощи пользователю.

С помощью компонента dataGridView происходит вывод информации, которая содержится в базе данных

Компонент Combobox отвечает за выбор пользователя.

Компонент CheckBox отвечает за выгрузку информации.

Компонент GroupBox собирает одну группу элементов управления в определенные границы для удобства пользователя, с возможность озаглавить данную группу компонентов.

Приложение АИС «Steam» состоит из 7 окон. Самое первое окно, которое увидит пользователь, является главным. От выбора в этом окне зависят последующие открытия окон программы(2 или 3). Данные окна связаны с 1 и без выбора и подтверждения пользователя не смогут открыться.

Первым окном является окно авторизации (Приложение А. Рисунок А.1). От пользователя будет требоваться ввод данных в виде своего логина или пароля. После заполнения всех полей, пользователь сможет дальше работать в программе.

Если человек заходит под логином и паролем администратора, то ему откроется второе окно, которое содержит компонент dataGridView для вывода

Изм.	Лист	№ докум.	Подпись	Дата

таблиц, соответствующие кнопки для управления информацией в таблицах и выпадающий список с названиями таблиц. (Приложение Б Рисунок Б.5) Администратор может удалить, изменить или обновить информацию в базе данных. Для добавления данных, необходимо нажать кнопку «добавить», где будет представлено еще одно окно, где администратор может внести нужную информацию и подтвердить операцию. В случае успеха, ему будет выведено новое окно с информацией об успехе операции. Для удаления информации, администратору необходимо выделить нужную строку в базе данных и нажать кнопку «удалить». Для изменения информации есть кнопка «изменить». Данная кнопка работает только с таблицей «Игры». Когда администратор её нажмет, ему откроется еще одно окно, где будут выведены данные, которые находятся в таблице на данный момент. После введения нужных данных, необходимо подтвердить операцию. При успехе будет выведено окно с информацией, об удачном изменении данных.

Третье окно откроется, если человек зайдет в приложение, как обычный пользователь. Ему будет доступна таблица со всей информацией по играм, в которых он может изучить нужную ему информацию. Также, он сможет приобрести игры выбрав и выделив нужную игру или объект и нажав кнопку «купить». Если игра уже содержится в библиотеке пользователя, то ему будет выведено соответствующее сообщение. Чтобы пользователю было легче найти нужную игру по определенному жанру или сумме, ему доступна сортировка по цене(от меньшего к большему) и выбор игр по определенному автору или жанру. Купленные игры он может посмотреть в таблице, находящейся в его библиотеке. Там же он может посмотреть конкретно свои отзывы на игры или с помощью дополнительной функции, позволяющей увидеть отзывы ото всех пользователей. Пользователь также может пополнить баланс Максимальная сумма баланса 99999. Пользователь может оставить отзыв на понравившуюся игру. Если он поменял свое мнение, то у него есть возможность изменить отзыв или же его удалить.

3.2 Руководство программиста

Для подключения к базе данных используется класс NpgsqlConnection для установления соединения с сервером PostgreSQL. Подключение к базе данных сделано в отдельном методе ConnectToDB().

```
public void ConnectToDB()  
{  
    connectionString = «Server=localhost;Port=5432;User  
Id=postgres;Password=postgres;Database=GameCatalogyDB;»;  
    conn = new NpgsqlConnection(connectionString);  
    conn.Open();  
}
```

Для вывода информации из таблиц используется следующая конструкция: SELECT <Столбцы> FROM <Таблица>;

Например, для вывода определенной таблицы для администратора будет работать следующая конструкция:

```
table = cbTables.Text;  
command = $»select * from {table}»;
```

В данном случае знак * означает вывод всех данных из таблицы, название которой передается в переменную.

Для удаления и изменения данных используются следующие конструкции соответственно:

```
command = $»update t_game_reviews set review = '{review}', rating  
= {rating} where id_game = {IdGame} and id_user = {IdUser}»;  
command = $»delete from t_game_reviews where id_game = {IdGame}  
and id_user = {IdUser} and review = '{review}' and rating =  
{rating}»;
```

Шаблон простой конструкции удаления и изменения данных соответственно: DELETE FROM <таблица> WHERE <условия> и UPDATE <таблица> SET <столбец>='данные';

Для выполнения всех команд используется метод ExecuteNonQuery - это метод объекта Command, который выполняет команды, не возвращающие

результатирующие наборы. Для возвращения результатов используется метод ExecuteReader - метод класса SqlCommand, который выполняет SQL-выражение и возвращает строки из таблицы. Подходит для SQL-выражения SELECT.

3.3 Руководство пользователя

Чтобы запустить программу, необходимо открыть файл GameCatalogy.exe. После запуска приложения откроется окно авторизации, где будут два поля ввода – логин и пароль и две кнопки войти и регистрация.(Приложение Б Рисунок Б.1) После введения данных и нажатия кнопки войти откроется окно программы с данными. (Приложение Б Рисунок Б.3) А если пользователь не зарегистрирован, то кнопка «зарегистрироваться» перенесет его на форму регистрации(Приложение Б Рисунок Б.2)

В данном окне можно выбрать следующие функции:

- 1) Сортировка игр по цене. При нажатии на данную функцию список игр отсортируется от меньшего к большему. Сортировка происходит в режиме реального времени. Если в цене стоит «0», значит игра была уже куплена и добавлена в библиотеку;
- 2) Функция «Показать только по» предоставляет возможность показать игры по определенным жанрам или авторам, которые интересуют пользователя.

Справа от списка игр выводятся необходимые системные требования к определенной игре, где пользователь может изучить, подойдет ли игра под его систему. Снизу находится раздел с балансом пользователя, чтобы он уже мог понять, сможет ли он позволить себе эту игру и кнопка пополнить баланс. Чтобы посмотреть свои игры, необходимо нажать кнопку «моя библиотека», которая перенесет пользователя в другое окно Приложение А Рисунок А.

В новом окне можно посмотреть все приобретенные игры и отзывы на них. А также поменять свой аватар.(Приложение Б Рисунок Б.4)

Изм.	Лист	№ докум.	Подпись	Дата

4 Тестирование АИС

Приложение, которое будет удовлетворять всем условиям проверки, стабильно работать при введении разных данных от пользователя, а так же защищать информацию, доступную только администратору – является целью данной курсовой работы.

Чтобы проверить приложение на его стабильность, было проведено тестирование по следующим пунктам:

- 1) Проверка правильности введения данных во время регистрации
- 2) Проверка на попытку входа незарегистрированному пользователю
- 3) Проверка попытки пополнить баланс «пустой» суммой
- 4) Проверка попытки удалить несуществующий отзыв
- 5) Проверка корректного изменения информации от администратора

Для проверки первого условия необходимо внести некорректные данные в окно авторизации(Рисунок 3). И после введения данных нажать кнопку «Подтвердить».

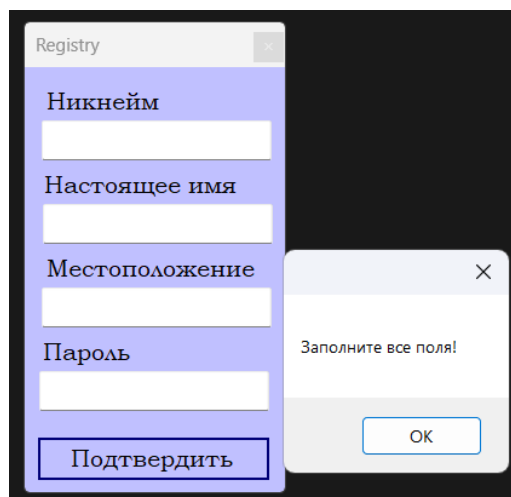


Рисунок 3 – Ошибка при вводе «пустых» значений

В ходе данной проверки, можно увидеть, что при введении некорректных данных, приложение укажет на ошибку. Ошибок не выявлено.

Проверка условия 2. Чтобы проверить данное условие, необходимо ввести несуществующие в таблице данные и нажать на кнопку «вход (Рисунок 4).

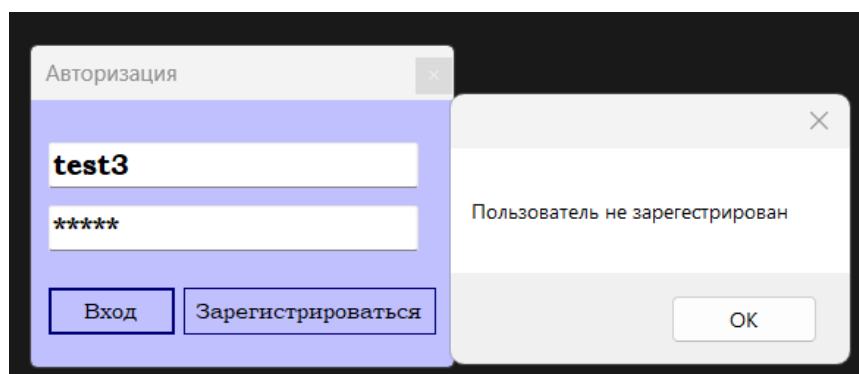


Рисунок 4 – Скриншот выполнение программы при попытке войти незарегистрированному пользователю

В ходе этой проверки, ошибок выявлено не было.

Чтобы проверить третье условие, нужно поставить пробел в поле для ввода суммы и нажать «пополнить баланс»

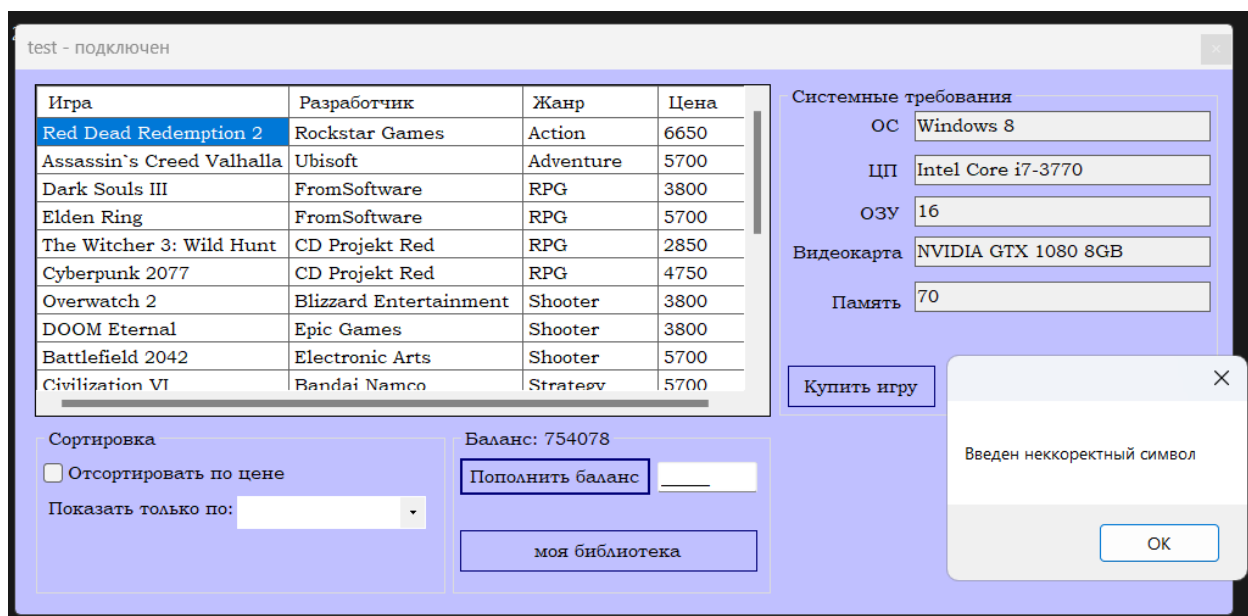


Рисунок 5 – Скриншот выполнение программы при изменении данных

В ходе этой проверки, ошибок обнаружено не было. Программа успешно выявила некорректность данных от пользователя.

Проверкой четвертого условия является удаление данных из таблицы, которых нет. Для проверки этого условия необходимо выбрать игру без отзывов и нажать кнопку «удалить отзыв». (Рисунок 6)

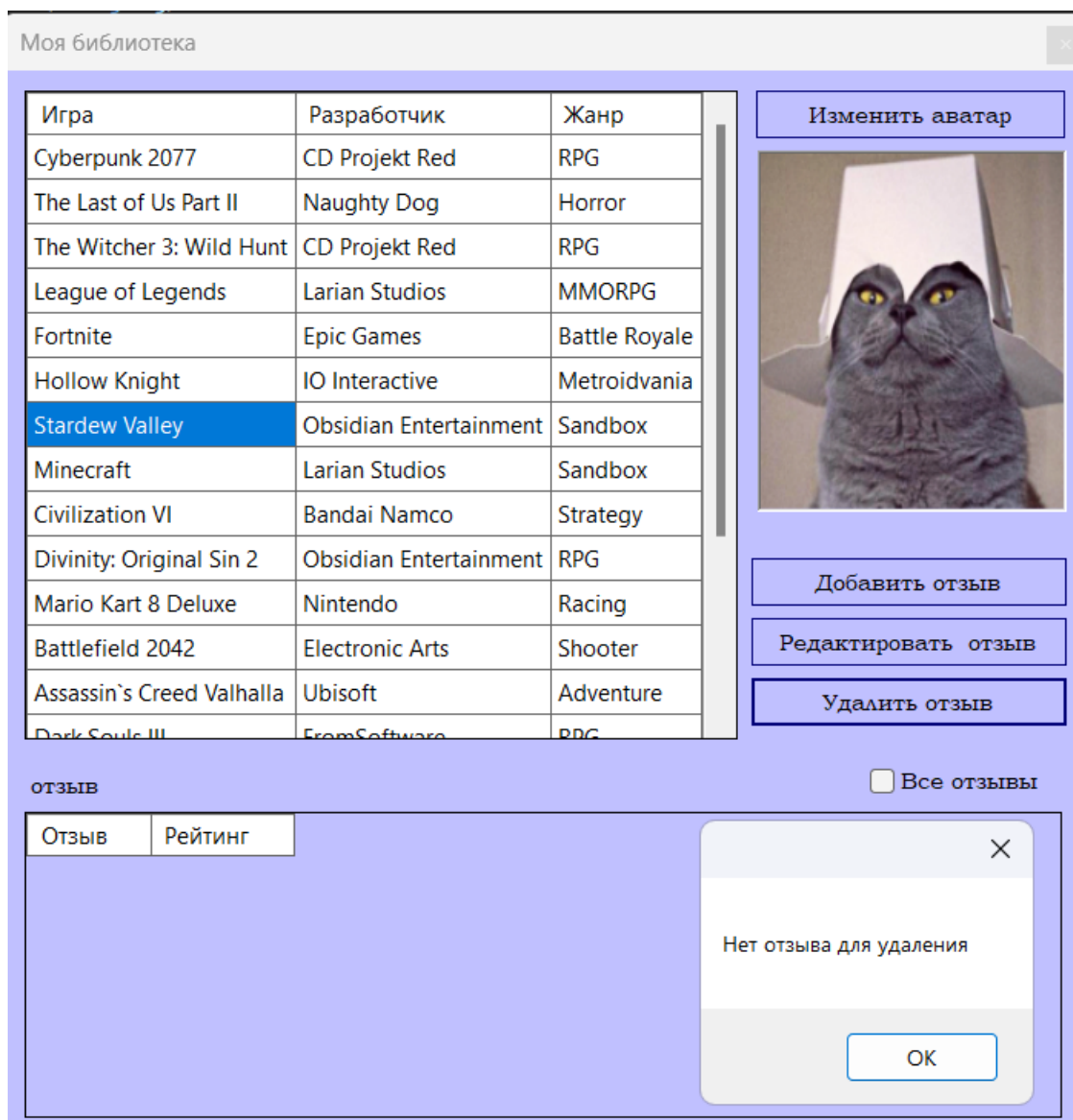


Рисунок 6– Скриншот выполнение программы при удалении данных

В результате операции было выведено сообщение о некорректных действиях пользователя.

Чтобы проверить пятое условие, необходимо зайти в программу от лица администратора, перейти к таблице с играми и с помощью кнопки «обновить» обновить информацию о игре(Рисунок 7, Рисунок 8)

Admin							
	id_developer	id_game	game	price	id_game_genre	id_requirements	state
▶	5	1	The Witcher 3: Wild Hunt	2850	3	4	Enabled
	2	2	Assassin's Creed Valhalla	5700	2	8	Enabled
	4	3	Red Dead Redemption 2	6650	1	11	Enabled
	5	4	Cyberpunk 2077	4750	3	15	Enabled
	6	5	DOOM Eternal	3800	4	7	Enabled
	9	6	The Last of Us Part II	5700	12	6	Enabled
	11	7	Dark Souls III	3800	3	3	Enabled
	18	8	Hollow Knight	1500	18	10	Enabled
	17	9	Hitman 3	5700	15	14	Enabled
	19	10	Stardew Valley	1500	17	9	Enabled
	7	11	Overwatch 2	3800	4	2	Enabled
	20	12	League of Legends	0	14	1	Enabled
	11	13	Elden Ring	5700	3	16	Enabled
	16	14	Mario Kart 8 Deluxe	4750	10	5	Enabled
	15	15	Civilization VI	5700	7	13	Enabled

Добавить
Обновить
Удалить
t_games

Рисунок 7 – Скриншот программы до изменения

Admin							
	id_developer	id_game	game	price	id_game_genre	id_requirements	state
	18	8	Hollow Knight	1500	18	10	Enabled
	17	9	Hitman 3	5700	15	14	Enabled
	19	10	Stardew Valley	1500	17	9	Enabled
	7	11	Overwatch 2	3800	4	2	Enabled
	20	12	League of Legends	0	14	1	Enabled
	11	13	Elden Ring	5700	3	16	Enabled
	16	14	Mario Kart 8 Deluxe	4750	10	5	Enabled
	15	15	Civilization VI	5700	7	13	Enabled
	14	16	Resident Evil Village	5700	11	18	Enabled
	20	17	Minecraft	2550	17	19	Enabled
	1	18	Battlefield 2042	5700	4	20	Enabled
	6	19	Fortnite	0	16	12	Enabled
	6	21	test	12345	8	15	Enabled
	19	20	Divinity: Original Sin 2	4275	3	17	Disabled
▶	5	1	The Witcher 3: Wild Hunt	3600	3	4	Enabled

Добавить
Обновить
Удалить
t_games

Рисунок 8 – Скриншот программы после изменения цены

В ходе проверки программа сработала стабильно, изменение данных прошло успешно.

Программа выполнила все условия проверки занесения данных от лица пользователя и от лица администратора. В приложении 1 представлены все данные, с которыми производились операции.

Изм.	Лист	№ докум.	Подпись	Дата

5 Заключение

В данной работе были выполнены все поставленные задачи, разработана информационная система «Steam», которая упрощает работу для добавления и редактирования данных. Программа помогает автоматизировать процесс добавления новых данных. Простой интерфейс программы, делает её понятной даже неопытному пользователю.

При разработке данного приложения были использованы такие алгоритмы как: изучение и создание SQL запросов, создание базы данных и расстановки ключей к ним.

Проект был частично прокомментирован и может эксплуатироваться по назначению.

Был проведен ряд тестов, которые показали весь функционал программы и показали стабильную работу приложения.

Изм.	Лист	№ докум.	Подпись	Дата

6 Список литературы

1. Краткий обзор языка С# [Электронный ресурс] // Microsoft – документация. 2020 URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/> (дата обращения 18.11.2024)
2. Кратка характеристика инфологической, даталогической и физической моделей [Электронный ресурс] // sdamzavas.net 2020 URL: <https://sdamzavas.net/3-6241.html> (дата обращения 25.11.2024)
3. IBExpert обзор [Электронный ресурс] // Wikipedia свободная энциклопедия 2011-2020. URL: <https://ru.wikipedia.org/wiki/IBExpert> (дата обращения 01.12.2024)
4. SQL запросы быстро. Часть 1 [Электронный ресурс] // 2006 – 2020 «Habr». URL: <https://habr.com/ru/post/480838/> (дата обращения 05.12.2024)

Изм.	Лист	№ докум.	Подпись	Дата

Приложение А

Заполненные данными таблицы

Таблица А.1

Пользователи

1	"gamerx"	"Алексей Смирнов"	"Россия"	1
2	"shadowhunter"	"Мария Иванова"	"Украина"	2
3	"prokiller123"	"Дмитрий Кузнецов"	"Казахстан"	3
4	"fastrunner"	"Анна Петрова"	"Беларусь"	4
5	"noobmaster69"	"Иван Сидоров"	"Россия"	5
6	"silentstorm"	"Ольга Павлова"	"Молдова"	6
7	"dragonslayer"	"Артем Васильев"	"Латвия"	7
8	"magicwizard"	"Екатерина Соколова"	"Эстония"	8
9	"cyberhacker"	"Максим Гончаров"	"Грузия"	9
10	"spacewalker"	"Светлана Морозова"	"Россия"	10
11	"nightwolf"	"Александр Ефремов"	"Казахстан"	11
12	"ironfist"	"Татьяна Романова"	"Беларусь"	12
13	"firephoenix"	"Сергей Попов"	"Украина"	13
14	"luckystar"	"Алина Григорьева"	"Россия"	14
15	"epicgamer"	"Игорь Захаров"	"Молдова"	15

Таблица А.2

Игры

1	"The Witcher 3: Wild Hunt"	5	3600	3	4
2	"Assassin`s Creed Valhalla"	2	5700	2	8
3	"Red Dead Redemption 2"	4	6650	1	11
4	"Cyberpunk 2077"	5	4750	3	15
5	"DOOM Eternal"	6	3800	4	7
6	"The Last of Us Part II"	9	5700	12	6
7	"Dark Souls III"	11	3800	3	3
8	"Hollow Knight"	18	1500	18	10
10	"Stardew Valley"	19	1500	17	9
11	"Overwatch 2"	7	3800	4	2
12	"League of Legends"	20	0	14	1
13	"Elden Ring"	11	5700	3	16

Таблица А.3

Требования

1	"Windows 7"	"Intel Core i3-6100"	"4"	"NVIDIA GTX 750 Ti 2GB"	"20"
2	"Windows 10"	"AMD Ryzen 3 1200"	"8"	"AMD Radeon RX 560 4GB"	"50"
3	"Windows 8"	"Intel Core i5-3470"	"16"	"NVIDIA GTX 960 2GB"	"35"
4	"Windows 11"	"AMD Ryzen 5 1600"	"8"	"NVIDIA GTX 1050 Ti 4GB"	"60"
5	"Windows 7"	"Intel Core i3-9100F"	"2"	"AMD Radeon R7 240 1GB"	"15"

Изм.	Лист	№ докум.	Подпись	Дата

МИВУ 09.03.04-19.000ПЗ

Лист

27

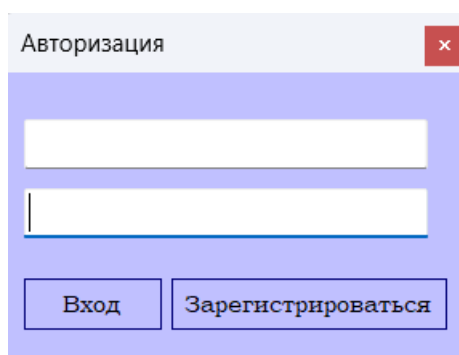
Таблица А.4

Отзывы

1	1	5	"Игра увлекательная, графика отличная"	4
2	2	10	"Местами скучная, но игра стоит того"	3
3	3	15	"Понравилась графика и геймплей, но есть баги"	5
4	4	3	"Игра неплохая, но хотелось бы больше контента"	3
5	5	8	"Затягивает, интересная механика"	4
6	6	12	"Отличная игра, люблю такой жанр"	5
7	7	7	"Нужно доработать управление, но в целом не плохо"	3
8	8	2	"Интересный сюжет, но немного однообразная"	4

Приложение Б

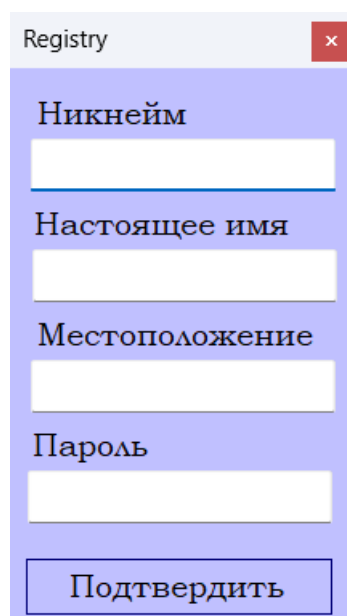
Скриншоты программы



Авторизация

Вход Зарегистрироваться

Рисунок Б.1 – Скриншот окна авторизации



Registry

Никнейм

Настоящее имя

Местоположение

Пароль

Подтвердить

Рисунок Б.2 – окно регистрации

test - подключен

Игра	Разработчик	Жанр	Цена
Red Dead Redemption 2	Rockstar Games	Action	6650
Assassin's Creed Valhalla	Ubisoft	Adventure	5700
Dark Souls III	FromSoftware	RPG	3800
Elden Ring	FromSoftware	RPG	5700
Cyberpunk 2077	CD Projekt Red	RPG	4750
The Witcher 3: Wild Hunt	CD Projekt Red	RPG	3600
Overwatch 2	Blizzard Entertainment	Shooter	3800
DOOM Eternal	Epic Games	Shooter	3800
Battlefield 2042	Electronic Arts	Shooter	5700
Civilization VI	Bandai Namco	Strategy	5700

Системные требования

ОС

Windows 8

ЦП

Intel Core i7-3770

ОЗУ

16

Видеокарта

NVIDIA GTX 1080 8GB

Память

70

Купить игру

Сортировка

☐ Отсортировать по цене

Показать только по:

Баланс: 754078

Пополнить баланс


моя библиотека

Рисунок Б.3 – рабочая область пользователя

Моя библиотека

Игра	Разработчик	Жанр
Cyberpunk 2077	CD Projekt Red	RPG
The Last of Us Part II	Naughty Dog	Horror
The Witcher 3: Wild Hunt	CD Projekt Red	RPG
League of Legends	Larian Studios	MMORPG
Fortnite	Epic Games	Battle Royale
Hollow Knight	IO Interactive	Metroidvania
Stardew Valley	Obsidian Entertainment	Sandbox
Minecraft	Larian Studios	Sandbox
Civilization VI	Bandai Namco	Strategy
Divinity: Original Sin 2	Obsidian Entertainment	RPG
Mario Kart 8 Deluxe	Nintendo	Racing
Battlefield 2042	Electronic Arts	Shooter
Assassin's Creed Valhalla	Ubisoft	Adventure
Dark Souls III	FromSoftware	RPG

Изменить аватар



Добавить отзыв

Редактировать отзыв

Удалить отзыв

отзыв

☐ Все отзывы

Отзыв	Рейтинг

Рисунок Б.4 – библиотека пользователя

Admin ✕

	id_developer	id_game	game	price	id_game_genre	id_requirements	state
▶	2	2	Assassin's Creed Valhalla	5700	2	8	Enabled
	4	3	Red Dead Redemption 2	6650	1	11	Enabled
	5	4	Cyberpunk 2077	4750	3	15	Enabled
	6	5	DOOM Eternal	3800	4	7	Enabled
	9	6	The Last of Us Part II	5700	12	6	Enabled
	11	7	Dark Souls III	3800	3	3	Enabled
	18	8	Hollow Knight	1500	18	10	Enabled
	17	9	Hitman 3	5700	15	14	Enabled
	19	10	Stardew Valley	1500	17	9	Enabled
	7	11	Overwatch 2	3800	4	2	Enabled
	20	12	League of Legends	0	14	1	Enabled
	11	13	Elden Ring	5700	3	16	Enabled
	16	14	Mario Kart 8 Deluxe	4750	10	5	Enabled
	15	15	Civilization VI	5700	7	13	Enabled
	14	16	Resident Evil Village	5700	11	18	Enabled

Добавить

Обновить

Удалить

t_games ▾

Рисунок Б.5 – рабочая область администратора

Приложение В

Код программы

<https://github.com/KariTimo/CourseWorkDataBase>

Изм.	Лист	№ докум.	Подпись	Дата

МИВУ 09.03.04-19.000ПЗ

Лист

32