Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
 высшего образования

**«Пермский национальный** **исследовательский   
политехнический университет»**

Факультет: Прикладной математики и механики

Кафедра: Вычислительной математики, механики и биомеханики

Направление магистратуры: 09.03.02 Информационные системы и технологии

Профиль: «Информационные системы и технологии»

**ОТЧЁТ**

по лабораторной работе  
по дисциплине  
**«Корпоративные информационные системы»**

Лабораторная работа № **3**

Тема работы: «**Проектирование интерфейса КИС**»

**Выполнил**: студент гр. ЦТУ-20-3б  
Галашов Данил Андреевич

**Проверил:** ст. пр. каф. ВММБ   
Банников Руслан Юрьевич

**Пермь 2024**

Оглавление

[Постановка задачи. 3](#_Toc164103742)

[Предметная область проектируемой БД. 4](#_Toc164103743)

[Mодель базы банных. 5](#_Toc164103744)

[Содержимое БД. 10](#_Toc164103745)

[Вывод. 21](#_Toc164103746)

# Постановка задачи.

Целю работы, является создание информационной системы «Агрегатор видео игр», предназначенной для, в первую очередь, для сравнения цен видео игр на разных онлайн торговых площадках.

Для выполнения поставленной задачи необходимо:

* Спроектировать и реализовать базу данных информационной системы «Агрегатор видео игр», удовлетворяющую поставленные требования технического задания.
* Спроектировать и реализовать интерфейс для работы с базой данных информационной системы «Агрегатор видео игр», удовлетворяющую поставленные требования технического задания.
* Провести тестирование информационной системы «Агрегатор видео игр»
* Написать техническую документацию для информационной системы «Агрегатор видео игр»

# Интерфейс.

В ходе проведенного исследования, предпочтение отдано языку программирования C#. Для реализации пользовательского интерфейса информационной системы он имеет несколько преимуществ перед аналогами, такими как Python и Java, в частности:

* Использование платформы .NET: C# разработан для работы на платформе .NET, которая предоставляет широкий набор инструментов и библиотек для разработки приложений.
* Библиотеки, такие как Windows Forms или Universal Windows Platform, обеспечивают удобные инструменты для создания современных и красивых интерфейсов.
* C# обладает возможностью легкой интеграции с базами данных и веб-службами, что позволяет облегчить взаимодействие частей приложения.

Для реализации функций агрегатора было разработано программное средство, включающее в себя следующие функции:

1) **Подключение к базе данных**. Для подключения к базе данных необходимо задать параметры подключения, а также задать функции для подключения и отключения.

MySqlConnection connection = new MySqlConnection("server=localhost;user=root;password=root;database=kyrsovaya\_v0;");

public void openConnection()

{

if (connection.State == System.Data.ConnectionState.Closed)

{

connection.Open();

}

}

public void closeConnection()

{

if (connection.State == System.Data.ConnectionState.Open)

{

connection.Close();

}

}

public MySqlConnection getConnection()

{

return connection;

}

2) **Авторизация.** Для авторизации необходимо логин и пароль, нажать кнопку «Авторизоваться» (Рис. 4). В случае нахождения в базе данных совпадений, в зависимости от роли пользователя он перейдет или на форму пользователя либо на форму администратора. Если совпадений найдено не было выводится сообщение, информирующее пользователя об этом.

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

{

String loginUser = textBoxUsername.Text;

String passwordUser = textBoxPassword.Text;

DataBase db = new DataBase();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM `user` WHERE `user\_name` = @username AND `user\_password` = @user\_password", db.getConnection());

command.Parameters.Add("@username", MySqlDbType.VarChar).Value = loginUser;

command.Parameters.Add("@user\_password", MySqlDbType.VarChar).Value = passwordUser;

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0)

{

DataRow[] rows = table.Select();

User user = new User(

Convert.ToString(rows[0]["user\_id"]),

Convert.ToString(rows[0]["user\_name"]),

Convert.ToString(rows[0]["user\_password"]),

Convert.ToString(rows[0]["user\_role"]));

this.Hide();

if (user.Role == "User")

{

Form1 form1 = new Form1(user);

form1.Show();

}

else if (user.Role == "Admin")

{

Form3 form3 = new Form3(user);

form3.Show();

}

}

else

{ MessageBox.Show("Не верный логин или пароль. Проверьте введеные данные."); }

}

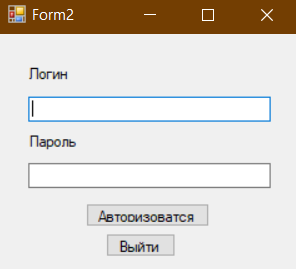


Рис 4. Форма Авторизации.

3) **Поиск**. Для выполнения поиска необходимо выбрать характеристику, по которой будет производиться поиск, ввести поисковой запрос и нажать кнопку «Найти». В случае нахождения в базе данных совпадений, игры должны выводиться по одной на страницу с возможностью переключения страниц (Рис. 5). Если совпадений найдено не было выводится сообщение, информирующее пользователя об этом.

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

{

string search = textBoxSearch.Text;

DataBase db = new DataBase();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command;

if (radioButtonTitel.Checked) {

command = new MySqlCommand("SELECT \* FROM `game\_summary` WHERE `titel` = @Titel", db.getConnection());

command.Parameters.Add("@Titel", MySqlDbType.VarChar).Value = search;

adapter.SelectCommand = command;

adapter.Fill(table);

if (!(table.Rows.Count > 0))

{

MessageBox.Show("По вашему запросу ничего не найдено");

FindedGames = new List<Game>();

UpdateGameInterfece();

}

else {

FromTableToListGame(table);

UpdateGameInterfece();

}

}

else if (radioButtonDeveloper.Checked) {

command = new MySqlCommand("SELECT \* FROM `game\_summary` WHERE `developer\_name` = @Developer;", db.getConnection());

command.Parameters.Add("@Developer", MySqlDbType.VarChar).Value = search;

adapter.SelectCommand = command;

adapter.Fill(table);

if (!(table.Rows.Count > 0))

{

MessageBox.Show("По вашему запросу ничего не найдено");

FindedGames = new List<Game>();

UpdateGameInterfece();

}

else {

FromTableToListGame(table);

UpdateGameInterfece();

}

}

else if (radioButtonPublisher.Checked)

{

command = new MySqlCommand("SELECT \* FROM `game\_summary` WHERE `publisher\_name` = @Publisher;", db.getConnection());

command.Parameters.Add("@Publisher", MySqlDbType.VarChar).Value = search;

adapter.SelectCommand = command;

adapter.Fill(table);

if (!(table.Rows.Count > 0))

{

MessageBox.Show("По вашему запросу ничего не найдено");

FindedGames = new List<Game>();

UpdateGameInterfece();

}

else {

FromTableToListGame(table);

UpdateGameInterfece();

}

}

else if (radioButtonWebStore.Checked) {

command = new MySqlCommand("SELECT \* FROM `game\_summary` WHERE `web\_store` = @Web\_Store;", db.getConnection());

command.Parameters.Add("@Web\_Store", MySqlDbType.VarChar).Value = search;

adapter.SelectCommand = command;

adapter.Fill(table);

if (!(table.Rows.Count > 0))

{

MessageBox.Show("По вашему запросу ничего не найдено");

FindedGames = new List<Game>();

UpdateGameInterfece();

}

else {

FromTableToListGame(table);

UpdateGameInterfece();

}

}

else if (radioButtonCost.Checked)

{

try { Convert.ToInt32(search); }

catch

{

MessageBox.Show("Цена должна быть числом");

return;

}

command = new MySqlCommand("SELECT \* FROM `game\_summary` WHERE `cost` < @Cost;", db.getConnection());

command.Parameters.Add("@Cost", MySqlDbType.Int32).Value = search;

adapter.SelectCommand = command;

adapter.Fill(table);

if (!(table.Rows.Count > 0))

{

MessageBox.Show("По вашему запросу ничего не найдено");

FindedGames = new List<Game>();

UpdateGameInterfece();

}

else

{

FromTableToListGame(table);

UpdateGameInterfece();

}

}

else {

MessageBox.Show("Параметры поcика не выбраны!");

}

}

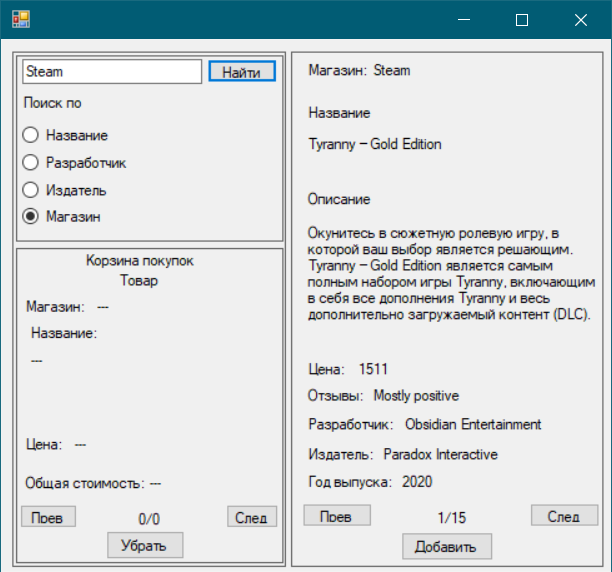


Рис. 5. Выполненный поиск по названию магазина

4) **Добавление товара в корзину**. Для добавления товара в корзину нужно выбрать игру и нажать на кнопку «Добавить», после этого должно появиться сообщение об успешном добавлении товара и в разделе «Корзина покупок» должна появиться выбранная игра (Рис. 6). В случае если игра не выбрана, выводится сообщение об ошибке.

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

{

int game\_id = Convert.ToInt32(FindedGames[GameMarker].Game\_Id);

DataBase db = new DataBase();

MySqlCommand command = new MySqlCommand("INSERT INTO `kyrsovaya\_v0`.`shopping\_cart` (`game\_game\_id`, `bill\_bill\_id`) VALUES (@Game\_Id, '1')", db.getConnection());

command.Parameters.Add("@Game\_Id", MySqlDbType.Int32).Value = game\_id;

db.openConnection();

if (command.ExecuteNonQuery() == 1)

{ MessageBox.Show("Игра успешно добавленна в корзину"); }

else

{ MessageBox.Show("Упс! При добавлении иры в корзину возникла ошибка"); }

db.closeConnection();

getShoppingCartSummary();

}

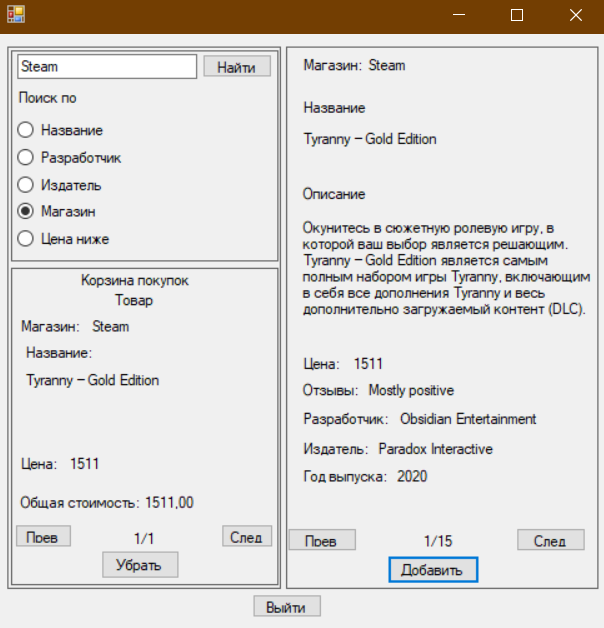


Рис. 6. Добавление товара в корзину

5) **Удаление товара из корзины**. Для удаления товара из корзины нужно выбрать игру в разделе «Корзина покупок» и нажать на кнопку «Убрать» (Рис. 7). После этого должно появиться сообщение об успешном удалении товара. В разделе «Корзина покупок» выбранная игра должна отсутствовать. В случае если игра не выбрана, выводится сообщение об ошибке.

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

{

int product\_id = Convert.ToInt32(ShoppingCart[ProductMarker].Product\_ID);

DataBase db = new DataBase();

MySqlCommand command = new MySqlCommand("DELETE FROM `kyrsovaya\_v0`.`shopping\_cart` WHERE `product\_id` = @Product\_ID", db.getConnection());

command.Parameters.Add("@Product\_ID", MySqlDbType.Int32).Value = product\_id;

db.openConnection();

if (command.ExecuteNonQuery() == 1)

{ MessageBox.Show("Игра успешно удалена в корзину"); }

else

{ MessageBox.Show("Упс! При удалении из корзину возникла ошибка"); }

db.closeConnection();

getShoppingCartSummary();

UpdateProductInterfece();

}

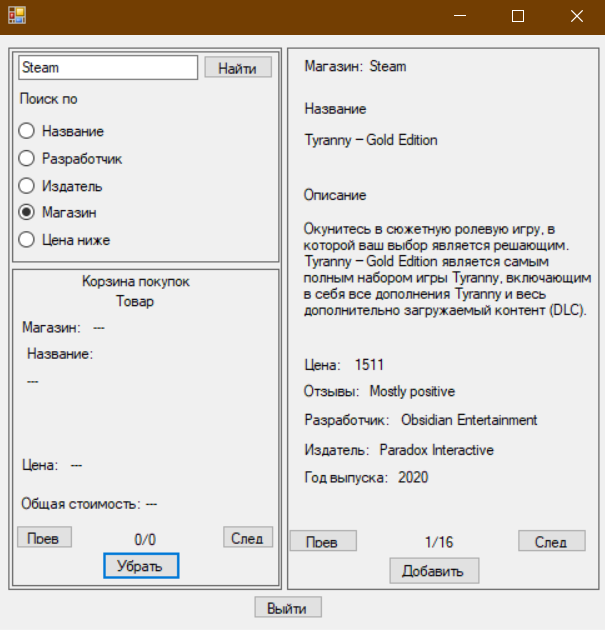


Рис. 7. Удаление игры из корзины.

6) **Добавление продукта**. Для добавления товара в корзину нужно заполнить все поля в разделе игру и нажать на кнопку «Добавить», после этого должно появиться сообщение об успешном добавлении товара и в разделе «Корзина покупок» должна появиться выбранная игра (Рис. 6). В случае если игра не выбрана, выводится сообщение об ошибке.

private List<Developer> getDevelopers()

{

List<Developer> developers = new List<Developer>();

DataBase db = new DataBase();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command;

command = new MySqlCommand("SELECT \* FROM `developer`", db.getConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

DataRow[] rows = table.Select();

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

{

Developer p = new Developer(Convert.ToInt32(rows[i]["developer\_id"]), Convert.ToString(rows[i]["developer\_name"]));

developers.Add(p);

}

return developers;

}

private List<Publisher> getPublishers()

{

List<Publisher> publishers = new List<Publisher>();

DataBase db = new DataBase();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command;

command = new MySqlCommand("SELECT \* FROM `publisher`", db.getConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

DataRow[] rows = table.Select();

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

{

Publisher p = new Publisher(Convert.ToInt32(rows[i]["publisher\_id"]), Convert.ToString(rows[i]["publisher\_name"]));

publishers.Add(p);

}

return publishers;

}

private void createDeveloper()

{

string developer\_name = textBoxDeveloperName.Text;

DataBase db = new DataBase();

MySqlCommand command = new MySqlCommand("INSERT INTO `kyrsovaya\_kis`.`developer` (`developer\_name`) VALUES (@Developer\_Name);", db.getConnection());

command.Parameters.Add("@Developer\_Name", MySqlDbType.String).Value = developer\_name;

db.openConnection();

if (command.ExecuteNonQuery() == 1)

{ MessageBox.Show("Разработчик успешно добавленн"); }

else

{ MessageBox.Show("Упс! При добавлении Разработчика возникла ошибка"); }

db.closeConnection();

}

private void createPublisher()

{

string publisher\_name = textBoxPublisherName.Text;

DataBase db = new DataBase();

MySqlCommand command = new MySqlCommand("INSERT INTO `kyrsovaya\_kis`.`publisher` (`publisher\_name`) VALUES (@Publisher\_Name);", db.getConnection());

command.Parameters.Add("@Publisher\_Name", MySqlDbType.String).Value = publisher\_name;

db.openConnection();

if (command.ExecuteNonQuery() == 1)

{ MessageBox.Show("Издатель успешно добавленн"); }

else

{ MessageBox.Show("Упс! При добавлении Издателя возникла ошибка"); }

db.closeConnection();

}

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

{

List<Developer> developers = getDevelopers();

List<Publisher> publishers = getPublishers();

Developer curent\_developer = null;

Publisher curent\_publisher = null;

int web\_store\_id=0;

string web\_store = textBoxWebStore.Text;

string titel = textBoxTitel.Text;

string description = textBoxDescription.Text;

string cost = textBoxCost.Text;

string reviews = comboBox1.Text;

string developer\_name = textBoxDeveloperName.Text;

string publisher\_name = textBoxPublisherName.Text;

string release\_year = textBoxReleaseYear.Text;

// Проверки полей.

if (web\_store == "")

{

MessageBox.Show("Не введено название магазаина");

return;

}

else if (web\_store == "Steam") { web\_store\_id = 1; }

else if (web\_store == "Epic Games") { web\_store\_id = 2; }

else

{

MessageBox.Show("Такого магазина не существует, проверьде корректность введенных данных.") ;

return;

}

if (titel == "")

{

MessageBox.Show("Не введено название игры");

return;

}

if(description == "")

{

MessageBox.Show("Не введено название описание игры");

return;

}

if(cost == "")

{

MessageBox.Show("Не введена стоимость игры");

return;

}

if(reviews == "")

{

MessageBox.Show("Не введены отзывы");

return;

}

if (developer\_name == "")

{

MessageBox.Show("Не введено название разработчика");

return;

}

if(publisher\_name == "")

{

MessageBox.Show("Не введено название издателя");

return;

}

if(release\_year == "")

{

MessageBox.Show("Не введен год выхода игры");

return;

}

try

{ int new\_cost = Convert.ToInt32(cost); }

catch

{

MessageBox.Show("Цена должна быть числом");

return;

}

try

{ int new\_release\_year = Convert.ToInt32(release\_year);

}

catch

{

MessageBox.Show("Год выхода должен быть числом");

return;

}

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

{

if (developers[i].Developer\_Name == developer\_name)

{

curent\_developer = developers[i];

}

}

if (curent\_developer == null)

{

createDeveloper();

developers = getDevelopers();

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

{

if (developers[i].Developer\_Name == developer\_name)

{

curent\_developer = developers[i];

}

}

}

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

{

if (publishers[i].Publisher\_Name == publisher\_name)

{

curent\_publisher = publishers[i];

}

}

if (curent\_publisher == null)

{

createPublisher();

publishers = getPublishers();

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

{

if (publishers[i].Publisher\_Name == publisher\_name)

{

curent\_publisher = publishers[i];

}

}

}

DataBase db = new DataBase();

MySqlCommand command = new MySqlCommand("INSERT INTO `kyrsovaya\_kis`.`game` (`titel`, `description`, `release\_year`, `cost`, `reviews`, `inventory\_inventory\_id`, `publisher\_publisher\_id`, `developer\_developer\_id`) VALUES (@Titel, @Description, @Release\_Year, @Cost, @Reviews, @Inventory\_Inventory\_Id, @Publisher\_Publisher\_Id, @Developer\_Developer\_Id);", db.getConnection());

command.Parameters.Add("@Titel", MySqlDbType.VarChar).Value = titel;

command.Parameters.Add("@Description", MySqlDbType.Text).Value = description;

command.Parameters.Add("@Release\_Year", MySqlDbType.Year).Value = release\_year;

command.Parameters.Add("@Cost", MySqlDbType.Int32).Value = cost;

command.Parameters.Add("@Reviews", MySqlDbType.Enum).Value = reviews;

command.Parameters.Add("@Inventory\_Inventory\_Id", MySqlDbType.Int32).Value = web\_store\_id;

command.Parameters.Add("@Publisher\_Publisher\_Id", MySqlDbType.Int32).Value = curent\_publisher.Publisher\_Id;

command.Parameters.Add("@Developer\_Developer\_Id", MySqlDbType.Int32).Value = curent\_developer.Developer\_Id;

db.openConnection();

if (command.ExecuteNonQuery() == 1)

{ MessageBox.Show("Продукт успешно добавленн"); }

else

{ MessageBox.Show("Упс! При добавлении Продукта возникла ошибка"); }

db.closeConnection();

}

7) **Удаление продукта**. Для удаления товара из корзины нужно выбрать игру в разделе «Результаты поиска» и нажать на кнопку «Удалить». После этого должно появиться сообщение об успешном удалении товара. В разделе «Результаты поиска» выбранная игра должна отсутствовать. В случае если игра не выбрана, выводится сообщение об ошибке.

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

{

int game\_id = Convert.ToInt32(FindedGames[GameMarker].Game\_Id);

DataBase db = new DataBase();

MySqlCommand command = new MySqlCommand("DELETE FROM `kyrsovaya\_kis`.`game` WHERE `game\_id` = @Game\_ID", db.getConnection());

command.Parameters.Add("@Game\_ID", MySqlDbType.Int32).Value = game\_id;

db.openConnection();

if (command.ExecuteNonQuery() == 1)

{ MessageBox.Show("Игра успешно удалена "); }

else

{ MessageBox.Show("Упс! При удалении возникла ошибка"); }

db.closeConnection();

FindButton\_Click(sender, e);

}

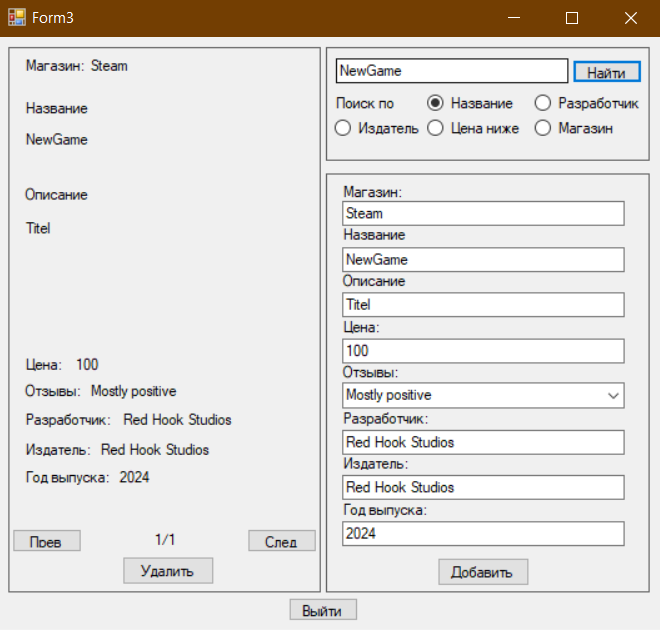


Рис. 8. Форма администратора.

# Вывод.

В результате проделанной работы был создан интерфейс для базы данных информационной системы «Агрегатор видео игр». Из-за сложности реализации запись отчетов в отдельный файл будет реализована в последующих версиях.

Следующим шагом будет тестирование работы базы данных информационной системы «Агрегатор видео игр».