

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

|  |
| --- |
| **КУРСОВОЙ ПРОЕКТ** |
|  |
| по междисциплинарному курсу: МДК.02.02. Технология разработки и защиты баз данных. |
|  |
| на тему:Разработка инфромационной системы для спортивного болельщика. |
|  |
| Студента группы 090203-9о-19/1  специальности 09.02.03 Программирование в компьютерных системах |
| Самохина Александра Юрьевича |

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

|  |  |  |
| --- | --- | --- |
| Дата защиты «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2022 г. | | |
| Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Заведующий отделением № 1 |  | И.А. Миланова |

Москва

2022

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 2](#_Toc103637393)

[1 ОБЩАЯ ЧАСТЬ 4](#_Toc103637394)

[1.1 Назначение и цели создания системы 4](#_Toc103637395)

[1.2 Обзор и анализ предметной области 4](#_Toc103637396)

[1.3 Жизненный цикл базы данных 5](#_Toc103637397)

[1.4 Выбор и характеристика СУБД 8](#_Toc103637398)

[1.5 Выбор и характеристика среды разработки приложения 8](#_Toc103637399)

[2 СПЕЦИАЛЬНАЯ ЧАСТЬ 10](#_Toc103637400)

[2.1 Постановка задачи 10](#_Toc103637401)

[2.2 Архитектура информационной системы 10](#_Toc103637402)

[2.3 Нормализация таблиц 11](#_Toc103637403)

[2.4 Описание таблиц 14](#_Toc103637404)

[2.5 Разработка приложения 17](#_Toc103637405)

[2.6 Инструкция пользователю 23](#_Toc103637406)

[ЗАКЛЮЧЕНИЕ 26](#_Toc103637407)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 27](#_Toc103637408)

[ПРИЛОЖЕНИЕ А. SQL скрипты на создание и заполнение базы данных 28](#_Toc103637409)

[ПРИЛОЖЕНИЕ Б. КОД ПРОГРАММЫ 50](#_Toc103637410)

[Код окна «регистрация» 50](#_Toc103637411)

[Код окна «бомбардиры» 52](#_Toc103637412)

[Код окна «игроки» 56](#_Toc103637413)

# ВВЕДЕНИЕ

Стремительный рост популярности компьютеров был зафиксирован в конце XX - в начале XXI в. Процесс внедрения компьютеров охватил большую часть сфер человеческой жизнедеятельности .

Довольно непросто в наши дни отыскать отрасль, где бы не была затронута компьютерная деятельность. Использование этих устройств сегодня, упрощает жизнь человеку, выполняя большую часть энергозатратной работы за него и позволяет сконцентрироваться на действительно важной проблеме. Одни из основных функций компьютеров - моделирование процессов, хранение, поиск, преобразование и использование информации в цифровой форме, они позволяют нам находить наиболее оптимальные решения.

Цель всей компьютеризации состоит в улучшении качества жизни людей за счет увеличения производительности и облегчения условий их труда.

Применение компьютера уже обыденность почти для каждого человека и редко является показателем материального благосостояния.

Курсовой проект посвящен разработке информационной системы для спортивного болельщика.

Основная цель проекта – создание приложения, которое позволяет болельщику следить за футбольным турниром, администраторам следить за актуальностью статистики и данных турнира. Также данное приложение должно поддерживать подключение базы данных (SQL).

# ОБЩАЯ ЧАСТЬ

## Назначение и цели создания системы

### Объект исследования

Информационная система для спортивного болельщика.

### Предмет исследования

Настольное приложение для спортивного болельщика, взаимодействующее с базой данных SQL.

### Основная цель проекта

Разработка информационной системы для спортивного болельщика, которая позволяет пользователю просматривать актуальные данные футбольного турнира. Также приложение взаимодействующее с базой данных: MS SQL SERVER 2019.

## Обзор и анализ предметной области

Приложение создаётся для информационного обслуживания пользователей, а также для администрирования информационного ресурса. В БД должна храниться следующая информация: даты, место проведения, результаты матчей, страны участников, спортивные клубы, составы команд, фамилии спортсменов, забивших голы, фамилии тренеров и т.д.

В соответствии с предметной областью система строится с учётом следующих особенностей:

1. один пользователь может запросить несколько отчётов о результатах матчей.
2. администраторы управляют приложением с возможностью добавить/удалить/редактировать всю информацию по турниру.

Выделим базовые сущности этой предметной области:

**Клубы**. Атрибуты клубов – код клуба, название клуба, город, страна, код тренера.

**Игроки**. Атрибуты игроков – код игрока, фамилия, имя, позиция, номер игрока в клубе, код клуба.

**Дата и место.** Атрибуты даты и места – код матча, дата и время, название турнира, код стадиона.

**Голы.** Атрибуты голов – код игрока, код матча, код клуба, кол-во голов.

## Жизненный цикл базы данных

Жизненный цикл базы данных включает следующие этапы:

* Анализ предметной области,
* Проектирование,
* Реализация,
* Тестирование,
* Эксплуатация и сопровождение.

### Анализ предметной области

Для разработки данного проекта требуется Microsoft SQL Server где будет храниться вся информация. Для написания программы требуется Visual Studio 2019. Для построения логической модели базы данных используется ER Assistant.

Разработчик располагает всеми необходимыми средствами для разработки баз данных: СУБД (MS SQL SERVER 2019, ER Asistant, Visual Studio 2019).

Необходимо создать настольное приложение для спортивного болельщика, взаимодействующее с базой данных SQL.

Требования к проекту:

* База данных должна быть расширяемой для добавления новых пользователей, администраторов, клубов, игроков, и т.д.,
* Авторизированный пользователь должен иметь возможность просматривать всю нужную статистику футбольного турнира.

### Проектирование

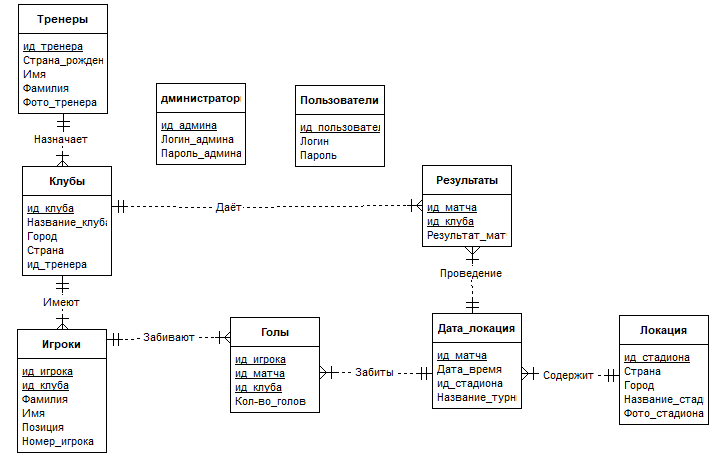


Рисунок 1 – Логическая модель базы данных

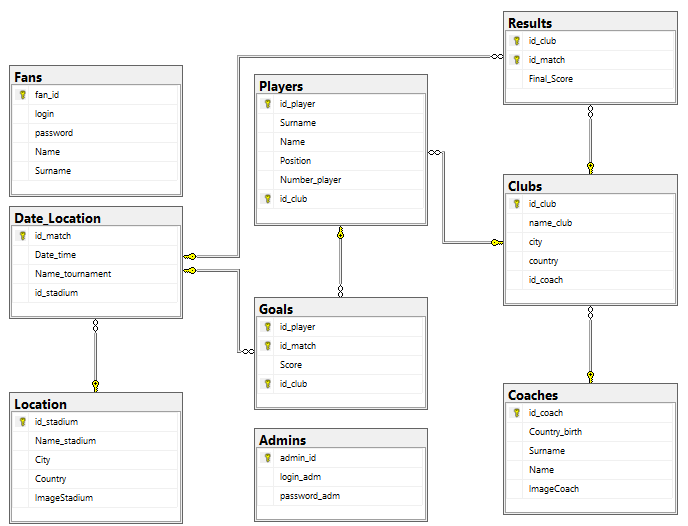


Рисунок 2 – Физическая модель базы данных

### Реализация

Разработка программы позволяющей работать с БД производилась в Visual Studio 2019 на языке C#, на платформе Windows Presentation Foundation (WPF).

### Тестирование

Тестирование и отладка выполнялись по ходу реализации программы. Все ошибки выявленные при запуске или работе с приложением – устранены. Тестирование прошло успешно.

### Эксплуатация и сопровождение

Продукт готов к использованию и дальнейшему сопровождению.

## Выбор и характеристика СУБД

Для хранения данных была выбрана реляционная СУБД «MS SQL SERVER 2019».

MS SQL SERVER 2019 был выбран из-за того, что – эта платформа обладает высокой доступностью, повышенной производительностью и безопасностью. Решение представляет собой хорошо масштабируемый, полностью реляционный, быстродействующий сервер, способный обрабатывать большие объемы данных для клиент-серверных приложений. Рекордная производительность MS SQL SERVER 2019 обеспечивается новыми технологиями работы с памятью.

Преимущество платформы:

* Рекордно высокая производительность,
* Быстрое получение результатов анализа локально,
* База данных менее уязвима.

Из недостатков можно выделить зависимость от операционной среды: СУБД работает только с системой Windows.

## Выбор и характеристика среды разработки приложения

Для разработки приложения была выбрана интегрированная среда разработки «Visual Studio 2019», включающая [интегрированную среду разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) программного обеспечения и ряд других инструментальных средств.

C# — популярный язык программирования. Он является основным инструментом для разработки программного обеспечения компании Microsoft. Этот объектно-ориентированный язык применяется для создания веб-приложений и десктопных программ.

Visual Studio признана лучшей IDE для C#. Дело в том, что оба продукта принадлежат корпорации Microsoft. Поэтому они идеально подходят для работы друг с другом.

Помимо того, что Visual Studio 2019 позволяет создавать [консольные](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) [приложения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), оно позволяет разрабатывать приложения с [графическим интерфейсом](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F), в том числе с поддержкой технологии [Windows Forms](https://ru.wikipedia.org/wiki/Windows_Forms), а также [веб-сайты](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82), [веб-приложения](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), [веб-службы](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0) как в [родном](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4), так и в [управляемом](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D0%BC%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) коде для всех платформ, поддерживаемых [Windows](https://ru.wikipedia.org/wiki/Windows), [Windows Mobile](https://ru.wikipedia.org/wiki/Windows_Mobile), и т.д.

В число преимуществ Visual Studio входит следующее:

* Среда содержит множество инструментов, которые очень хорошо работают на C#,
* Наличие бесплатной версии — Community Edition,
* Community содержит все что нужно для независимого разработчика,
* Самое эффективное ПО для разработки на любой платформе, включая .Net и C#,
* Возможность хранения данных в облаке.

# **СПЕЦИАЛЬНАЯ ЧАСТЬ**

## Постановка задачи

Спроектировать средствами MS SQL SERVER 2019 базу данных и создать средствами C# приложение, позволяющее просматривать турнирные таблицы чемпионата/кубка. В базе данных должна храниться следующая информация: даты, место проведения, результаты матчей, страны участников, спортивные клубы, составы команд, фамилии спортсменов, забивших голы, фамилии тренеров и т.д. Предусмотреть возможность добавления, удаления, редактирования записей.

Обеспечить:

* поиск  матчей заданной  команды,
* поиск  матчей по месту проведения,
* поиск фамилий спортсменов, забивших голы в выбранном матче,
* просмотр списка игроков заданной  команды,
* возможность нахождения самых результативных игроков,
* возможность нахождения самых результативных команд.

Предусмотреть выдачу отчётов. Обеспечить защиту базы данных и приложения.

## Архитектура информационной системы

База данных

Сервер

Клиент

Рисунок 3 – Клиент-серверная модель

Архитектура информационной системы представляет собой клиент-серверную модель с базой данных. На сервере хранится вся информация, в свою очередь пользователь приложения может взаимодействовать с сервером, где расположена база данных.Сначала клиент должен пройти авторизацию или же регистрацию. Далее клиент может просматривать статистику, смотреть результаты матчей, сформировать отчёт матча, также можно смотреть информацию о стадионах, игроках и т.д. Сервер в данном случае - это компьютер пользователя, клиент сначла обращается к нему, только после этого сервер уже обращается к базе данных и получает все необходимые данные, которые нужны клиенту.

Преимущества:

* Распределение функций вычислительной системы между несколькими независимыми компьютерами в сети,
* Все данные хранятся на сервере, который защищён лучше большинства клиентов,
* Высокая безопасность,
* Высокая надёжность.

Недостатки:

* Неработоспособность сервера может сделать неработоспособной всю вычислительную сеть,
* Поддержка работы данной системы требует отдельного специалиста — системного администратора,
* Высокая стоимость оборудования.

## Нормализация таблиц

Таблицы были приведены к первой нормальной форме, все атрибуты просты, далее неделимы.

Таблицы приведены ко второй нормальной форме так как каждый не ключевой атрибут функционально полно зависит от ключа.

Таблицы были приведены к третьей нормальной форме, отсутствует транзитивная зависимость.

Код тренера \*

Страна рождения

Фамилия

Имя

Фото тренера

Рисунок 4 – «Тренеры»

Код клуба \*

Название клуба

Город

Страна

Код тренера

Рисунок 5 – «Клубы»

Код игрока \*

Имя

Фамилия

Позиция

Номер в клубе

Код клуба

Рисунок 6 – «Игроки»

Код стадиона \*

Фото

Название стадиона

Страна

Город

Рисунок 7 – «Место»

Код матча \*

Дата и время

Код стадиона

Название турнира

Рисунок 8 – «Дата и место»

Код игрока \*

Код матча \*

Код клуба \*

Количество голов

Рисунок 9 – «Голы»

Код матча \*

Код клуба \*

Финальный счёт

Рисунок 10 – «Результаты»

Код администратора \*

Логин

Пароль

Рисунок 11 – «Администраторы»

Код пользователя \*

Логин

Пароль

Фамилия

Имя

Рисунок 12 – «Пользователи»

## Описание таблиц

Таблица 1 - Coaches (Тренеры)

|  |  |  |  |
| --- | --- | --- | --- |
| Название  столбца | Описание | Тип | Примечание |
| id\_coach | Код тренера | INT | PK, NOT NULL |
| Country\_birth | Национальность | varchar(50) | NULL |
| Surname | Фамилия тренера | varchar(50) | NOT NULL |
| Name | Имя тренера | varchar(50) | NULL |
| ImageCoach | Фото тренера | varbinary(MAX) | NULL |

Таблица 2 – Clubs (Клубы)

|  |  |  |  |
| --- | --- | --- | --- |
| Название  столбца | Описание | Тип | Примечание |
| id\_club | Код клуба | INT | PK, NOT NULL |
| name\_club | Название клуба | varchar(50) | NOT NULL |
| city | Город клуба | varchar(50) | NULL |
| country | Страна клуба | varchar(50) | NULL |
| id\_coach | Код тренера | INT | NOT NULL |

Таблица 3 - Players (Игроки)

|  |  |  |  |
| --- | --- | --- | --- |
| Название  столбца | Описание | Тип | Примечание |
| id\_player | Код игрока | INT | PK, NOT NULL |
| Surname | Фамилия игрока | varchar(50) | NOT NULL |
| Name | Имя игрока | varchar(50) | NULL |
| Position | Позиция | varchar(50) | NOT NULL |
| Number\_player | Номер игрока в клубе | INT | NULL |
| id\_club | Код клуба | INT | PK, NOT NULL |

Таблица 4 - Location (Место)

|  |  |  |  |
| --- | --- | --- | --- |
| Название  столбца | Описание | Тип | Примечание |
| id\_stadium | Код стадиона | INT | PK, NOT NULL |
| Name\_stadium | Название стадиона | varchar(50) | NOT NULL |
| City | Город | varchar(50) | NULL |
| Country | Страна | varchar(50) | NULL |
| ImageStadium | Фотография стадиона | varbinary(MAX) | NULL |

Таблица 5 – Date\_Location (Дата и место)

|  |  |  |  |
| --- | --- | --- | --- |
| Название  столбца | Описание | Тип | Примечание |
| id\_match | Код матча | INT | PK, NOT NULL |
| Date\_time | Дата и время | smalldatetime | NOT NULL |
| Name\_tournament | Название турнира | varchar(50) | NOT NULL |
| id\_stadium | Код стадиона | INT | NOT NULL |

Таблица 6 - Goals (Голы)

|  |  |  |  |
| --- | --- | --- | --- |
| Название  столбца | Описание | Тип | Примечание |
| id\_player | Код игрока | INT | PK, NOT NULL |
| id\_match | Код матча | INT | PK, NOT NULL |
| Score | Кол-во мячей | INT | NOT NULL |
| id\_club | Код клуба | INT | PK, NOT NULL |

Таблица 7 – Admins (Администраторы)

|  |  |  |  |
| --- | --- | --- | --- |
| Название  Столбца | Описание | Тип | Примечание |
| admin\_id | Код админа | INT | PK, NOT NULL |
| login\_adm | Логин админа | varchar(50) | NOT NULL |
| password\_adm | Пароль админа | varchar(50) | NOT NULL |

Таблица 8 – Fans (Пользователи)

|  |  |  |  |
| --- | --- | --- | --- |
| Название  Столбца | Описание | Тип | Примечание |
| fan\_id | Тип тура | INT | PK, NOT NULL |
| login | Тип питания | varchar(50) | NOT NULL |
| password | Тип питания | varchar(50) | NOT NULL |
| Name | Тип питания | varchar(50) | NOT NULL |
| Surname | Тип питания | varchar(50) | NOT NULL |

## Разработка приложения

### Диаграмма вариантов использования

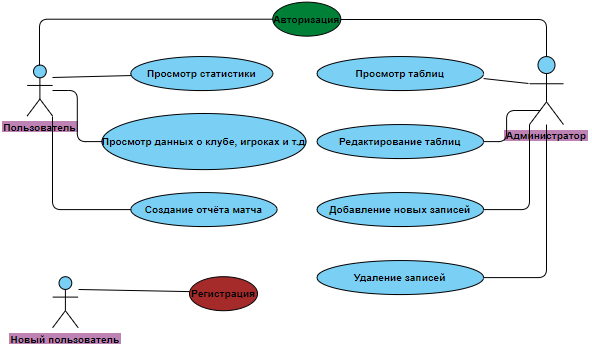


Рисунок 13 – Диаграмма вариантов использования

### Решение главной задачи проекта

Главной задачей проекта является создание информационной системы для спортивного болельщика, позволяющая просматривать результаты игр и статистику, а также формировать отчёт матчей.

В данном фрагменте кода осуществляется формирование отчёта матча (смотреть рисунок – 14, 15).

namespace FootballAppListView

{

/// <summary>

/// Логика взаимодействия для Results\_Window.xaml

/// </summary>

public partial class Results\_Window : Window

{

public Results\_Window()

{

InitializeComponent();

DGridResults.ItemsSource = FootballEntities.GetContext().Match\_Location().ToList();

}

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

{

StartWindow win1 = new StartWindow();

win1.Show();

this.Close();

}

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

{

Main\_Window win2 = new Main\_Window();

win2.Show();

this.Close();

}

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

{

var SelectOtchet = DGridResults.SelectedItems.Cast<Match\_Location\_Result>().FirstOrDefault();

Otchet otchet = new Otchet(SelectOtchet);

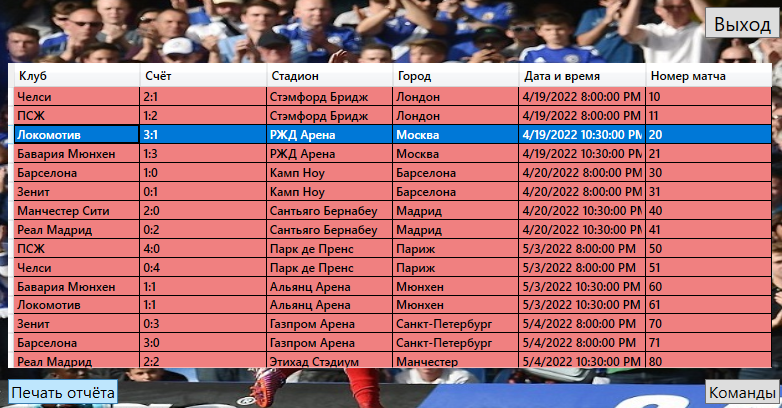
otchet.Show();

}

}

}

}



Риунок 14 – скриншот «выбора матча для печати отчёта»

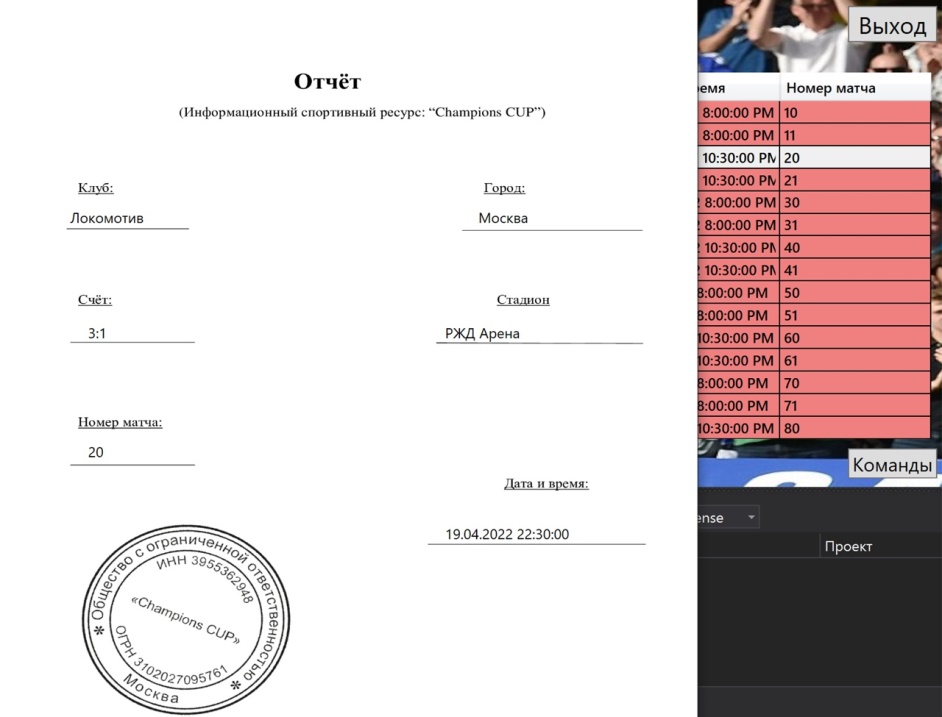


Рисунок 15 – скриншот отчёта

Осуществление хранения данных с поиском по фамилии/клубу во фрагменте кода (смотреть рисунок – 16, 17):

namespace FootballAppListView

{

/// <summary>

/// </summary>

public partial class Players\_Window : Window

{

string \_name;

public Players\_Window()

{

InitializeComponent();

DGridPlayers.ItemsSource = FootballEntities.GetContext().Players.ToList();

}

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

{

Main\_Window2 win6 = new Main\_Window2();

win6.Show();

this.Close();

}

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

{

Clubs Name = null;

\_name = FindPlayers.Text;

Name = FootballEntities.GetContext().Clubs.Where(b => b.name\_club == \_name).FirstOrDefault();

if (Name == null)

{

DGridPlayers.ItemsSource = FootballEntities.GetContext().Players.ToList();

}

else

{

DGridPlayers.ItemsSource = FootballEntities.GetContext().Players.Where(b => b.Clubs.name\_club == \_name).ToList();

}

}

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

{

Players Name = null;

\_name = FindPlayersName.Text;

Name = FootballEntities.GetContext().Players.Where(b => b.Surname == \_name).FirstOrDefault();

if (Name == null)

{

DGridPlayers.ItemsSource = FootballEntities.GetContext().Players.ToList();

}

else

{

DGridPlayers.ItemsSource = FootballEntities.GetContext().Players.Where(b => b.Surname == \_name).ToList();

}

}

}

}

Скриншоты результатов:

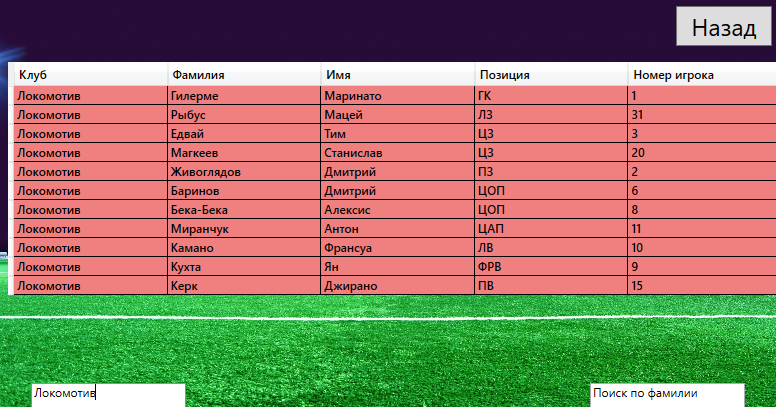


Рисунок 16 – Скриншот «поиска игроков по клубной принадлежности»



Рисунок 17 – Скриншот «поиска игрока по фамилии»

### Тестирование приложения

Таблица 10 – Метод проверки требований к приложению.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действие | Ожидаемый результат | Результат |
| 1 | Запуск приложения | Приложение подключается к серверу и базам данных | Успешно |
| 2 | Регистрация нового пользователя | Создается новая учетная запись в базе данных | Успешно |
| 3 | Авторизация пользователя | Зарегистрированный пользователь переходит на окно пользователя | Успешно |
| 4 | Авторизация администратора | Зарегистрированный администратор переходит на окно администратора | Успешно |
| 5 | Редактирование данных администратором | При выполнении одного из действий (Создания, удаления, изменения) происходит создание записи, удаление записи, и редактирование записи соответственно | Успешно |
| 6 | Выход | Переход на окно авторизации | Успешно |
| 7 | Формирование отчёта | Отчёт формируется при выборе матча и последующего нажатия на кнопку | Успешно |
| 8 | Поиск лучших бомбардиров турнира | Список бомбардиров предлагает на выбор «топ-5» и «все бомбардиры», после выбора откроется таблица игроков с количеством голов | Успешно |
| 9 | Поиск игроков забивших в выбранном матче | При вводе кода матча (чётного), будут показаны игроки забившие в этом матче | Успешно |
| 10 | Поиск игрока по клубу или фамилии | При вводе названия клуба/фамилии в нужные текстовые поля, будут показаны игроки с заданной клубной принадлежностью/фамилией | Успешно |

### Защита информационной системы

Информационная система защищена от большинства ошибочных действий пользователя, в том числе и администратора. При вводе некорректных данных в большинстве случаев выходит соответствующее предупреждающее сообщение об ошибке.

Также идет проверка подключения к базе данных, после чего идет дальнейшее взаимодействие пользователя с программой, например, проверка логина и пароля пользователя, для проверки зарегистрирован такой пользователь или нет.

## Инструкция пользователю

Для пользования СУБД «Champions CUP», необходимо из архива перенести папки coach и stadium на диск C (для вывода изображений тренеров и стадионов), запустить скрипт на создание SQL-базы данных. Чтобы начать пользоваться нужно пройти авторизацию или зарегистрироваться.

По умолчанию есть два пользователя

1. Клиент:

* Логин: samokhin28,
* Пароль: 1511.

1. Администратор:

* Логин: komar1511,
* Пароль: 1511.

Для авторизации пользователь должен быть зарегистрирован. Для новых пользователей доступна функция регистрации. Если в результате регистрации были некорректно введены данные, программа выдаст соответствующие сведения об ошибке.

Пройдя авторизацию за клиента, произойдет переход на страницу, в которой пользователь может посмотреть тренеров команд, стадионы, клубы и их статистику, ознакомиться с игроками турнира и количеством их голов, перейти на страницу результаты, изучить и сформировать отчёт.

Пройдя авторизацию за администратора, произойдет переход на страницу, где откроется панель с выбором нужной таблицы для добавления/редактирования/удаления записей

### Общие сведения об информационной системе

База данных

Сервер

Клиент

Рисунок 18 – Клиент-серверная модель

Вся информационная система завязана на клиент-серверной модели, где идет взаимодействие пользовательского приложения через сервер к базам данных. В качестве баз данных выступает MS SQL SERVER 2019.

В качестве клиента выступает приложение «Champions CUP». Сервер расположен на компьютере пользователя.

### Требования к техническим средствам

В состав технических средств должен входить компьютер, включающий:

* оперативную память 2Гб,
* процессор Intel начиная с Pentium II и старше или процессор AMD схожий по характеристикам,
* клавиатура и мышь (стационарный компьютер) или тачпад,
* видеокарта или встроеный графический процессор,
* монитор,
* SSD или HDD диск.

### Требования к программным средствам

В состав программных средств должно входить:

* операционная система Windows 10 или версия выше,
* MS SQL SERVER 2019 .

### Формы ввода

Все формы ввода проверяют корректность введенных данных, которые вводит пользователь. Если данные не соответствуют условиям, то будет выводиться соответствующая ошибка.

### Отчёты

Отчёт, который сделал пользователь, отображается сразу на экране и готов к печати.

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта была разработана база данных для спортивного болельщика, а также приложение для просмотра статистики, информации об играх и формирования отчёта о них.

Программа отвечает всем требованиям предметной области, таблицы соответствуют правилам нормализации, что позволяет обеспечить целостность информации.

Средствами C# было создано приложение, которое решает все поставленные задачи проекта. Из этого можно сделать вывод, что курсовой проект был выполнен.

В процессе выполнения курсового проекта был приобретен практический опыт создания сложных систем. Был пройден жизненный цикл программного обеспечения:

* анализ предметной области,
* проектирование,
* реализация,
* тестирование,
* эксплуатация и сопровождение.

Приложением могут пользоваться два типа пользователей ‑ администратор и клиент. Клиенту даны такие возможности как просмотр полной информации о турнире(клубы,игроки,стадионы,результаты матчей), формирование отчёта о матче. Администратор может осуществлять добавление, удаление, редактирование записей в таблицах.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Общие сведения о переходах WPF – Url: https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/app-development/navigation-overview?view=netframeworkdesktop-4.8 (дата обращения: 20.04.2022).
2. Руководство по MS SQL SERVER 2019 2019 – Url: https://metanit.com/sql/sqlserver/ (дата обращения: 22.04.2022).
3. Информация о жизненном цикле базы данных – Url: <https://drive.google.com/file/d/1NRxBLext6kYa_TDt43PImjRHidBpK2_K/view?usp=sharing> (дата обращения: 22.04.2022).
4. Основная SQL информация – Url: <https://intuit.ru/studies/courses/5/5/info> (дата обращения: 24.04.2022).
5. Базы данных и системы управления базами данных – Url: <http://znanium.com/bookread2.php?book=946561> (дата обращения: 24.04.2022).
6. Информация о DataGrid – Url: <https://metanit.com/sharp/wpf/5.14.php> (дата обращения: 29.04.2022).
7. Информация о DataGrid – Url: <https://docs.microsoft.com/ru/dotnet/desktop/wpf/controls/datagrid?view=netframeworkdesktop-4.8> (дата обращения: 29.04.2022).
8. Информация о ListView – Url: https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.forms.listview?view=netframework-4.7.2 (дата обращения: 30.04.2022).
9. Информация о среде разработки – Url: https://spb.deveducation.com/blog/luchshie-ide-dlya-c-razrabotchika/ (дата обращения: 10.05.2022).

# ПРИЛОЖЕНИЕ А. SQL скрипты на создание и заполнение базы данных

USE master

IF EXISTS (SELECT \* FROM SYS.DATABASES WHERE NAME='Football')

DROP DATABASE Football

GO

CREATE DATABASE Football

ON PRIMARY

( NAME = Football,

FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Football.mdf', SIZE=3 MB, MAXSIZE=100, FILEGROWTH=10)

LOG ON

(NAME = Football\_log,

FILENAME ='C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Football\_log.ldf', SIZE=3MB, MAXSIZE=100, FILEGROWTH=10)

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME='Coaches' AND TYPE='U')

DROP TABLE Coaches

GO

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME='Clubs' AND TYPE='U')

DROP TABLE Clubs

GO

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME='Players' AND TYPE='U')

DROP TABLE Players

GO

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME='Location' AND TYPE='U')

DROP TABLE Location

GO

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME='Date\_Location' AND TYPE='U')

DROP TABLE Date\_Location

GO

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME='Goals' AND TYPE='U')

DROP TABLE Goals

GO

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME='Results' AND TYPE='U')

DROP TABLE Results

GO

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME='Fans' AND TYPE='U')

DROP TABLE Fans

GO

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME='Admins' AND TYPE='U')

DROP TABLE Fans

GO

USE Football

GO

CREATE TABLE Coaches(

id\_coach int PRIMARY KEY,

Country\_birth varchar(50) NULL,

Surname varchar(50) NOT NULL,

Name varchar(50) NULL,

ImageCoach varbinary(MAX) NULL

)

CREATE TABLE Clubs(

id\_club int,

name\_club varchar(50) NOT NULL,

city varchar(50) NULL,

country varchar(50) NULL,

id\_coach int NOT NULL,

PRIMARY KEY (id\_club),

CONSTRAINT fk\_coach FOREIGN KEY (id\_coach)

REFERENCES Coaches (id\_coach)

)

CREATE TABLE Players(

id\_player int,

Surname varchar(50) NOT NULL,

Name varchar(50) NULL,

Position varchar(50) NOT NULL,

Number\_player int NULL,

id\_club int NOT NULL,

PRIMARY KEY(id\_club, id\_player),

CONSTRAINT fk\_id\_clubs FOREIGN KEY (id\_club)

REFERENCES Clubs(id\_club)

)

CREATE TABLE Location(

id\_stadium int PRIMARY KEY,

Name\_stadium varchar(50) NOT NULL,

City varchar(50) NULL,

Country varchar(50) NULL,

ImageStadium varbinary(MAX) NULL

)

CREATE TABLE Date\_Location(

id\_match int PRIMARY KEY,

Date\_time smalldatetime NOT NULL,

Name\_tournament varchar(50) NOT NULL,

id\_stadium int NOT NULL,

CONSTRAINT fk\_id\_stadium FOREIGN KEY (id\_stadium)

REFERENCES Location(id\_stadium)

)

CREATE TABLE Results(

id\_club int NOT NULL,

id\_match int NOT NULL,

Final\_Score varchar(5) NOT NULL CHECK(Final\_Score LIKE '[0-9]:[0-9]'),

PRIMARY KEY(id\_club, id\_match),

CONSTRAINT fk\_id\_clubs\_results FOREIGN KEY (id\_club)

REFERENCES Clubs(id\_club),

CONSTRAINT fk\_id\_matches\_results FOREIGN KEY (id\_match)

REFERENCES Date\_Location(id\_match)

)

CREATE TABLE Goals(

id\_player int NOT NULL,

id\_match int NOT NULL,

Score int NOT NULL,

id\_club int NOT NULL,

PRIMARY KEY (id\_player, id\_club, id\_match),

CONSTRAINT fk\_id\_players FOREIGN KEY (id\_club, id\_player)

REFERENCES Players(id\_club, id\_player),

CONSTRAINT fk\_id\_match FOREIGN KEY (id\_match)

REFERENCES Date\_Location(id\_match)

)

CREATE TABLE Fans(

fan\_id int PRIMARY KEY,

login varchar(50) NOT NULL,

password varchar(50) NOT NULL,

Name varchar(50) NOT NULL,

Surname varchar(50) NOT NULL

)

CREATE TABLE Admins(

admin\_id int PRIMARY KEY,

login\_adm varchar(50) NOT NULL,

password\_adm varchar(50) NOT NULL,

)

GO

INSERT INTO Coaches(id\_coach, Country\_birth, Surname, Name)

VALUES

(1001, 'Германия', 'Тухель', 'Томас'),

(1002, 'Германия', 'Гиздоль', 'Маркус'),

(1003, 'Испания', 'Хави', 'Фернандо'),

(1004, 'Испания', 'Гвардиола', 'Хосеп'),

(1005, 'Италия', 'Анчелотти', 'Карло'),

(1006, 'Россия', 'Семак', 'Сергей'),

(1007, 'Германия', 'Нагельсманн', 'Юлиан'),

(1008, 'Аргентина', 'Почеттино', 'Маурисио'),

(1009, 'Россия', 'Ловчев', 'Евгений')

INSERT INTO Clubs(id\_club, Name\_club, Country, City, id\_coach)

VALUES

(101, 'Челси', 'Англия', 'Лондон', 1001),

(102, 'Локомотив', 'Россия', 'Москва', 1002),

(103, 'Барселона', 'Испания', 'Барселона', 1003),

(104, 'Манчестер Сити', 'Англия', 'Манчестер', 1004),

(105, 'Реал Мадрид', 'Испания', 'Мадрид', 1005),

(106, 'Зенит', 'Россия', 'Санкт-Петербург', 1006),

(107, 'Бавария Мюнхен', 'Германия', 'Мюнхен', 1007),

(108, 'ПСЖ', 'Франция', 'Париж', 1008)

INSERT INTO Players(id\_player, Surname, Name, Position, Number\_player, id\_club)

VALUES

(1, 'Менди', 'Эдуар', 'ГК', 1, 101),

(2, 'Алонсо', 'Маркус', 'ЛЗ', 27, 101),

(3, 'Силва', 'Тьяго', 'ЦЗ', 4, 101),

(4, 'Аспиликуэта', 'Сэзар', 'ЦЗ', 28, 101),

(5, 'Джеймс', 'Рис', 'ПЗ', 13, 101),

(6, 'Канте', 'Нголо', 'ЦОП', 7, 101),

(7, 'Ковачич', 'Матео', 'ЦП', 8, 101),

(8, 'Хаверц', 'Кай', 'ЦАП', 22, 101),

(9, 'Пулишич', 'Кристиан', 'ЛВ', 10, 101),

(10, 'Лукаку', 'Ромелу', 'ФРВ', 9, 101),

(11, 'Вернер', 'Тимо', 'ПВ', 11, 101),

(12, 'Гилерме', 'Маринато', 'ГК', 1, 102),

(13, 'Рыбус', 'Мацей', 'ЛЗ', 31, 102),

(14, 'Едвай', 'Тим', 'ЦЗ', 3, 102),

(15, 'Магкеев', 'Станислав', 'ЦЗ', 20, 102),

(16, 'Живоглядов', 'Дмитрий', 'ПЗ', 2, 102),

(17, 'Баринов', 'Дмитрий', 'ЦОП', 6, 102),

(18, 'Бека-Бека', 'Алексис', 'ЦОП', 8, 102),

(19, 'Миранчук', 'Антон', 'ЦАП', 11, 102),

(20, 'Камано', 'Франсуа', 'ЛВ', 10, 102),

(21, 'Кухта', 'Ян', 'ФРВ', 9, 102),

(22, 'Керк', 'Джирано', 'ПВ', 15, 102),

(23, 'Тер-Штеген', 'Марк', 'ГК', 1, 103),

(24, 'Альба', 'Жорди', 'ЛЗ', 30, 103),

(25, 'Пике', 'Жерар', 'ЦЗ', 3, 103),

(26, 'Гарсия', 'Эрик', 'ЦЗ', 4, 103),

(27, 'Алвес', 'Дани', 'ПЗ', 22, 103),

(28, 'Бускетс', 'Серхио', 'ЦОП', 8, 103),

(29, 'Де Йонг', 'Фрэнки', 'ЦП', 28, 103),

(30, 'Депай', 'Мемфис', 'ЦАП', 9, 103),

(31, 'Дембеле', 'Усман', 'ЛВ', 10, 103),

(32, 'Аубамеянг', 'Пьер-Емерик', 'ФРВ', 28, 103),

(33, 'Гонсалес', 'Николас', 'ПВ', 98, 103),

(34, 'Мораес', 'Эдерсон', 'ГК', 31, 104),

(35, 'Зинченко', 'Александр', 'ЛЗ', 11, 104),

(36, 'Диаш', 'Рубен', 'ЦЗ', 3, 104),

(37, 'Стоунз', 'Джон', 'ЦЗ', 4, 104),

(38, 'Уокер', 'Кайл', 'ПЗ', 5, 104),

(39, 'Гюндоган', 'Илкай', 'ЦП', 20, 104),

(40, 'Сильва', 'Бернардо', 'ЦАП', 21, 104),

(41, 'Марез', 'Рияд', 'ЦАП', 28, 104),

(42, 'Стерлинг', 'Рахим', 'ЛФД', 10, 104),

(43, 'Джезус', 'Габриэль', 'ФРВ', 9, 104),

(44, 'Фоден', 'Фил', 'ПФД', 15, 104),

(45, 'Куртуа', 'Тибо', 'ГК', 1, 105),

(46, 'Менди', 'Флориан', 'ЛЗ', 2, 105),

(47, 'Милитао', 'Эдер', 'ЦЗ', 3, 105),

(48, 'Фернандес', 'Начо', 'ЦЗ', 4, 105),

(49, 'Карвахаль', 'Дани', 'ПЗ', 5, 105),

(50, 'Каземиро', 'Мундо', 'ЦОП', 6, 105),

(51, 'Модрич', 'Лука', 'ЦП', 7, 105),

(52, 'Кроос', 'Тони', 'ЦП', 8, 105),

(53, 'Азар', 'Эден', 'ЛП', 10, 105),

(54, 'Бензема', 'Карим', 'ФРВ', 9, 105),

(55, 'Бэйл', 'Гарет', 'ПП', 11, 105),

(56, 'Крицюк', 'Станислав', 'ГК', 1, 106),

(57, 'Круговой', 'Даниил', 'ЛЗ', 5, 106),

(58, 'Ловрен', 'Деян', 'ЦЗ', 2, 106),

(59, 'Ракицкий', 'Ярослав', 'ЦЗ', 3, 106),

(60, 'Чистяков', 'Дмитрий', 'ЦЗ', 4, 106),

(61, 'Караваев', 'Вячеслав', 'ПЗ', 42, 106),

(62, 'Барриос', 'Вильмор', 'ЦОП', 6, 106),

(63, 'Оздоев', 'Магомед', 'ЦОП', 8, 106),

(64, 'Малком', 'Аларкон', 'ЦАП', 7, 106),

(65, 'Дзюба', 'Артём', 'ФРВ', 22, 106),

(66, 'Азмун', 'Сердар', 'ФРВ', 9, 106),

(67, 'Нойер', 'Мануэль', 'ГК', 1, 107),

(68, 'Дэвис', 'Альфонсо', 'ЛЗ', 2, 107),

(69, 'Зюле', 'Николас', 'ЦЗ', 3, 107),

(70, 'Эрнандес', 'Люка', 'ЦЗ', 4, 107),

(71, 'Павар', 'Бэнжамин', 'ПЗ', 5, 107),

(72, 'Горецка', 'Леон', 'ЦП', 8, 107),

(73, 'Мюллер', 'Томас', 'ЦП', 10, 107),

(74, 'Забицер', 'Марсель', 'ЦАП', 22, 107),

(75, 'Коман', 'Кингсли', 'ЛВ', 11, 107),

(76, 'Левандовски', 'Роберт', 'ФРВ', 9, 107),

(77, 'Гнабри', 'Серж', 'ПВ', 20, 107),

(78, 'Навас', 'Кейлор', 'ГК', 1, 108),

(79, 'Бернат', 'Хуан', 'ЛЗ', 23, 108),

(80, 'Маркиньос', 'Яго', 'ЦЗ', 3, 108),

(81, 'Диалло', 'Абду', 'ЦЗ', 4, 108),

(82, 'Хакими', 'Ашраф', 'ПЗ', 22, 108),

(83, 'Гей', 'Идрисса', 'ЦОП', 20, 108),

(84, 'Паредес', 'Леандро', 'ЦОП', 8, 108),

(85, 'Месси', 'Лионель', 'ЦАП', 31, 108),

(86, 'Неймар', 'Виста', 'ЛВ', 10, 108),

(87, 'Мбаппе', 'Килиан', 'ФРВ', 9, 108),

(88, 'Ди Мария', 'Анхель', 'ПВ', 11, 108)

INSERT INTO Location(id\_stadium, Name\_stadium, Country, City)

VALUES

(10, 'Стэмфорд Бридж', 'Англия', 'Лондон'),

(20, 'РЖД Арена', 'Россия', 'Москва'),

(30, 'Камп Ноу', 'Испания', 'Барселона'),

(40, 'Этихад Стэдиум', 'Англия', 'Манчестер'),

(50, 'Сантьяго Бернабеу', 'Испания', 'Мадрид'),

(60, 'Газпром Арена', 'Россия', 'Санкт-Петербург'),

(70, 'Альянц Арена', 'Германия', 'Мюнхен'),

(80, 'Парк де Пренс', 'Франция', 'Париж'),

(90, 'Велодром', 'Франция', 'Марсель'),

(100, 'Олд Траффорд', 'Англия', 'Манчестер'),

(110, 'Сан Сиро', 'Италия', 'Милан'),

(120, 'Открытие Арена', 'Россия', 'Москва'),

(130, 'Боррусия-парк', 'Германия', 'Дортмунд'),

(140, 'Эмирейтс Стэдиум', 'Англия', 'Лондон')

INSERT INTO Date\_Location(id\_match, Date\_time, id\_stadium, Name\_tournament)

VALUES

(10, CAST('19/04/2022 20:00:00' as smalldatetime), 10, 'Champions CUP'),

(11, CAST('19/04/2022 20:00:00' as smalldatetime), 10, 'Champions CUP'),

(20, CAST('19/04/2022 22:30:00' as smalldatetime), 20, 'Champions CUP'),

(21, CAST('19/04/2022 22:30:00' as smalldatetime), 20, 'Champions CUP'),

(30, CAST('20/04/2022 20:00:00' as smalldatetime), 30, 'Champions CUP'),

(31, CAST('20/04/2022 20:00:00' as smalldatetime), 30, 'Champions CUP'),

(40, CAST('20/04/2022 22:30:00' as smalldatetime), 50, 'Champions CUP'),

(41, CAST('20/04/2022 22:30:00' as smalldatetime), 50, 'Champions CUP'),

(50, CAST('03/05/2022 20:00:00' as smalldatetime), 80, 'Champions CUP'),

(51, CAST('03/05/2022 20:00:00' as smalldatetime), 80, 'Champions CUP'),

(60, CAST('03/05/2022 22:30:00' as smalldatetime), 70, 'Champions CUP'),

(61, CAST('03/05/2022 22:30:00' as smalldatetime), 70, 'Champions CUP'),

(70, CAST('04/05/2022 20:00:00' as smalldatetime), 60, 'Champions CUP'),

(71, CAST('04/05/2022 20:00:00' as smalldatetime), 60, 'Champions CUP'),

(80, CAST('04/05/2022 22:30:00' as smalldatetime), 40, 'Champions CUP'),

(81, CAST('04/05/2022 22:30:00' as smalldatetime), 40, 'Champions CUP')

INSERT INTO Results(id\_match, id\_club, Final\_Score)

VALUES

(10, 101, '2:1'),

(11, 108, '1:2'),

(20, 102, '3:1'),

(21, 107, '1:3'),

(30, 103, '1:0'),

(31, 106, '0:1'),

(40, 104, '2:0'),

(41, 105, '0:2'),

(50, 108, '4:0'),

(51, 101, '0:4'),

(60, 107, '1:1'),

(61, 102, '1:1'),

(70, 106, '0:3'),

(71, 103, '3:0'),

(80, 105, '2:2'),

(81, 104, '2:2')

INSERT INTO Goals(id\_match, id\_player, id\_club, Score)

VALUES

(10, 10, 101, 2),

(10, 84, 108, 1),

(20, 18, 102, 1),

(20, 19, 102, 1),

(20, 21, 102, 1),

(20, 69, 107, 1),

(30, 32, 103, 1),

(40, 36, 104, 1),

(40, 40, 104, 1),

(50, 85, 108, 1),

(50, 86, 108, 1),

(50, 87, 108, 2),

(60, 76, 107, 1),

(60, 14, 102, 1),

(70, 28, 103, 1),

(70, 30, 103, 2),

(80, 54, 105, 1),

(80, 52, 105, 1),

(80, 44, 104, 2)

GO

INSERT INTO Fans

VALUES

(1, 'samokhin28', '1511', 'Красавец', 'Вопупонин')

GO

INSERT INTO Admins

VALUES

(1, 'komar1511', '1511')

use Football

go

UPDATE Coaches

SET ImageCoach =

(SELECT \* FROM OPENROWSET(BULK N'C:\Coach\tuchel.jpg', SINGLE\_BLOB) AS image)

WHERE id\_coach = 1001

use Football

go

UPDATE Coaches

SET ImageCoach =

(SELECT \* FROM OPENROWSET(BULK N'C:\Coach\gisdol.jpg', SINGLE\_BLOB) AS image)

WHERE id\_coach = 1002

use Football

go

UPDATE Coaches

SET ImageCoach =

(SELECT \* FROM OPENROWSET(BULK N'C:\Coach\xavi.jpg', SINGLE\_BLOB) AS image)

WHERE id\_coach = 1003

use Football

go

UPDATE Coaches

SET ImageCoach =

(SELECT \* FROM OPENROWSET(BULK N'C:\Coach\guardiola.jpg', SINGLE\_BLOB) AS image)

WHERE id\_coach = 1004

use Football

go

UPDATE Coaches

SET ImageCoach =

(SELECT \* FROM OPENROWSET(BULK N'C:\Coach\anchelotti.jpg', SINGLE\_BLOB) AS image)

WHERE id\_coach = 1005

use Football

go

UPDATE Coaches

SET ImageCoach =

(SELECT \* FROM OPENROWSET(BULK N'C:\Coach\nagelsmann.jpg', SINGLE\_BLOB) AS image)

WHERE id\_coach = 1007

use Football

go

UPDATE Coaches

SET ImageCoach =

(SELECT \* FROM OPENROWSET(BULK N'C:\Coach\pochettino.jpg', SINGLE\_BLOB) AS image)

WHERE id\_coach = 1008

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\Stamford.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 10

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\RZD.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 20

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\Camp.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 30

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\Etihad.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 40

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\Santiago.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 50

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\Gazprom.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 60

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\Allianz.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 70

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\Parc.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 80

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\Velodrom.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 90

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\Old.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 100

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\Sansiro.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 110

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\Otkritie.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 120

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\Borussia.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 130

use Football

go

UPDATE Location

SET ImageStadium =

(SELECT \* FROM OPENROWSET(BULK N'C:\Stadium\Emirates.jpg', SINGLE\_BLOB) AS image)

WHERE id\_stadium = 140

USE Football

GO

IF OBJECT\_ID ( 'Top\_Scorer\_Clubs3', 'P' ) IS NOT NULL

DROP PROCEDURE Top\_Scorer\_Clubs3;

GO

CREATE PROCEDURE Top\_Scorer\_Clubs3

AS

SELECT TOP (3) dbo.Clubs.country, dbo.Clubs.city, dbo.Clubs.name\_club, SUM(dbo.Goals.Score) AS Score

FROM dbo.Clubs INNER JOIN

dbo.Goals ON dbo.Clubs.id\_club = dbo.Goals.id\_club

GROUP BY dbo.Clubs.country, dbo.Clubs.city, dbo.Clubs.name\_club

ORDER BY Score DESC

GO

IF OBJECT\_ID ( 'Top\_Scorer5', 'P' ) IS NOT NULL

DROP PROCEDURE Top\_Scorer5;

GO

CREATE PROCEDURE Top\_Scorer5

AS

SELECT TOP (5) dbo.Players.Surname, dbo.Players.Name, dbo.Players.Number\_player, dbo.Players.Position, dbo.Clubs.name\_club, SUM(dbo.Goals.Score) AS Score

FROM dbo.Clubs INNER JOIN

dbo.Players ON dbo.Clubs.id\_club = dbo.Players.id\_club INNER JOIN

dbo.Goals ON dbo.Players.id\_club = dbo.Goals.id\_club AND dbo.Players.id\_player = dbo.Goals.id\_player

GROUP BY dbo.Players.Surname, dbo.Players.Name, dbo.Players.Number\_player, dbo.Players.Position, dbo.Clubs.name\_club

ORDER BY Score DESC, dbo.Players.Number\_player ASC

GO

IF OBJECT\_ID ( 'Top\_ScorerAll', 'P' ) IS NOT NULL

DROP PROCEDURE Top\_ScorerAll;

GO

CREATE PROCEDURE Top\_ScorerAll

AS

SELECT dbo.Players.Surname, dbo.Players.Name, dbo.Players.Number\_player, dbo.Players.Position, dbo.Clubs.name\_club, SUM(dbo.Goals.Score) AS Score

FROM dbo.Clubs INNER JOIN

dbo.Players ON dbo.Clubs.id\_club = dbo.Players.id\_club INNER JOIN

dbo.Goals ON dbo.Players.id\_club = dbo.Goals.id\_club AND dbo.Players.id\_player = dbo.Goals.id\_player

GROUP BY dbo.Players.Surname, dbo.Players.Name, dbo.Players.Number\_player, dbo.Players.Position, dbo.Clubs.name\_club

ORDER BY Score DESC, dbo.Players.Number\_player ASC

GO

IF OBJECT\_ID ( 'Coaches\_proc', 'P' ) IS NOT NULL

DROP PROCEDURE Coaches\_proc;

GO

CREATE PROCEDURE Coaches\_proc

AS

SELECT dbo.Coaches.Surname, dbo.Coaches.Name, dbo.Coaches.Country\_birth, dbo.Coaches.ImageCoach, dbo.Clubs.name\_club

FROM dbo.Coaches INNER JOIN

dbo.Clubs ON dbo.Coaches.id\_coach = dbo.Clubs.id\_coach

GO

IF OBJECT\_ID ( 'Score\_In\_Match', 'P' ) IS NOT NULL

DROP PROCEDURE Score\_In\_Match;

GO

CREATE PROCEDURE Score\_In\_Match

AS

SELECT DISTINCT dbo.Clubs.name\_club, dbo.Players.Surname, dbo.Players.Position, dbo.Goals.Score, goals.id\_match

FROM dbo.Goals INNER JOIN

dbo.Players ON dbo.Goals.id\_club = dbo.Players.id\_club AND dbo.Goals.id\_player = dbo.Players.id\_player INNER JOIN

dbo.Results ON dbo.Goals.id\_club = dbo.Results.id\_club INNER JOIN

dbo.Clubs ON dbo.Players.id\_club = dbo.Clubs.id\_club AND dbo.Results.id\_club = dbo.Clubs.id\_club

ORDER BY Goals.id\_match

GO

IF OBJECT\_ID ( 'Match\_Location', 'P' ) IS NOT NULL

DROP PROCEDURE Match\_Location;

GO

CREATE PROCEDURE Match\_Location

AS

SELECT dbo.Results.Final\_Score, dbo.Location.City, dbo.Clubs.name\_club, dbo.Location.Name\_stadium, dbo.Results.id\_match, dbo.Date\_Location.Date\_time

FROM dbo.Clubs INNER JOIN

dbo.Results ON dbo.Clubs.id\_club = dbo.Results.id\_club INNER JOIN

dbo.Date\_Location ON dbo.Results.id\_match = dbo.Date\_Location.id\_match INNER JOIN

dbo.Location ON dbo.Date\_Location.id\_stadium = dbo.Location.id\_stadium

ORDER BY dbo.Results.id\_match

# ПРИЛОЖЕНИЕ Б. КОД ПРОГРАММЫ

Отмечу здесь несколько из важных окон моей программы.

## Код окна «регистрация»

XAML:

<Window x:Class="FootballAppListView.Reg\_Window"

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

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

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:FootballAppListView"

mc:Ignorable="d"

Title="Reg\_Window" Height="450" Width="800">

<Grid>

<Grid.Background>

<ImageBrush ImageSource="Resources/Football\_Reg.jpg"></ImageBrush>

</Grid.Background>

<Grid HorizontalAlignment="Left" Height="320" Margin="10,25,0,0" VerticalAlignment="Top" Width="771">

<TextBlock HorizontalAlignment="Left" Margin="609,51,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Height="48" Width="90" Text="Логин" FontSize="24" FontWeight="Heavy"/>

<TextBox x:Name="PasswordText" Text="{Binding password}" HorizontalAlignment="Left" Height="46" Margin="564,182,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="160" FontSize="18"/>

</Grid>

<TextBox x:Name="SurnameText" Text="{Binding Surname}" HorizontalAlignment="Left" Height="48" Margin="126,111,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="160" FontSize="18"/>

<TextBox x:Name="NameText" Text="{Binding Name}" HorizontalAlignment="Left" Height="46" Margin="126,207,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="160" FontSize="18"/>

<TextBlock FontWeight="Heavy" HorizontalAlignment="Left" Margin="10,109,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Height="48" Width="139" Text="Фамилия" FontSize="23"/>

<TextBlock FontWeight="Heavy" HorizontalAlignment="Left" Margin="62,209,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Height="44" Width="59" FontSize="24" RenderTransformOrigin="0.423,0.303"><Run Text="И"/><Run Text="мя"/></TextBlock>

<Button Content="Зарегистрироваться" FontWeight="Bold" Background="DarkGreen" HorizontalAlignment="Left" Margin="342,349,0,0" VerticalAlignment="Top" Width="125" Click="Button\_Click" Height="33"/>

<TextBox x:Name="LoginText" Text="{Binding login}" HorizontalAlignment="Left" Height="46" Margin="574,111,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="160" FontSize="18"/>

<Button Content="Назад" Name="BtnBack" Click="BtnBack\_Click" Margin="678,10,10,361" FontSize="24"/>

<TextBlock FontWeight="Heavy" HorizontalAlignment="Left" Margin="608,174,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Height="48" Width="126" Text="Пароль" FontSize="24"/>

</Grid>

</Window>

C#:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace FootballAppListView

{

/// <summary>

/// Логика взаимодействия для Reg\_Window.xaml

/// </summary>

public partial class Reg\_Window : Window

{

string \_login;

private Fans \_currentFans = new Fans();

int maxid = FootballEntities.GetContext().Fans.Max(u => u.fan\_id);

public Reg\_Window(Fans SelectFans)

{

InitializeComponent();

if (SelectFans != null)

\_currentFans = SelectFans;

DataContext = \_currentFans;

}

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

{

\_login = LoginText.Text;

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentFans.login)) errors.AppendLine("Введите логин");

if (string.IsNullOrWhiteSpace(\_currentFans.password)) errors.AppendLine("Введите пароль");

if (string.IsNullOrWhiteSpace(\_currentFans.Surname)) errors.AppendLine("Введите фамилию");

if (string.IsNullOrWhiteSpace(\_currentFans.Name)) errors.AppendLine("Введите имя");

if (FootballEntities.GetContext().Fans.Any(f => f.login == \_login)) errors.AppendLine("Пользователь с таким логином уже существует");

//проверка пароля:

string str2; int i = 0; bool boo; int yes = 0;

if (\_currentFans.password.Length < 6) errors.AppendLine("Пароль должен быть длиннее 6 символов");

str2 = \_currentFans.password.ToLower();

if (\_currentFans.password == str2) errors.AppendLine("В пароле должны быть большие буквы");

char[] array = \_currentFans.password.ToCharArray();

while (\_currentFans.password.Length > i)

{

boo = Char.IsDigit(array[i]);

if (boo == true) yes = yes + 1;

i = i + 1;

}

if (\_currentFans.password.Length <= yes) errors.AppendLine("Пароль должен включать в себя ещё и буквы, большие и маленькие");

if (yes == 0) errors.AppendLine("Пароль должен включать в себя ещё и цифры");

if (errors.Length > 0) { MessageBox.Show(errors.ToString()); return;

}

\_currentFans.fan\_id = maxid + 1;

FootballEntities.GetContext().Fans.Add(\_currentFans);

try

{

FootballEntities.GetContext().SaveChanges();

MessageBox.Show("Okey!");

StartWindow win1 = new StartWindow();

win1.Show();

this.Close();

}

catch (Exception ex)

{

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

}

}

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

{

StartWindow win1 = new StartWindow();

win1.Show();

this.Close();

}

}

}

## Код окна «бомбардиры»

XAML:

<Window x:Class="FootballAppListView.Strikers\_Window"

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

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

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:FootballAppListView"

mc:Ignorable="d"

Title="Strikers\_Window" Height="450" Width="800">

<Window.Background>

<ImageBrush ImageSource="C:\Users\komar1511\source\repos\FootballAppListView\FootballAppListView\Resources\Football\_Strikers.jpg"/>

</Window.Background>

<Grid>

<Button Content="Назад" Name="BtnBack" Click="BtnBack\_Click" Margin="678,10,10,361" FontSize="24"/>

<Button Content="Выбрать" Name="BtnClick" Click="Btn\_Click" Margin="78,90,596,289" FontSize="24"/>

<Frame BorderThickness="0" x:Name="StrikersFrame" HorizontalAlignment="Left" Height="412" Margin="270,0,0,0" VerticalAlignment="Top" Width="344" NavigationUIVisibility="Hidden"/>

<ComboBox HorizontalContentAlignment="Center" Name="ComboStrikers" HorizontalAlignment="Left" Margin="49,59,0,0" VerticalAlignment="Top" Width="171" Height="26" SelectionChanged="ComboStrikers\_SelectionChanged"/>

</Grid>

</Window>

C#:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace FootballAppListView

{

/// <summary>

/// Логика взаимодействия для Window1.xaml

/// </summary>

public partial class Strikers\_Window : Window

{

public Strikers\_Window()

{

InitializeComponent();

Strikers\_class.StrikersFrame = StrikersFrame;

ComboStrikers.Items.Add("ТОП-5");

ComboStrikers.Items.Add("Все бомбардиры");

}

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

{

}

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

{

Main\_Window2 win3 = new Main\_Window2();

win3.Show();

this.Close();

}

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

{

if (ComboStrikers.SelectedIndex == 0) //первый элемент списка

{

Strikers\_class.StrikersFrame.Content = new Strikers\_Page1();

}

if (ComboStrikers.SelectedIndex == 1)//второй элемент списка

{

Strikers\_class.StrikersFrame.Content = new Strikers\_Page2();

}

}

}

}

### Страница ТОП-5 бомбардиров

XAML:

<Page x:Class="FootballAppListView.Strikers\_Page1"

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

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

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:FootballAppListView"

mc:Ignorable="d"

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

Title="Strikers\_Page1">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="373"/>

<RowDefinition Height="73"/>

</Grid.RowDefinitions>

<DataGrid x:Name="DGridStrikers" BorderThickness="0" BorderBrush="Transparent" FontWeight="DemiBold" Background="Transparent" RowBackground="LightBlue" AutoGenerateColumns="False" IsReadOnly="True" Margin="10,0,0,73" Grid.RowSpan="2">

<DataGrid.Columns>

<DataGridTextColumn Width="\*" Header="Клуб" Binding="{Binding name\_club}" ></DataGridTextColumn>

<DataGridTextColumn Width="\*" Header="Фамилия" Binding="{Binding Surname}"></DataGridTextColumn>

<DataGridTextColumn Width="\*" Header="Голы" Binding="{Binding Score}" ></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid>

<Frame x:Name="StrikersFrame" Margin="0,0,0,110.5" NavigationUIVisibility="Hidden"/>

</Grid>

</Page>

C#:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace FootballAppListView

{

/// <summary>

/// Логика взаимодействия для Strikers\_Page1.xaml

/// </summary>

public partial class Strikers\_Page1 : Page

{

public Strikers\_Page1()

{

InitializeComponent();

DGridStrikers.ItemsSource = FootballEntities.GetContext().Top\_Scorer5().ToList();

}

}

}

### Страница всех бомбардиров

XAML:

<Page x:Class="FootballAppListView.Strikers\_Page2"

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

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

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:FootballAppListView"

mc:Ignorable="d"

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

Title="Strikers\_Page2">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="373"/>

<RowDefinition Height="73"/>

</Grid.RowDefinitions>

<DataGrid x:Name="DGridStrikers" BorderThickness="0" BorderBrush="Transparent" FontWeight="DemiBold" Background="Transparent" RowBackground="LightBlue" AutoGenerateColumns="False" IsReadOnly="True" Margin="10,0,0,-2" Grid.RowSpan="2">

<DataGrid.Columns>

<DataGridTextColumn Width="\*" Header="Клуб" Binding="{Binding name\_club}" ></DataGridTextColumn>

<DataGridTextColumn Width="\*" Header="Фамилия" Binding="{Binding Surname}"></DataGridTextColumn>

<DataGridTextColumn Width="\*" Header="Голы" Binding="{Binding Score}" ></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid>

<Frame x:Name="StrikersFrame" NavigationUIVisibility="Hidden" Grid.RowSpan="2"/>

</Grid>

</Page>

C#:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace FootballAppListView

{

/// <summary>

/// Логика взаимодействия для Strikers\_Page2.xaml

/// </summary>

public partial class Strikers\_Page2 : Page

{

public Strikers\_Page2()

{

InitializeComponent();

DGridStrikers.ItemsSource = FootballEntities.GetContext().Top\_ScorerAll().ToList();

}

}

}

## Код окна «игроки»

XAML:

<Window x:Class="FootballAppListView.Players\_Window"

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

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

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:FootballAppListView"

mc:Ignorable="d"

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

<Window.Background>

<ImageBrush ImageSource="Resources\Football\_pitch.jpg"/>

</Window.Background>

<Grid>

<DataGrid x:Name="DGridPlayers" BorderThickness="0" BorderBrush="Transparent" FontWeight="DemiBold" Background="Transparent" RowBackground="LightCoral" AutoGenerateColumns="False" IsReadOnly="True" Margin="10,66,0,40.5" VerticalScrollBarVisibility="Hidden">

<DataGrid.Columns>

<DataGridTextColumn Width="\*" Header="Клуб" Binding="{Binding Clubs.name\_club}"></DataGridTextColumn>

<DataGridTextColumn Width="\*" Header="Фамилия" Binding="{Binding Surname}"></DataGridTextColumn>

<DataGridTextColumn Width="\*" Header="Имя" Binding="{Binding Name}" ></DataGridTextColumn>

<DataGridTextColumn Width="\*" Header="Позиция" Binding="{Binding Position}" ></DataGridTextColumn>

<DataGridTextColumn Width="\*" Header="Номер игрока" Binding="{Binding Number\_player}" ></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid>

<Button Content="Назад" Name="BtnBack" Click="BtnBack\_Click" Margin="678,10,10,361" FontSize="24"/>

<TextBox x:Name="FindPlayers" HorizontalAlignment="Left" Height="25" Margin="33,387,0,0" TextWrapping="Wrap" Text="Поиск по клубу" VerticalAlignment="Top" Width="155" TextChanged="FindPlayers\_TextChanged"/>

<TextBox x:Name="FindPlayersName" HorizontalAlignment="Left" Height="25" Margin="592,387,0,0" TextWrapping="Wrap" Text="Поиск по фамилии" VerticalAlignment="Top" Width="155" TextChanged="FindPlayersName\_TextChanged"/>

</Grid>

</Window>

C#:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace FootballAppListView

{

/// <summary>

/// Логика взаимодействия для club0.xaml

/// </summary>

public partial class Players\_Window : Window

{

string \_name;

public Players\_Window()

{

InitializeComponent();

DGridPlayers.ItemsSource = FootballEntities.GetContext().Players.ToList();

}

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

{

Main\_Window2 win6 = new Main\_Window2();

win6.Show();

this.Close();

}

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

{

Clubs Name = null;

\_name = FindPlayers.Text;

Name = FootballEntities.GetContext().Clubs.Where(b => b.name\_club == \_name).FirstOrDefault();

if (Name == null)

{

DGridPlayers.ItemsSource = FootballEntities.GetContext().Players.ToList();

}

else

{

DGridPlayers.ItemsSource = FootballEntities.GetContext().Players.Where(b => b.Clubs.name\_club == \_name).ToList();

}

}

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

{

Players Name = null;

\_name = FindPlayersName.Text;

Name = FootballEntities.GetContext().Players.Where(b => b.Surname == \_name).FirstOrDefault();

if (Name == null)

{

DGridPlayers.ItemsSource = FootballEntities.GetContext().Players.ToList();

}

else

{

DGridPlayers.ItemsSource = FootballEntities.GetContext().Players.Where(b => b.Surname == \_name).ToList();

}

}

}

}

И т.д.