Константинов Иван ББИ 164

1. **Координация в футбольной трансферной сети.**

2. <https://github.com/Konstantinov10/Konstantinov_KDZ>

3.Одиночный проект.

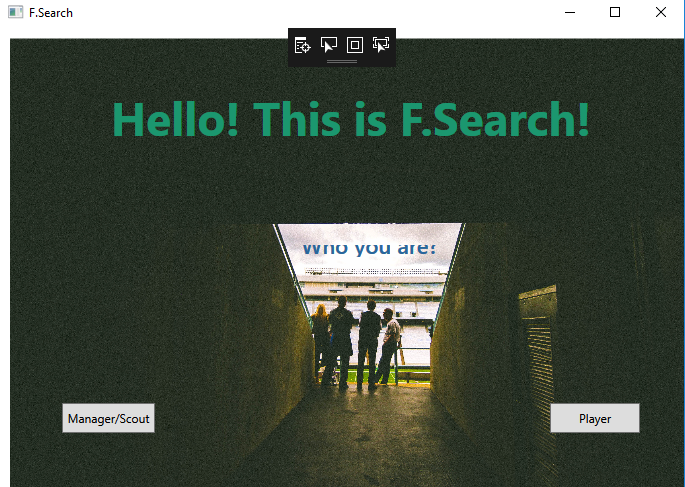
4. Программа "F.Search" создана для двух типов пользователей - "игроки" и "менеджеры/скауты". Для обоих есть возможность авторизации, которая позволяет перейти к функционалу. "Игроки" могут указать свои параметры и попасть в трансферную сеть, которая будет отображаться для всех менеджеров, зарегистрированных в приложении. Также, в случае ошибки в указанных параметрах , у игроков есть возможность исправить ее, а изменения моментально сохранятся. "Менеджеры" могут смотреть трансферную сеть, в которой может осуществляться поиск по всем параметрам, вывод всех игроков, зарегистрированных в "F.Search", полная очистка (для нового поиска) и выборочное удаление "ненужных игроков".Таким образом, футболисты получают шанс заинтересовать менеджера топ-клуба, а сами скауты - усилить свою команду.

5. Для создания "F.Search" использовались:

* Графическое приложение (WPF)
* Операции с данными: создание, редактирование, удаление, чтение, (CRUD) + поиск
* Хранение данных в файле
* Защита от некорректного ввода
* Единый стиль именования (Naming Convention)
* GitHub
* Несколько сущностей
* Сериализация
* Страничная навигация (по аналогии с мобильным приложением)
* Авторизация
* Адаптивность

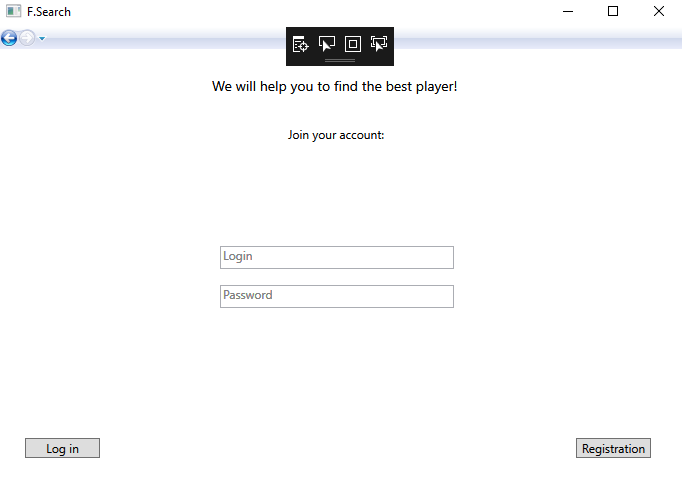
6. Интерфейс программы:

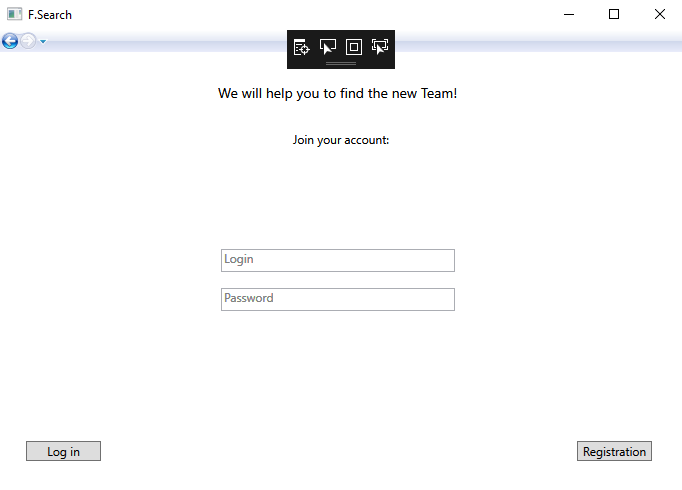
Стартовая страница:



-Позволяет определить тип пользователя с помощью нажатия на соответствующую кнопку.

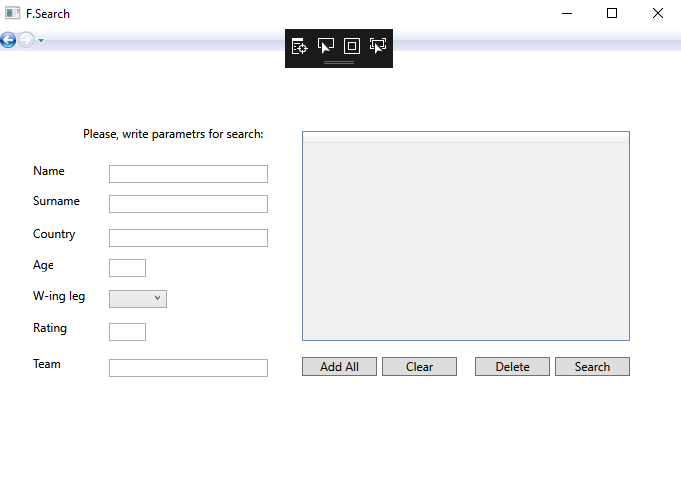
Страницы авторизации:





-Позволяют зарегистрироваться и (или) войти в приложение

Страница менеджера:



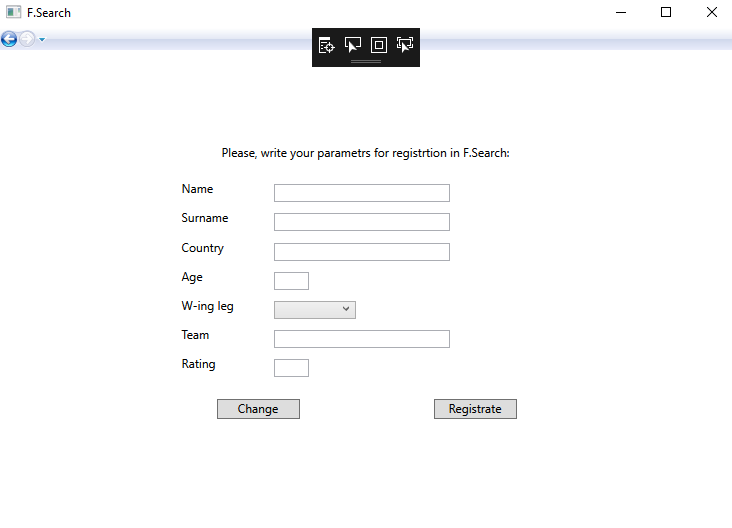
-"Add All" показывает всех игроков, зарегистрированных в "F.Search"

-"Clear" очищает dataGrid

-"Delete" позволяет удалить выбранного из выведенных на показ игроков

-"Search" ищет игроков по параметрам, которые нужно указать в textBox или comboBox слева от dataGrid.

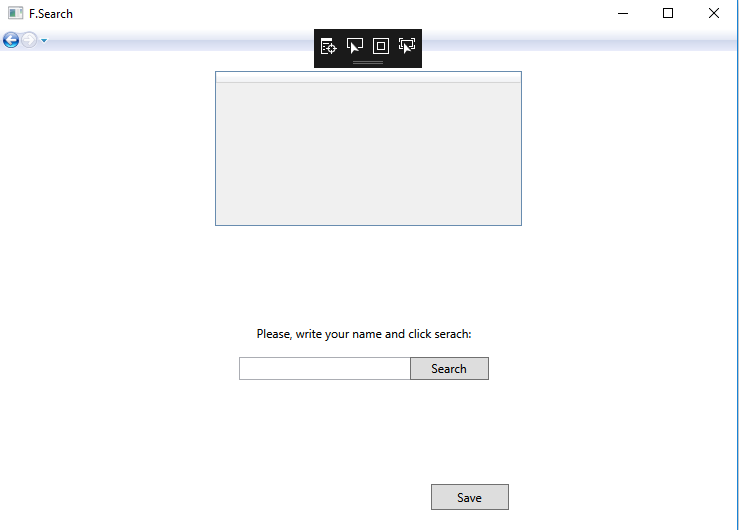
Страница игрока:



-"Registrate" добавляет игрока в трансферную сеть "F.Search", параметры заполняются выше.

-"Change" отправляет пользователя на страницу внесения изменений

Страница внесения изменений:



-Пользователь пишет свое имя, нажимает кнопку "Search"

-Далее можно вносить любые корректировки в параметры

-Как только ошибки исправлены, пользователь может сохранить изменения с помощью кнопки "Save", после чего эти изменения будут в трансферной сети

7. Состав классов:

Класс "Player":

{

[Serializable]

public class Player

{

private string \_name;

public string Name

{

get { return \_name; }

set { \_name = value; }

}

private string \_surname;

public string Surname

{

get { return \_surname; }

set { \_surname = value; }

}

private string \_wleg;

public string Wleg

{

get { return \_wleg; }

set { \_wleg = value; }

}

private string \_country;

public string Country

{

get { return \_country; }

set { \_country = value; }

}

private int \_age;

public int Age

{

get { return \_age; }

set { \_age = value; }

}

private int \_rating;

public int Rating

{

get { return \_rating; }

set { \_rating = value; }

}

private string \_team;

public string Team

{

get { return \_team; }

set { \_team = value; }

}

public string PlayerasLine()

{

string line = null;

line += \_name;

line += " " + \_surname;

line += " " + \_country;

line += " " + \_age;

line += " " + \_wleg;

line += " " + \_rating;

line += " " + \_team;

return line;

}

public Player(string name, string surname, string country, int age, string wleg, int rating, string team)

{

\_name = name;

\_surname = surname;

\_country = country;

\_age = age;

\_wleg = wleg;

\_rating = rating;

\_team = team;

}

}

}

Данный класс дает возможность "собрать" футболиста по его отдельным параметрам(имя, фамилия, страна, возраст, команда, рейтинг), прописывается также возможность сериализации данного класса.

Класс "Log\_Pas":

{

[Serializable]

public class Log\_Pas

{

private string \_login;

public string Login

{

get { return \_login; }

set { \_login = value; }

}

private string \_password;

public string Password

{

get { return \_password; }

set { \_password = value; }

}

public Log\_Pas( string login, string password)

{

\_login = login;

\_password = password;

}

}

}

Представляет собой собранные вместе логин и пароль, также сериализуем. Данный класс используется в авторизации пользователей.

Класс "Pages":

{

class Pages

{

private static MainPage \_mainPage = new MainPage();

private static StartPlayerPage \_startPlayerPage = new StartPlayerPage();

private static StartManagerPage \_startManagerPage = new StartManagerPage();

private static RegPlayerPage \_regPlayerPage = new RegPlayerPage();

private static PlayerChangePage \_playerChangePage = new PlayerChangePage();

private static SearchManagerPage \_searchManagerPage = new SearchManagerPage();

public static MainPage MainPage

{

get

{

return \_mainPage;

}

}

public static StartPlayerPage StartPlayerPage

{

get

{

return \_startPlayerPage;

}

}

public static StartManagerPage StartManagerPage

{

get

{

return \_startManagerPage;

}

}

public static RegPlayerPage RegPLayerPage

{

get

{

return \_regPlayerPage;

}

}

public static PlayerChangePage PlayerChangePage

{

get

{

return \_playerChangePage;

}

}

public static SearchManagerPage SearchManagerPage

{

get

{

return \_searchManagerPage;

}

}

}

}

Класс для работы страничной навигации (принимает и возвращает страницы).

8. Список используемых источников:

* <http://www.cyberforum.ru>
* <https://msdn.microsoft.com/ru-ru/default.aspx>
* <http://www.programmersforum.ru>
* <https://professorweb.ru>