

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра інформаційних систем та технологій

Курсова робота

З дисципліни «Програмування - 1. Основи програмування»

Тема: Плеєр

Керівник

ст. викл. Яланецький В. А.

Допущений до захисту»

(Особистий підпис керівника)

« » _____ 2022р.

Захищений з оцінкою

_____ (оцінка)

Виконавець

ст. Маркін Є.Д.

залікова книжка № ІС-1121

гр. ІС-11

(особистий підпис виконавця)

« » _____ 2022 р.

Члени комісії:

(особистий підпис) (розшифровка підпису)

(особистий підпис) (розшифровка підпису)

Київ – 2022

ЛИСТ-ЗАВДАННЯ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ
СІКОРСЬКОГО»

(назва навчального закладу)

Кафедра ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

Дисципліна «Основи програмування»

Курс 1 Група ІС-11 Семестр 2

ЗАВДАННЯ

на курсову роботу студента

Маркіна Єгора Дмитровича

1. Тема роботи: *Плеєр*

2. Строк здачі студентом закінченої роботи: **22.06.2022**

3. Вихідні дані до роботи: *Інструментарій: мова C#; середовище VS; інтерфейс WinForm; Цільові вимоги: зберігати списки слухачів та треків у текстових файлах; один слухач може замовити один трек, доки не поверне назад; виконавець може додати багато треків;*

4. Зміст розрахунково – пояснювальної записки (перелік питань, що підлягають розробці):

1) Розглянути існуючі рішення та визначити сутності об'єкта розробки.

2) Розробити класи для дій на кожній формі;

3) Визначити кожній сутності її атрибути та ключ

4) Побудувати діаграму зв'язків «сутність – зв'язок» (ER–діаграма)

5) Запрограмувати функціонал та інтерфейс мінімально-працездатного застосунку

6) Сформулювати інструкцію користувача мінімально-працездатного застосунку

7) Протестувати мінімально-працездатний застосунок

Додатки: Вихідний код програми; Приклад виконання застосунку

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):

Діаграма класів. Діаграма зв'язків «сутність – зв'язок» (ER–діаграма)

6. Дата видачі завдання: **13.04.2022**

КАЛЕНДАРНИЙ ПЛАН:

№, п/п	Назва етапів виконання курсової роботи	Строк Виконання етапів роботи	Підписи або примітки
1.	<i>Аналіз предметної області проєктування та огляд існуючих рішень</i>	16.04.2022	
2.	<i>Об'єктно-орієнтоване проєктування архітектури застосунку</i>	01.05.2022	
3.	<i>Розробка функціоналу та інтерфейсу застосунку</i>	15.05.2022	
4.	<i>Формування інструкції користувача застосунком</i>	01.06.2022	
5.	<i>Тестування мінімально-працездатного застосунку</i>	07.06.2022	
6.	<i>Оформлення пояснювальної записки та Додатків</i>	15.06.2022	
7.	<i>Передзахист курсової роботи</i>	22.06.2022	
8.	<i>Захист курсової роботи</i>	27.06.2022	

Студент _____
(підпис)

Єгор МАРКІН

Керівник _____
(підпис)

Валерій ЯЛАНЕЦЬКИЙ

« ____ » _____ 20 ____ р.

АНОТАЦІЯ

Маркін Є.Д. Графічний інтерфейс плеєра. КПІ ім. Ігоря Сікорського, Київ, 2022.

Робота містить 36 сторінок, 34 рисунок, 6 таблиць, 3 посилання на джерела та посилання на код роботи.

Ключові слова: графічний інтерфейс, плеєр, трек, виконавець Windows Forms, C#.

Об'єкт розробки: плеєр.

Мета розробки: створення зручного, інтуїтивно зрозумілого інтерфейсу для прослуховування та роботи з треками.

У курсовій роботі розроблено графічний інтерфейс роботи з плеєром. Дану задачу було реалізовано за допомогою Windows Forms та мови програмування C#.

Результат роботи може бути корисним для створення майбутніх додатків на основі цього проєкту.

ЗМІСТ

ВСТУП.....	6
1 ВИМОГИ ДО СИСТЕМИ ТА ОПИС ІНСТРУМЕНТІВ РОЗРОБКИ.....	7
1.1 Функціональні вимоги до системи.....	7
1.2 Нефункціональні вимоги до системи.....	7
1.3 Опис інструментів розробки.....	7
2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ.....	8-14
2.1 Діаграма прецедентів.....	8
2.2 Опис сценаріїв використання системи.....	9-14
3 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ.....	15-20
3.1 Загальна структура проекту.....	15
3.2 Компоненти рівня доступу до даних.....	16-20
4 ІНСТРУКЦІЯ КОРИСТУВАЧА.....	21-24
5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	25
ВИСНОВКИ.....	26
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	27
ДОДАТКИ.....	28-36

ВСТУП

Дану тему можна вважати актуальною, оскільки в сучасності майже кожна людина слухає музику, по дорозі, під час роботи чи просто відпочиваючи. Тому даний застосунок є актуальним та матиме попит серед великою кількості користувачів.

Об'єктом дослідження є автоматизація використання плеєра.

Предметом дослідження є робота з плеєром.

Метою роботи є створення програми, що дозволяє додавати, переключати треки та виконувати дії над ними. Для цього система повинна мати певний перелік властивостей та вирішувати такі задачі:

Перегляд треків, створення нового трека, видалення трека.

Завданням роботи є створення графічного інтерфейсу для роботи з плеєром.

Практичне значення курсової роботи полягає в тому, що робота може бути використана у якості основи плеєра для створення подібних додатків.

Структура роботи: дана робота складається з шести розділів, вступу, висновків та використаних джерел. Теж у роботі присутнє лист-завдання та анотація. Робота містить 36 сторінок, 34 рисунки, 6 таблиць, 3 посилання на джерела та посилання на код роботи.

У даній роботі застосовані Windows Forms у додатку Visual Studio 2019, мова програмування C# з її основними функціями та властивостями.

1 ВИМОГИ ДО СИСТЕМИ ТА ОПИС ІНСТРУМЕНТІВ РОЗРОБКИ

1.1 Функціональні вимоги до системи

Система має відповідати наступним функціональним вимогам:

- користувач має мати доступ до всіх треків.
- користувач може замовити трек.
- користувач може подивитися додаткову інформацію про трек.

1.2 Нефункціональні вимоги до системи

Система має відповідати наступним функціональним вимогам:

- система повинна мати відкриту архітектуру;
- інтерфейс користувача має бути зручним та інтуїтивно-зрозумілим;

1.3 Опис інструментів розробки

IDE: Visual Studio 2019

Core Framework: .NET Core 4.7.2

UI: Windows Forms

Version Control: GitHub

<https://github.com/TastyRoti/spoty>

Language: C#

2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ

2.1 Діаграма прецедентів

Діаграма прецедентів системи представлена на рисунку 2.1.

Актором є користувач системи.

Користувачу доступна вся функціональність програми.

Детально всі сценарії використання описані у наступному підрозділі.

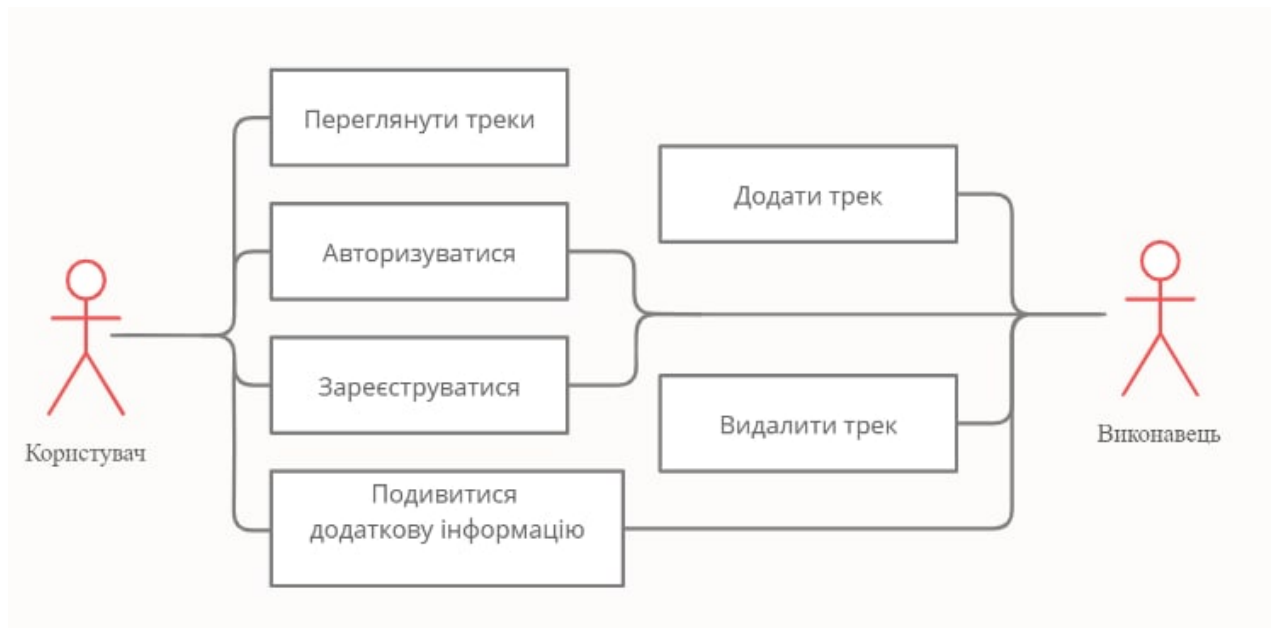


Рисунок 2.1.1—діаграма прецедентів.

2.2 Опис сценаріїв використання системи

Детальні описи сценаріїв використання наведено у таблицях 2.1-2.6.

Таблиця 2.1 – Сценарій використання «авторизація».

Назва	Авторизація
ID	1
Опис	Користувач, використовуючи форму авторизується в програму
Актори	Користувач, виконавець
Вигоди	Кожен користувач має свій особистий, захищений обліковий запис
Частота користування	Постійно
Тригери	Користувач вводить логін і пароль у поля
Передумови	Поля для авторизації доступні у початковій формі
Постумови	Користувач авторизується
Основний розвиток	Користувач вводить логін і пароль, натискає на кнопку авторизації
Альтернативні розвитку	Користувач натискає на кнопку реєстрації, та переходить до вікна створення нового облікового запису
Виняткові ситуації	Користувач вводить неправильний логін або пароль

Таблиця 2.2 – Сценарій використання «реєстрація».

Назва	Реєстрація
ID	2
Опис	Користувач, використовуючи форму, створює новий обліковий запис
Актори	Користувач, виконавець
Вигоди	Кожен користувач має свій особистий, захищений обліковий запис
Частота користування	Постійно
Тригери	Користувач вводить логін і пароль у поля і створює новий обліковий запис
Передумови	Поля для реєстрації доступні у формі реєстрації
Постумови	Користувач створює новий обліковий запис
Основний розвиток	Користувач вводить логін та пароль, натискає кнопку реєстрації
Альтернативні розвитку	Користувач натискає на кнопку, що переносить на попередню сторінку.
Виняткові ситуації	Користувач вводить логін, пароль, але закриває форму; користувач нічого не вводить

Таблиця 2.3 – Сценарій використання «Перегляд треків».

Назва	Перегляд треків
ID	3
Опис	Користувач, використовуючи форму переглядає треки
Актори	Користувач, виконавець
Вигоди	Якщо користувачі не можуть переглядати треки, то вони навряд чи продовжать користуватися сервісом
Частота користування	Постійно
Тригери	Користувач переглядає треки
Передумови	На форму перегляду треків. Кнопки перегляду треків доступні у основній формі плеєра
Постумови	Користувач переглядає треки
Основний розвиток	Користувач натискає кнопки перегляду треків
Альтернативні розвитку	-
Виняткові ситуації	-

Таблиця 2.4 – Сценарій використання «Додавання треку».

Назва	Додавання треку
ID	4
Опис	Виконавець додає новий трек
Актори	Виконавець
Вигоди	Якщо виконавці не зможуть додавати свої треки, то вони навряд чи продовжать користуватися сервісом
Частота користування	Постійно
Тригери	Виконавець вводить назву, виконавця, альбом, плейлист треку у спеціальні поля
Передумови	Можливість додати трек доступна на основній формі плеєра, на яку можна перейти лише з початкової сторінки в ролі виконавця
Постумови	Виконавець додає новий трек
Основний розвиток	Виконавець вписує назву, виконавця, альбом, плейлист треку та нажимає кнопку додавання
Альтернативні розвитку	Виконавець натискає на кнопку, що переносить на попередню сторінку
Виняткові ситуації	Виконавець вписує назву, виконавця, альбом, плейлист треку та нажимає кнопку, що переносить на попередню сторінку

Таблиця 2.5 – Сценарій використання «Видалення трека».

Назва	Видалення трека
ID	5
Опис	Виконавець, використовуючи кнопку для видалення, видаляє трек
Актори	Виконавець
Вигоди	Якщо виконавці не зможуть видаляти свої треки, то вони навряд чи продовжать користуватися сервісом
Частота користування	Постійно
Тригери	Виконавець обирає трек, що хоче видалити та натискає на кнопку видалення.
Передумови	Кнопка для видалення треків доступна на основній формі плеєра, на яку можна перейти лише з сторінки авторизації в ролі виконавця
Постумови	Виконавець видаляє трек
Основний розвиток	Користувач обирає трек, що хоче видалити та натискає на кнопку видалення.
Альтернативні розвитку	-
Виняткові ситуації	-

Таблиця 2.6 – Сценарій використання «Переглянути додаткову інформацію».

Назва	Переглянути додаткову інформацію
ID	6
Опис	Користувач, використовуючи кнопку, може дізнатися більше про трек
Актори	Користувач, виконавець
Вигоди	Більший функціонал приваблює більше користувачів
Частота користування	Іноді
Тригери	Користувач дивиться додаткову інформацію
Передумови	Кнопка додаткової інформації знаходиться на основній формі плеєра, на яку можна перейти лише з сторінки авторизації
Постумови	Користувач дізнається додаткову інформацію
Основний розвиток	Користувач кнопку додаткової інформації і дивиться додаткову інформацію
Альтернативні розвитку	-
Виняткові ситуації	-

3 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1 Загальна структура проекту

Загальна структура проекту представлена на рисунку 3.1.1.

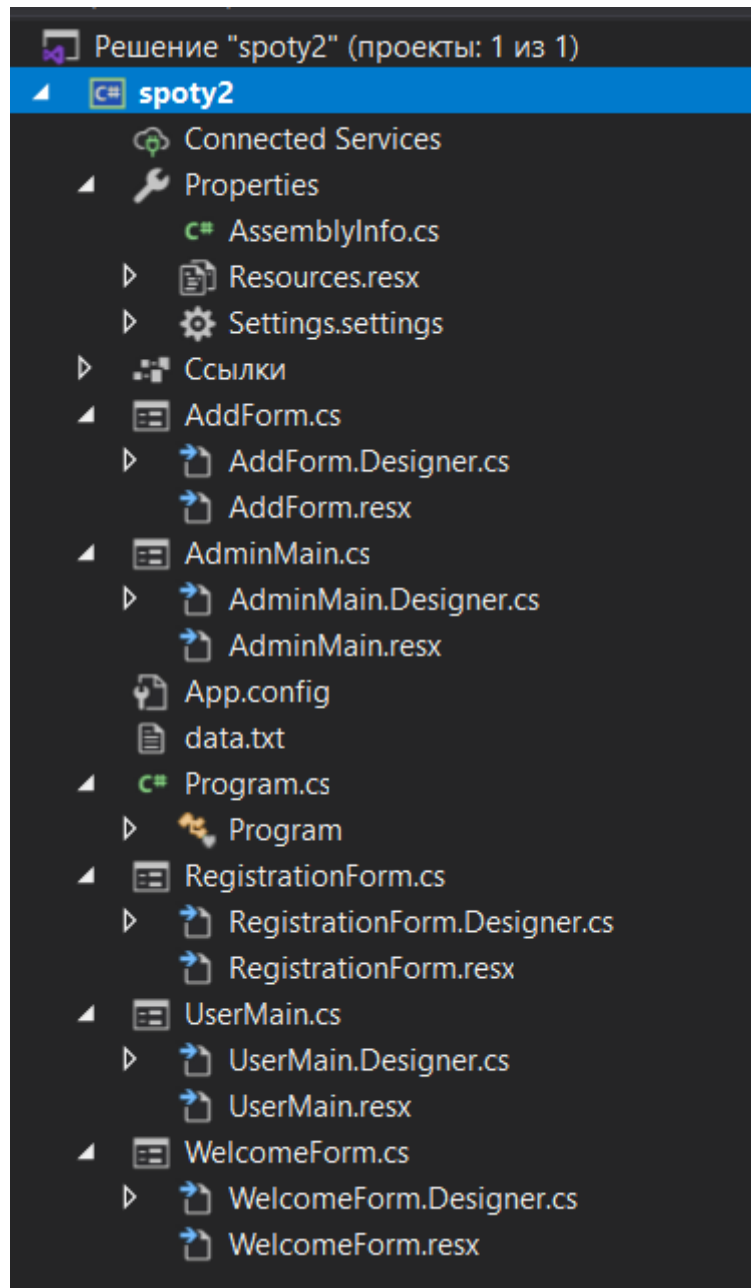


Рисунок 3.1.1 – Загальна структура проекту

3.2 Компоненти рівня доступу до даних

Основні сутності та інтерфейси рівня доступу до даних та ег-діаграма наведені на рисунку 3.2.1.

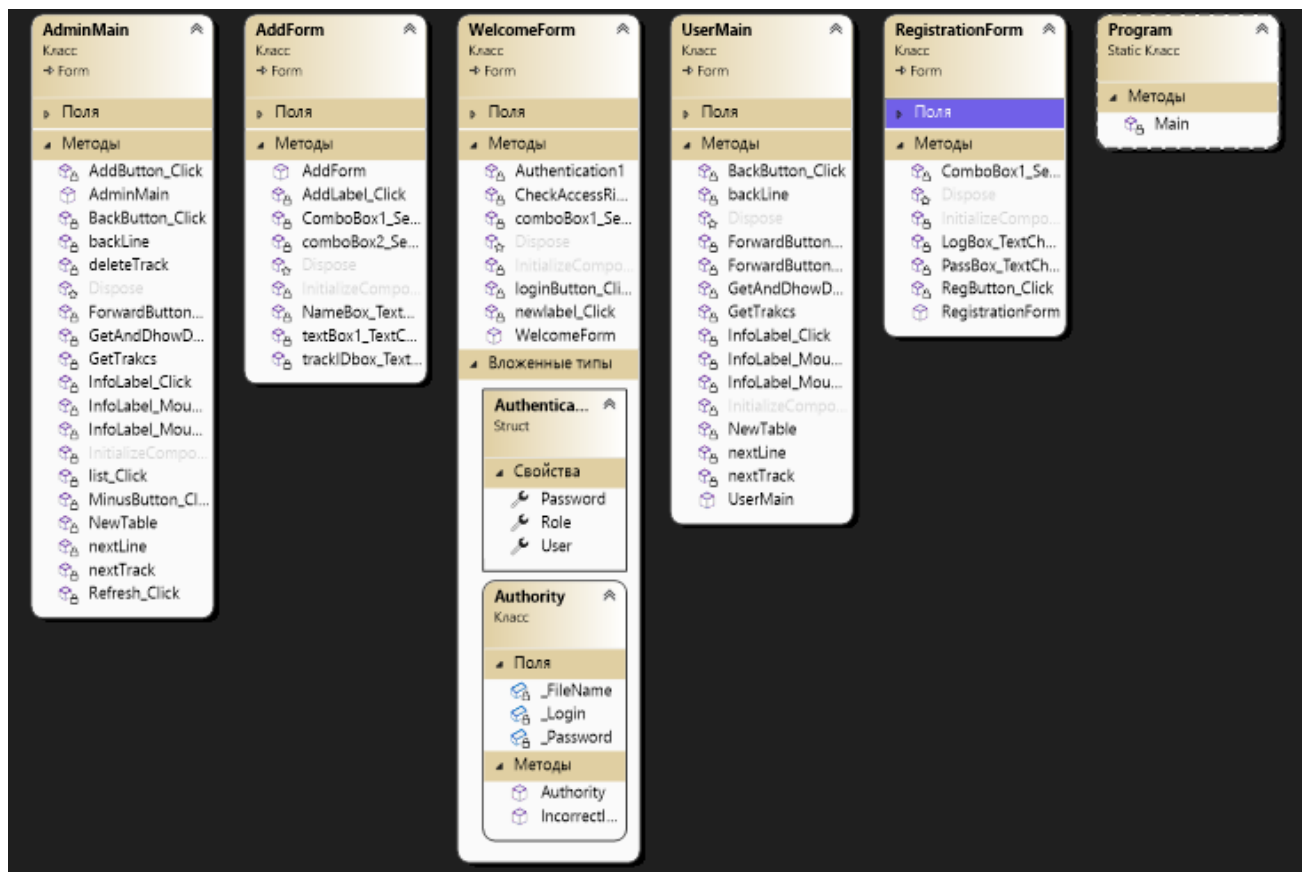


Рисунок 3.2.1 – Діаграма класів//ег-діаграма

Список класів:

1. Клас WelcomeForm – нащадок від Form, реалізує авторизацію, перехід на наступні форми (основна форма плеєра, форма додавання треку, форма реєстрації).

Методи:

- Authentication1– метод, за допомогою якого програма авторизує користувача [Додаток А.1]

- `CheckAccessRight` – метод, за допомогою якого програма визначає яка роль буде використовуватися в програмі (користувач, авиконавець). [Додаток А.2]
- `loginButton_Click` – метод, переносить користувача на основну форму плеєра. [Додаток А.3]
- `newlabel_Click` – переносить користувача на форму реєстрації. [Додаток А.4]

2. Клас `RegistrationForm` – нащадок від `Form`, реалізує створення нового облікового запису.

Методи:

- `ComboBox1_SelectedIndexChanged` – метод, що визначає роль майбутнього користувача (користувач, виконавець) . [Додаток А.5]
- `LogBox_TextChanged` – визначає логін майбутнього користувача [Додаток А.6]
- `PassBox_TextChanged` – визначає пароль майбутнього користувача [Додаток А.7]
- `RegButton_Click` - реєструє новий обліковий запис користувача [Додаток А.8]

3. Клас `AdminMain` – нащадок від `Form`, реалізує основні методи користування плеєром в ролі виконавця.

Методи:

- AddButton_Click – метод, що реалізує перехід до форми додавання нового трека. [Додаток А.9]
- MinusButton_Click – реалізує метод видалення треку. [Додаток А.10]
- ForwardButton_Click – реалізує метод переходу від вибраного трека до наступного. [Додаток А.11]
- BackButton_Click – реалізує метод переходу від вибраного трека до попереднього. [Аналогічний додатку 11]
- nextLine – реалізує методи по вибору наступного рядка. [Додаток А.12]
- backLine – реалізує методи по вибору попереднього рядка. [Аналогічний додатку А.12]
- deleteTrack – метод видалення треку [Додаток А.13]
- nextTrack – метод переходу трека (до попереднього; до наступного) [Додаток А.14]
- GetAndDhowData – метод, який робить запит в базу даних для виводу треків в таблицю. [Додаток А.15]
- GetTrakcs – метод який виводить треки в таблицю.[Додаток А.16]
- Refresh_Click – метод який оновлює таблицю. [Додаток А.17]

- InfoLabel_Click – метод який визиває додаткову інформацію про трек. [Додаток А.18]
 - NewTable – метод який створює нову таблицю, для виводу додаткової інформації про треки. [Додаток А.19]
4. Клас UserMain – нащадок від Form, реалізує основні методи користування плеєром в ролі користувача.
- ForwardButton_Click – реалізує метод переходу від вибраного трека до наступного. [Додаток А.11]
 - BackButton_Click – реалізує метод переходу від вибраного трека до попереднього. [Аналогічний додатку 11]
 - nextLine – реалізує методи по вибору наступного рядка. [Додаток А.12]
 - backLine – реалізує методи по вибору попереднього рядка. [Аналогічний додатку А.12]
 - nextTrack – метод переходу трека (до попереднього; до наступного) [Додаток А.14]
 - GetAndDhowData – метод, який робить запит в базу даних для виводу треків в таблицю. [Додаток А.15]
 - GetTrakcs – метод який виводить треки в таблицю.[Додаток А.16]
 - Refresh_Click – метод який оновлює таблицю. [Додаток А.17]

- InfoLabel_Click – метод який визиває додаткову інформацію про трек. [Додаток А.18]
- NewTable – метод який створює нову таблицю, для виводу додаткової інформації про треки. [Додаток А.19]

5. Клас AddForm – нащадок від Form, реалізує додавання трека.

Методи:

- ComboBox1_SelectedIndexChanged – реалізує вибір виконавця треку. [Додаток А.20]
- comboBox2_SelectedIndexChanged – реалізує вибір альбому. [Додаток А.21]
- NameBox_TextChanged – метод що задає назву майбутньому треку. [Додаток А.22]
- textBox1_TextChanged – метод що визначає довжину трека. [Додаток А.23]
- AddLabel_Click – метод що додає трек із заданими полями в основну форму плеєра. [Додаток А.24]

4 ІНСТРУКЦІЯ КОРИСТУВАЧА

При запуску програми користувач потрапляє на початкову сторінку.

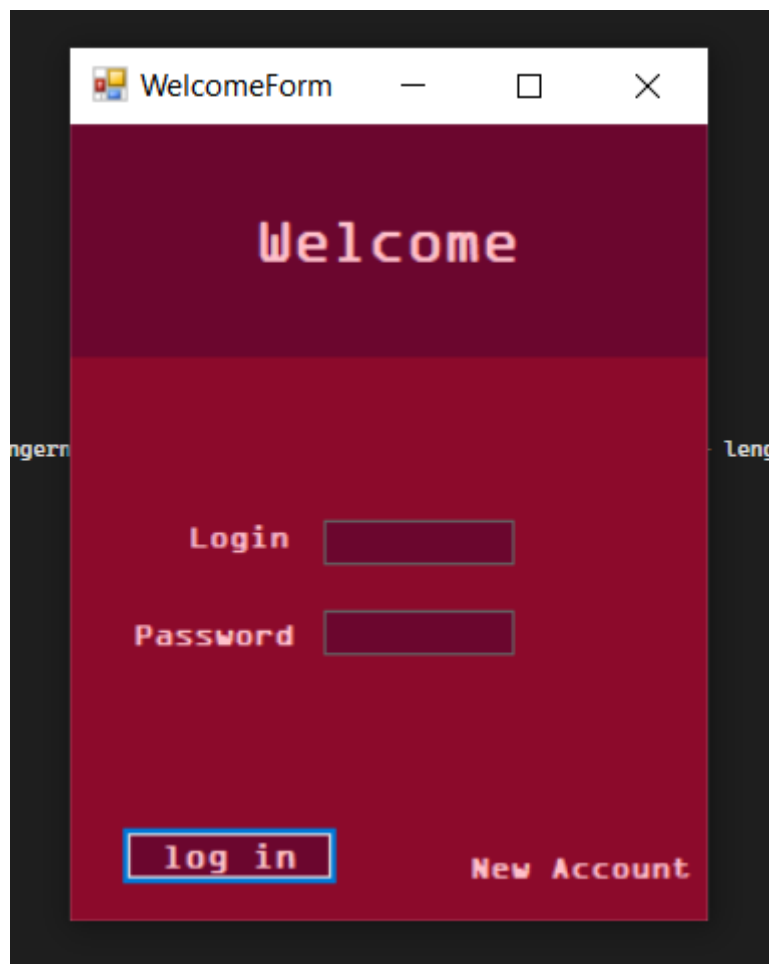


Рисунок 4.1 – Початкова сторінка

Щоб авторизуватися потрібно ввести логін та пароль. Якщо акаунт ще не створений, його можна створити натиснувши на кнопку “New Account”. Натиснувши на кнопку “log in”, ви потрапляєте у основне середовище плеєра

У формі реєстрації необхідно ввести свій логін та пароль, а також вибрати тип користувача: користувач або виконавець. При натисканні кнопки “Registration”, програма створює новий обліковий запис та повертає вас у вікно авторизації.

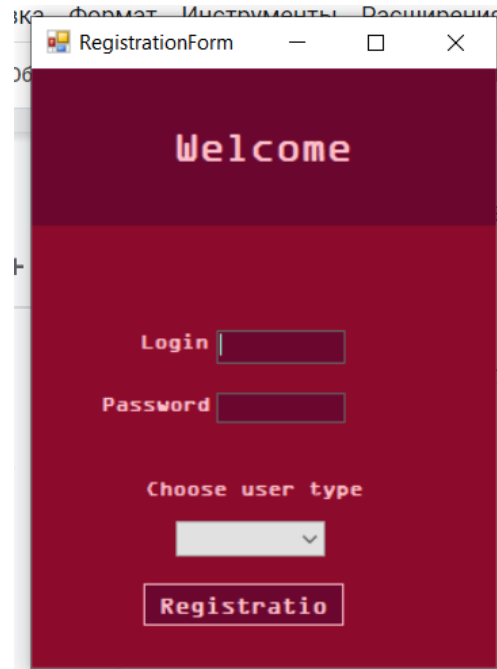


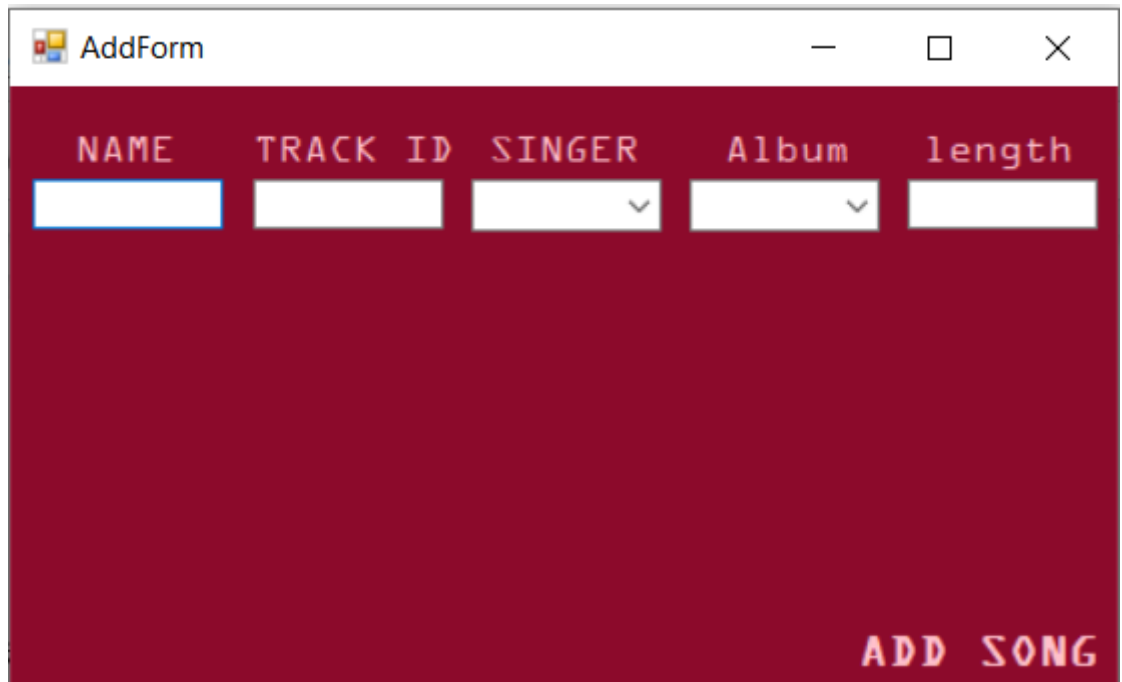
Рисунок 4.2 – Сторінка реєстрації

Після авторизації, вас вітає основна форма плеєра



Рисунок 4.3 – Основна програма

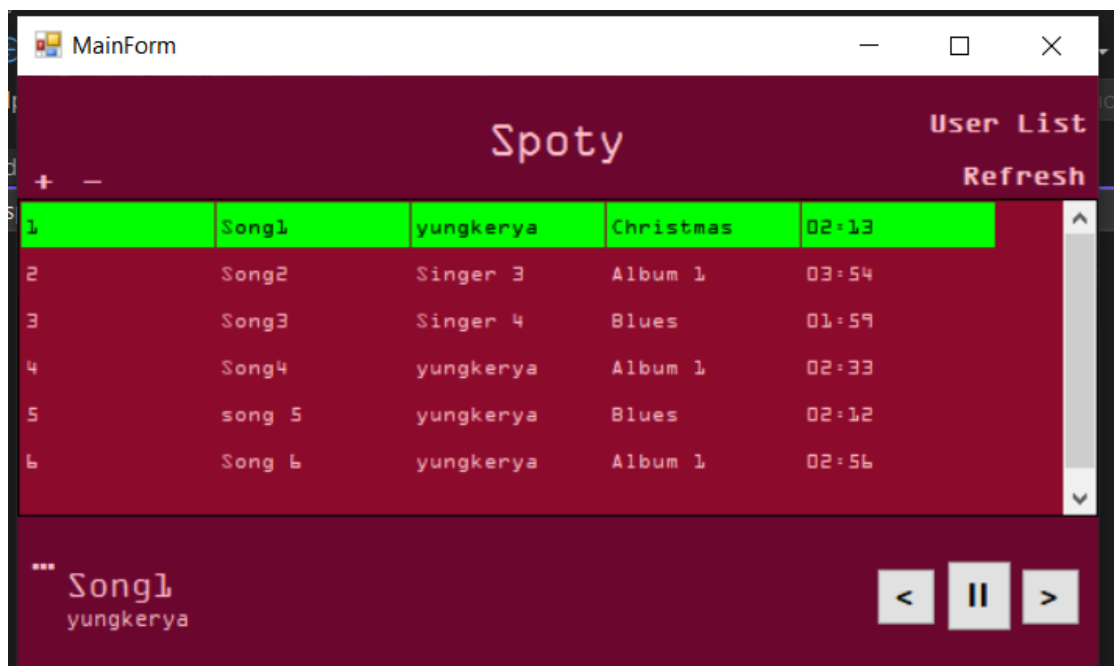
При натисканні на “+”, вас буде відправлено у форму додавання треку:



ADD SONG

Рисунок 4.4 – Форма додавання нового трека

Заповнивши всі поля, та натиснувши кнопку “ADD SONG”, ви успішно додасте новий трек у ваш плеєр.



Index	Song Name	Singer	Album	Duration
1	Song1	yungkerya	Christmas	02:13
2	Song2	Singer 3	Album 1	03:54
3	Song3	Singer 4	Blues	01:59
4	Song4	yungkerya	Album 1	02:33
5	song 5	yungkerya	Blues	02:12
6	Song 6	yungkerya	Album 1	02:56

Song1
yungkerya

Рисунок 4.5 – Доданий трек

Натиснувши кнопку “–”, ви видалите вибраний трек.

При натиску на кнопку “...” з’являється повідомлення з докладнішою інформацією про трек:

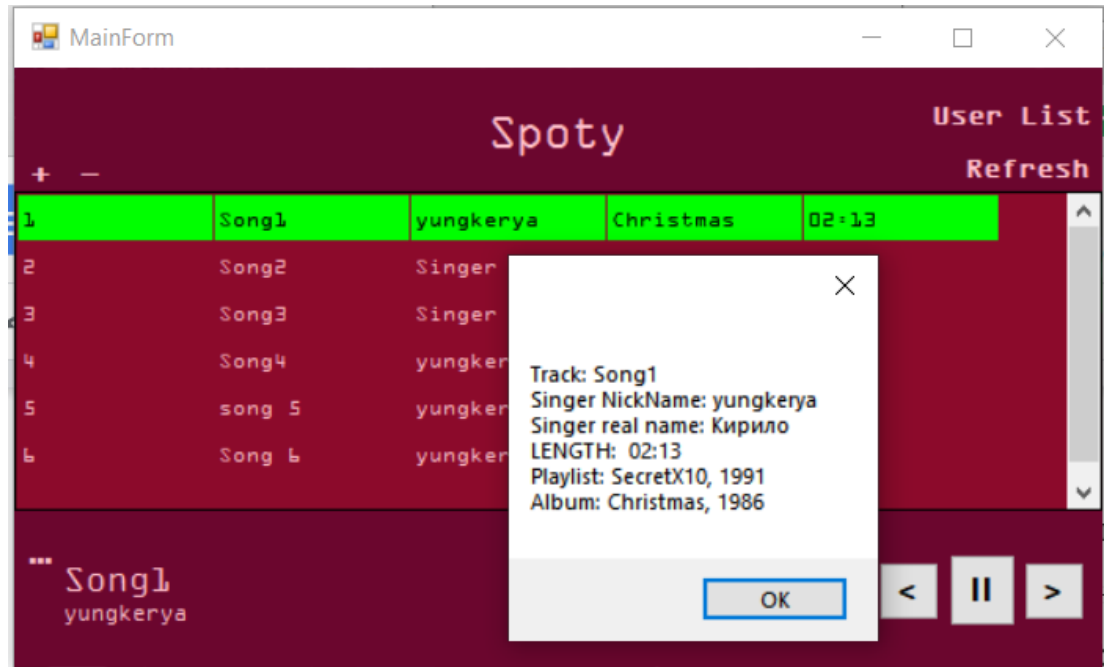


Рисунок 4.7 – додаткова інформація про трек

5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У даній програмі помилкою може бути лише неввід правильного логіна або паролю

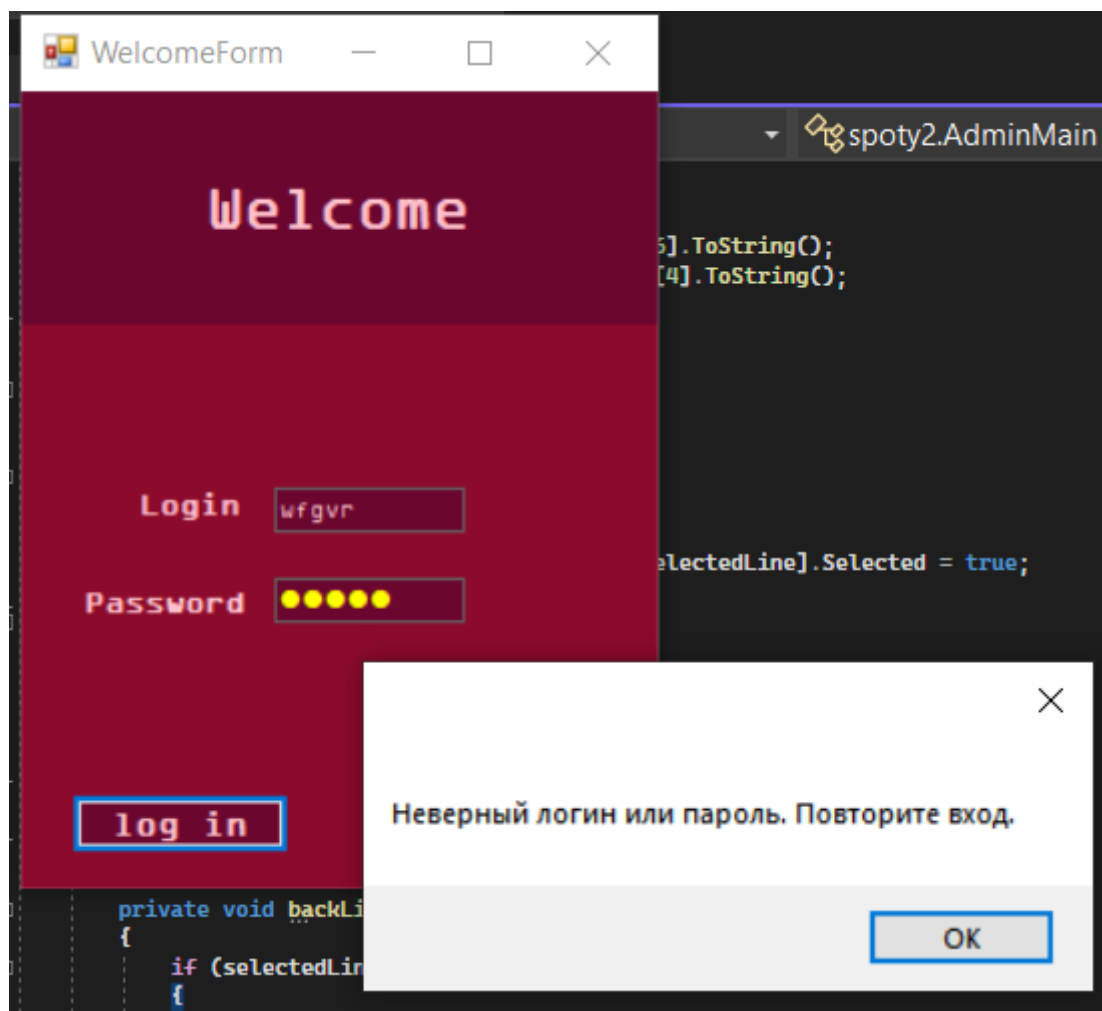


Рисунок 5.1 – Помилка про невірний логін або пароль

ВИСНОВКИ

У ході даної курсової роботи був розроблений та створений проєкт програми плеєра. Робота була виконана у Windows Forms на мові С#, для виконання застосовувались їх основні функції.

У цій роботі застосовувалось 5 форм: WelcomeForm, , RegistrationForm, UserMain, AdminMain, AddForm. Кожна з них має свою унікальну функцію та є незамінною у цій програмі.

Програма виконує такі функції: авторизація, реєстрація, перегляд треків, додавання треків, видалення треків, перегляд додаткової інформації.

Описані можливі винятки, які можуть виникнути під час використання програми та перешкоджати її правильному використанню. У цьому випадку програма шукає помилку і дозволяє користувачеві виправити введені дані. В результаті програма працює правильно, виконує весь свій функціонал.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. **Microsoft Docs** – Доступ до ресурсу:
<https://docs.microsoft.com/ru-ru/dotnet/csharp/>
2. **Metanit**– Доступ до ресурсу: <https://metanit.com/>
3. **Stackoverflow** – Доступ до ресурсу: <https://ru.stackoverflow.com/>

Посилання на код курсової роботи на GitHub:

<https://github.com/TastyRoti/spoty>

ДОДАТОК А

Ссылка: 1

```
private void Authentication1()
{
    {
        if (usernamebox.Text != "" && passbox.Text != "")
        {
            CheckAccessRight();
        }
        else
        {
            MessageBox.Show("Заповніть усі поля авторизації");
        }
    }
}
```

Рисунок А.1 – метод Authentication1 класу WelcomeForm

Ссылка: 1

```
private void CheckAccessRight()
{
    Authority NewAuthority = new Authority(FileName, usernamebox.Text, passbox.Text);

    var X = NewAuthority.IncorrectInput();

    if (X.Count() != 0)
    {
        foreach (var Row in X)
        {
            if (Row.Role == 1)
            {
                this.Hide();
                AdminMain mainForm = new AdminMain();
                mainForm.Show();
            }
            else
            {
                this.Hide();
                UserMain mainForm = new UserMain();
                mainForm.Show();
            }
        }
    }
}
```

Рисунок А.2 – метод CheckAccessRight класу WelcomeForm

```

Ссылка: 1
private void loginButton_Click(object sender, EventArgs e)
{
    Authentication1();
}

```

Рисунок А.3 – метод loginButton_Click класу WelcomeForm

```

Ссылка: 1
private void newlabel_Click(object sender, EventArgs e)
{
    RegistrationForm redform = new RegistrationForm();
    redform.Show();
}

```

Рисунок А.4 – метод newlabel_Click класу WelcomeForm

```

Ссылка: 1
private void ComboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (ComboBox1.SelectedIndex == 0)
    {
        role = 1;
    }
    else
    {
        role = 0;
    }
}

```

Рисунок А.5 – метод ComboBox1_SelectedIndexChanged класу RegistrationForm

```

Ссылка: 1
private void LogBox_TextChanged(object sender, EventArgs e)
{
    login = LogBox.Text;
}

```

Рисунок А.6 – метод LogBox_TextChanged класу RegistrationForm

```

Ссылка: 1
private void PassBox_TextChanged(object sender, EventArgs e)
{
    password = PassBox.Text;
}

```

Рисунок А.7 – метод PassBox_TextChanged класу RegistrationForm

```

Ссылка: 1
private void RegButton_Click(object sender, EventArgs e)
{
    res = login + ";" + password + ";" + role + Environment.NewLine;
    File.AppendAllText("data.txt", res);
    this.Hide();
}

```

Рисунок А.8 – метод RegButton_Click класу RegistrationForm

```

Ссылка: 1
private void AddButton_Click(object sender, EventArgs k)
{
    AddForm addForm = new AddForm();
    addForm.Show();
}

```

Рисунок А.9 – метод AddButton_Click класу AdminMain

```

Ссылка: 1
private void MinusButton_Click(object sender, EventArgs e)
{
    deleteTrack();
}

```

Рисунок А.10 – метод MinusButton_Click класу AdminMain

```

Ссылка: 1
private void ForwardButton_Click(object sender, EventArgs e)
{
    if (index >= LenTable - 1)
    {
    }
    else
    {
        index++;
        nextTrack();
    }

    nextLine();
    NewTable();
}

```

Рисунок А.11 – метод ForwardButton_Click класу AdminMain

```

Ссылка: 1
private void nextLine()
{
    if (selectedLine == LenTable - 1 )
    {
        tracks.ClearSelection();
        tracks.Rows[selectedLine].Cells[selectedLine].Selected = true;
    }
    else
    {
        selectedLine++;
        tracks.ClearSelection();
        tracks.Rows[selectedLine].Cells[selectedLine].Selected = true;
    }
}

```

Рисунок А.12 – метод nextLine класу AdminMain

```

Ссылка: 1
private void deleteTrack()
{
    foreach (DataGridViewRow row in tracks.SelectedRows)
    {
        tracks.Rows.RemoveAt(row.Index);

        sql = "DELETE FROM track WHERE track_id = 6;";

        try
        {
            connection = new SqlConnection(connectionString);
            connection.Open();
            com = new SqlCommand(sql, connection);
            data = new SqlDataAdapter(com);
            dT = new DataTable();
            data.Fill(dT);
            LenTable = dT.Rows.Count;
            connection.Close();
        }
        catch (Exception k)
        {
            MessageBox.Show(k.Message);
        }
    }
}

```

Рисунок А.13 – метод deleteTrack класу AdminMain

```

Ссылка: 3
private void nextTrack()
{
    NewTable();
    TrackNameLabel.Text = dT.Rows[index][6].ToString();
    SingerNameLabel.Text = dT.Rows[index][4].ToString();
}

```

Рисунок А.14 – метод nextTrack класу AdminMain


```

Ссылка: 1
private void GetAndDhowData(string SQLQuery, DataGridView dataGrid)
{
    connection = new SqlConnection(connectionString);
    connection.Open();
    com = new SqlCommand(SQLQuery, connection);
    data = new SqlDataAdapter(com);
    dT = new DataTable();
    data.Fill(dT);
    dataGrid.DataSource = dT.DefaultView;
    LenTable = dT.Rows.Count;
    connection.Close();
}

```

Рисунок А.15 – метод GetAndDhowData класу AdminMain

```

Ссылка: 2
private void GetTrakcs()
{
    sql = " SELECT dbo.track.track_id, dbo.track.track_name, dbo.singer.singer_name, dbo.album.album_name, dbo.track.length"
        + " FROM      dbo.album INNER JOIN"
        + " dbo.track ON dbo.album.album_id = dbo.track.album_id INNER JOIN"
        + " dbo.playlist ON dbo.track.playlist_id = dbo.playlist.playlist_id INNER JOIN"
        + " dbo.singer ON dbo.track.singer_id = dbo.singer.singer_id";

    try
    {
        GetAndDhowData(sql, tracks);
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message);
    }

    tracks.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
}

```

Рисунок А.16 – метод GetTrakcs класу AdminMain

```

Ссылка: 1
private void Refresh_Click(object sender, EventArgs e)
{
    tracks.DataSource = null;
    tracks.Update();
    tracks.Refresh();
    GetTrakcs();
}

```

Рисунок А.17 – метод Refresh_Click класу AdminMain

```

Ссылка: 1
private void InfoLabel_Click(object sender, EventArgs e)
{
    MessageBox.Show("Track: " + dT.Rows[index][6].ToString()
        + "\nSinger NickName: " + dT.Rows[index][4].ToString()
        + "\nSinger real name: " + dT.Rows[index][5].ToString()
        + "\nLENGTH: " + dT.Rows[index][8].ToString()
        + "\nPlaylist: " + dT.Rows[index][2].ToString() + ", " + dT.Rows[index][3].ToString()
        + "\nAlbum: " + dT.Rows[index][1].ToString() + ", " + dT.Rows[index][0].ToString());
}

```

Рисунок А.18 – метод InfoLabel_Click класу AdminMain

```

Ссылка: 4
private void NewTable()
{
    sql = " SELECT dbo.album.album_year, dbo.album.album_name, dbo.playlist.playlist_name, dbo.playlist.playlist_year, dbo.singer.singer_name, dbo.singer.singer_id FROM dbo.album INNER JOIN"
        + " dbo.track ON dbo.album.album_id = dbo.track.album_id INNER JOIN"
        + " dbo.playlist ON dbo.track.playlist_id = dbo.playlist.playlist_id INNER JOIN"
        + " dbo.singer ON dbo.track.singer_id = dbo.singer.singer_id";

    try
    {
        connection = new SqlConnection(connectionString);
        connection.Open();
        com = new SqlCommand(sql, connection);
        data = new SqlDataAdapter(com);
        dT = new DataTable();
        data.Fill(dT);
        LenTable = dT.Rows.Count;
        connection.Close();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message);
    }

    tracks.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
}

```

Рисунок А.19 – метод NewTable класу AdminMain

```

Ссылка: 1
private void ComboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    switch (ComboBox1.SelectedIndex)
    {
        case 0:
            singername = 3;
            break;
        case 1:
            singername = 4;
            break;
        case 2:
            singername = 1;
            break;
        case 3:
            singername = 2;
            break;
    }
}

```

Рисунок А.20 – метод ComboBox1_SelectedIndexChanged класу AddForm.cs

```

Ссылка: 1
private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    switch (comboBox2.SelectedIndex)
    {
        case 0:
            albumname = 1;
            break;
        case 1:
            albumname = 2;
            break;
        case 2:
            albumname = 3;
            break;
    }
}

```

Рисунок А.21 – метод ComboBox2_SelectedIndexChanged класу AddForm.cs

```

Ссылка: 1
private void NameBox_TextChanged(object sender, EventArgs e)
{
    track_name = NameBox.Text;
}

```

Рисунок А.22 – метод NameBox_TextChanged класу AddForm.cs

```

Ссылка: 1
private void textBox1_TextChanged(object sender, EventArgs e)
{
    length = textBox1.Text;
}

```

Рисунок А.23 – метод textBox1_TextChanged класу AddForm.cs

```

COMMAND: 1
private void AddLabel_Click(object sender, EventArgs k)
{

    this.Hide();

    sql = "insert into track values('" + track_id + "', '" + track_name + "', '" + singername + "', '" + albumname + "', '1', '2000', '" + length + "')";

    try
    {
        connection = new SqlConnection(connectionString);
        connection.Open();
        com = new SqlCommand(sql, connection);
        data = new SqlDataAdapter(com);
        dt = new DataTable();
        data.Fill(dt);
        LenTable = dt.Rows.Count;
        connection.Close();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message);
    }
}

```

Рисунок А.24 – метод AddLabel_Click класу AddForm.cs