

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**на тему: «Інформаційна система підтримки проведення змагань з
бальних танців»**

за напрямом підготовки 6.050101 «Комп'ютерні науки»

Виконавець роботи: студент групи ІТ-51 Жук Олена Костянтинівна

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ « » _____ 2019 р.

Науковий керівник

_____ к. т. н., доц., Марченко А. В.
(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

_____ Шифрін Д. М.
(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2019

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки та інформаційні технології»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик

« ____ » _____ 2019 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Жук Олена Костянтинівна

1 Тема роботи «Інформаційна система підтримки проведення змагань з бальних танців»

керівник роботи Марченко Анна Вікторівна, к. т. н., доцент _____,

затверджені наказом по університету від « 17 » травня 2019 р. № 0834-III

2 Строк подання студентом роботи « » _____ 2019 р.

3 Вхідні дані до роботи технічне завдання на розробку інформаційної системи, матеріали

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, постановка задачі та методи дослідження, моделювання інформаційної системи, розробка інформаційної системи підтримки проведення змагань зі спортивних бальних танців

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність, мета і задачі, огляд інформаційних систем підтримки проведення змагань зі спортивних бальних танців, порівняння знайдених ІС, функціональні вимоги, схема бази даних, контекстна діаграма, діаграма декомпозиції, діаграма варіантів використання, перелік програмних засобів реалізації, приклади реалізації, акт впровадження, висновки

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Ідентифікація мети ІС	04.09.18-07.09.18	
2	Пошук аналогів	10.09.18-17.09.18	
3	Визначення вимог	18.09.18-05.10.18	
4	Функціональні вимоги	08.10.18-16.10.18	
5	Нефункціональні вимоги	17.10.18-25.10.18	
6	Визначення засобів для реалізації	26.10.18-02.11.18	
7	Планування структури WBS	05.11.18-13.11.18	
8	Планування структури OBS	14.11.18-22.11.18	
9	Створення календарного плану	23.11.18-11.12.18	
10	Розрахунок фінансів	12.12.18-28.12.18	
11	Виявлення ризиків	31.12.18-16.01.19	
12	Налаштування середовища розробки	17.01.19-30.01.19	
13	Розробка БД	31.01.19-20.02.19	
14	Проектування інтерфейсу ІС	21.02.19-20.03.19	
15	Реалізація інтерфейсу доступу до ІС	21.03.19-24.04.19	
16	Тестування методом чорного ящика	25.04.19-06.06.19	
17	Виправлення виявлених помилок	07.05.19-13.05.19	
18	Розробка пояснювальної записки	14.05.19-24.05.19	
19	Створення настанови з використання	27.05.19-31.05.19	
20	Архівація	03.06.19-16.06.19	

Студент

(підпис)

Жук О. К.

Керівник роботи

(підпис)

к. т. н. доц. Марченко А. В.

РЕФЕРАТ

Тема бакалаврської роботи: «Інформаційна система підтримки змагань з бальних танців».

Пояснювальна записка містить вступ, 4 розділи, висновки, додатки та список літератури, включає 112 сторінок, 7 таблиць, 57 ілюстрацій, 20 джерел.

Перший розділ описує виконаний аналіз предметної області та систем, схожих на розроблювану, завдяки чому чітко зрозуміло актуальність розробки інформаційної системи(ІС). Також розділ визначає головні аспекти функціонування ІС.

Другий розділ включає в себе визначення головної мети дипломної роботи, основних задач та функціоналу. Крім того, наведено вибір методів та засобів реалізації ІС.

Третій розділ визначає опис функціонального моделювання процесу реалізації ІС, моделювання бази даних та варіантів використання. Також описані основні принципи функціонування ІС.

У четвертому розділі наведено розробку інформаційної системи, яка є результатом використання обраних раніше методів та засобів реалізації. Детально описуються етапи розробки та результати реалізації ІС.

У результаті розробки отримано ІС підтримки проведення змагань зі спортивних бальних танців, яка забезпечує автоматизовану підтримку необхідних бізнес-процесів з організації змагань зі спортивних бальних танців.

Ключові слова: інформаційна система, змагання, спортивні бальні танці, КСТ Силует, Скейтинг.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Дослідження актуальності проблеми	7
1.2 Аналіз систем підтримки проведення танцювальних змагань та визначення проблеми.....	10
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ	15
2.1 Мета та задачі.....	15
2.2 Вибір засобів реалізації.....	16
3 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	18
3.1 Структурно-функціональне моделювання процесу інформаційної підтримки проведення змагань зі спортивних бальних танців	18
3.2 Моделювання варіантів використання інформаційної системи підтримки проведення змагань зі спортивних бальних танців	21
3.3 Проектування бази даних	23
4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ПРОВЕДЕННЯ ЗМАГАНЬ ЗІ СПОРТИВНИХ БАЛЬНИХ ТАНЦІВ	25
4.1 Установка та запуск компонентів ІС	25
4.2 Результат реалізації інформаційної системи	29
ВИСНОВОК.....	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47
Додаток А. Технічне завдання	49
Додаток Б. Планування робіт.....	53
Додаток В. Лістинг коду.....	61
Додаток Г. Настанова з використання	109
Додаток Д. Копії акту впровадження ІС.....	112

ВСТУП

Сучасний рівень розвитку інформаційних технологій може полегшувати організацію та проведення будь-яких змагань за допомогою інформаційних систем (ІС) через автоматизацію виконання бізнес-процесів. Значимою є розроблення такої системи і для підтримки проведення змагань зі спортивних бальних танців.

Суддівство в бальних танцях відрізняється від суддівства в інших видах спорту відсутністю точної оцінки за правильне виконання тих чи інших елементів. До завдання судді входить оцінка виконавської майстерності між танцювальними парами шляхом порівняння. Тому дуже важливо на змаганнях мати таку інформаційну систему, що допомагає отримати результати танцювальних пар з оцінювання суддями.

«Скейтинг» система застосовується не тільки в спортивних бальних танцях, однак, саме в цій області вона отримала широке розповсюдження. На сьогоднішній день існують спеціалізовані програми, які проводять підрахунок і розподіл місць в автоматичному режимі.

Застосування такої системи оцінювання танцювальних пар надає більшої об'єктивності такому індивідуальному у суддівстві виду мистецтва та спорту, як бальний танець.

Інформаційна система підтримки змагань зі спортивних бальних танців містить в собі не тільки систему підрахунку результатів змагань, а також реєстрацію танцювальних пар на змагання та можливість друку результатів.

Дана ІС створена для її використання робітниками КСТ Силует у м. Суми. У результаті розробки отримано ІС, яка забезпечує автоматизовану підтримку необхідних бізнес-процесів з організації змагань зі спортивних бальних танців.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження актуальності проблеми

У сучасному світі найбільш популярним напрямом танцювального спорту є спортивні бальні танці. Даний вид танцювальної галузі визнається Міжнародним олімпійським комітетом як кандидат в олімпійські види спорту, але ймовірність включення до Олімпіади дуже мала, бо почастишало скорочення видів олімпійського спорту.

Спортивні бальні танці розділяються на дві програми: європейську та латиноамериканську. Кожна програма має п'ять танців, тобто у сумі 10 танців.

Європейська програма:

- Вальс(Waltz);
- Танго(Tango);
- Віденський вальс(Viennese Waltz);
- Фокстрот(Foxtrot);
- Квікстеп(Quickstep).

Латиноамериканська програма:

- Ча-ча-ча(Cha-cha-cha);
- Самба (Samba);
- Румба(Rumba);
- Пасодобль(Pasodoble);
- Джайв(Jive).

Для того, щоб створити більш менш рівномірну конкуренцію на танцювальному майданчику, в спортивних бальних танцях введено систему класів та вікових категорій. Система класів відображає рівень підготовки танців та система вікових категорій розподіляє танцювальні пари за віковими групами. У кожній категорії та класі є правила по фігурам та костюмам, які

оновлюються кожного року. Тому дуже важливо для танцювальної пари стежити за змінами та дотримуватися цих правил аби уникнути ризику бути дискваліфікованим зі змагання. Чим вище клас, тим більше танців та фігур можна виконувати на змаганнях. Також зі зміною у віковій категорії змінюється і зовнішній вигляд пари.

Класифікація танцювальних пар за віковими групами(категоріями):

- Діти 1 – 4-5 років;
- Діти 2 – 6-7 років;
- Ювенали 1 – 8-9 років;
- Ювенали 2 – 10-11 років;
- Юніори 1 – 12-13 років;
- Юніори 2 – 14-15 років;
- Молодь 1 – 16-18 років;
- Молодь 2 – 19-20 років;
- Дорослі – 21-35 років.

Класифікація танцювальних пар за рівнем підготовки(класами):

- Дебют – учасниками даного класу є діти, що танцюють 1-2 роки. Складається з 2 танців: вальс та ча-ча-ча;
- Школа – до 2 танців категорії Дебют додається джайв;
- N – початковий клас з 4 танцями: вальс, квікстеп, ча-ча-ча, джайв;
- E – в даному класі виконується 6 танців: вальс, танго, квікстеп, ча-ча-ча, самба та джайв;
- D – до танців класу E додаються ще 2 танця: віденський вальс та румба;
- C – клас, до якого входять усі 10 танців;
- RS – клас для рейтингових пар класу C;
- Open – відкритий клас.

Змагання зі спортивних бальних танців проводяться починаючи з 1907 року. Танцювальні пари оцінювалися за певними правилами та критеріями та

з розвитком танцювальної галузі, а саме спортивних бальних танців, ці правила з часом змінювались. Існує багато критеріїв оцінки танцювальної майстерності, тому суддівство є суб'єктивним оцінюванням на змаганнях.

З 1947 року було впроваджено систему «Скейтинг» для отримання результатів змагань. В основі даної системи лежить пошук судьями кращої танцювальної пари, навіть якщо є розбіжності в оцінках. Система «Скейтинг» складається з 11 правил:

- правило 1: встановлює порядок виставлення оцінок в суддівських протоколах в попередніх турах;
- правила 2-4: порядок виставлення оцінок в суддівських протоколах в фіналі;
- правила 5-8: порядок обробки оцінок суддів, які оцінюють виконавську майстерність, для визначення лічильною комісією результатів, отриманих парами в окремих танцях;
- правила 9-11: порядок обробки результатів виступу по всіх танцях для визначення лічильною комісією підсумкових місць, зайнятих парами в даному змаганні [1].

Кожне правило застосовується до конкретного кроку в процесі оцінювання та подальшого табличного виведення результатів. Ці одинадцять правил описують лише загальне значення кожного з критеріїв. У фінальному турі кожної категорії турніру суддя має поставити місця від першого до останнього, в залежності, скільки пар танцює у фіналі, наприклад одне перше місце, одне друге і так далі. Отже, до завдання суддівської колегії входить не аналіз якості танцю, а порівняння танцювальних пар між собою.

Наприкінці змагання кожен, хто бажає дізнатися, як та чи інша танцювальна пара була оцінена судьями, може переглянути протоколи категорій, в яких приймала участь дана пара. Завдяки цим звітам можна дізнатися на якому рівні в залежності від інших знаходиться ця пара.

В основі роботи розробленої інформаційної системи підтримки змагань зі спортивних бальних танців лежить реєстрація танцювальних пар, оцінювання їхньої майстерності судьями та отримання результатів змагань.

Отримана інформаційна система дозволяє користувачу оцінювати танцювальні пари або проводити адміністрування турніру. У режимі адміністратора по закінченню тієї чи іншої процедури можна роздрукувати протоколи результатів.

1.2 Аналіз систем підтримки проведення танцювальних змагань та визначення проблеми

Перед початком розробки інформаційної системи (ІС) був проведений аналіз існуючих програм, що використовуються для підрахунку результатів змагань зі спортивних бальних танців. Дана галузь заробітку дуже сильно розвинена та багато танцювальних клубів запрошують лічильні комісії з інших міст аби автоматизувати необхідні бізнес-процеси на змаганнях.

У результаті пошуку програм, що використовує «Скейтинг» систему, в мережі Інтернет було знайдено тільки одну, яка використовується в Україні і знаходиться у відкритому доступі для ознайомлення. Усі інші знайдені ІС застосовуються в інших країнах Далі наведено короткий опис кожної з програм, за якими буде проведено аналіз.

1. Програма ведення змагань «Турнир»[2]

Дана програма це повноцінний інструмент, за допомогою можна вести змагання, друкувати грамоти і видавати протоколи результатів змагань. Існує декілька конфігурацій даної програми: стандартна, професійна та мережева.

Далі наведено скріншот з оглядового відео про програму (рис. 1.1), тому що наведених скріншотів прикладів програми у відкритому доступі, а також демо версії програми для ознайомлення немає.

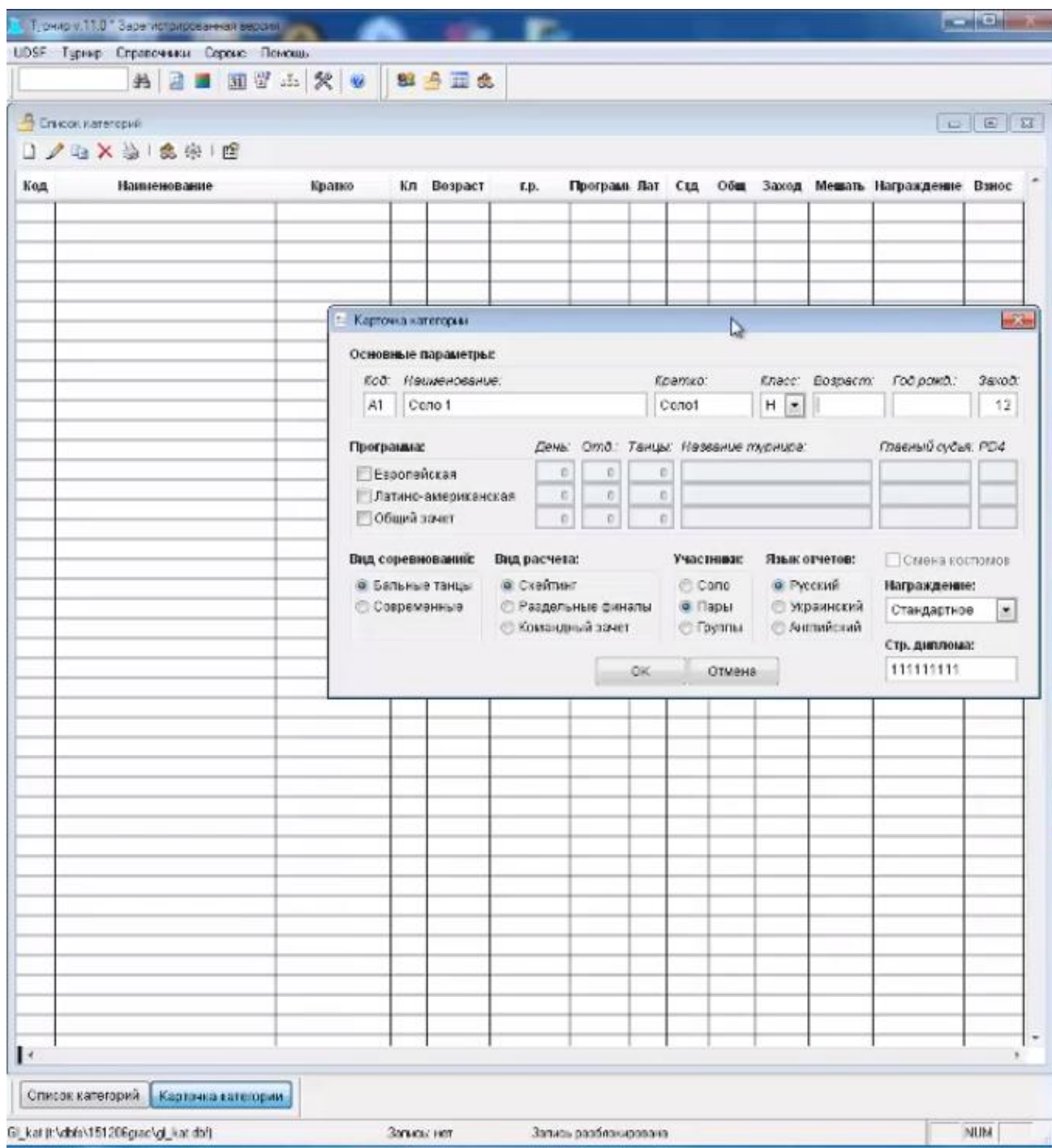


Рисунок 1.1 – Программа «Турнир»

2. Программа Skating System 5 [3]

На сьогоднішній день програма Skating System 5 є однією з найпоширеніших програм по обробці результатів на змаганнях з танцювального спорту на території Російської Федерації.

Інтерфейс програми надає швидкий доступ до будь-якої активної групи на турнірі. У головній формі програми (рис. 1.2) був застосований принцип закладок, до яких також можна швидко отримати доступ. Кожна закладка відповідає за певний функціонал програми: реєстрація, складання регламенту, перегляд результатів, введення суддівських оцінок, адміністрування суддів і т.д.

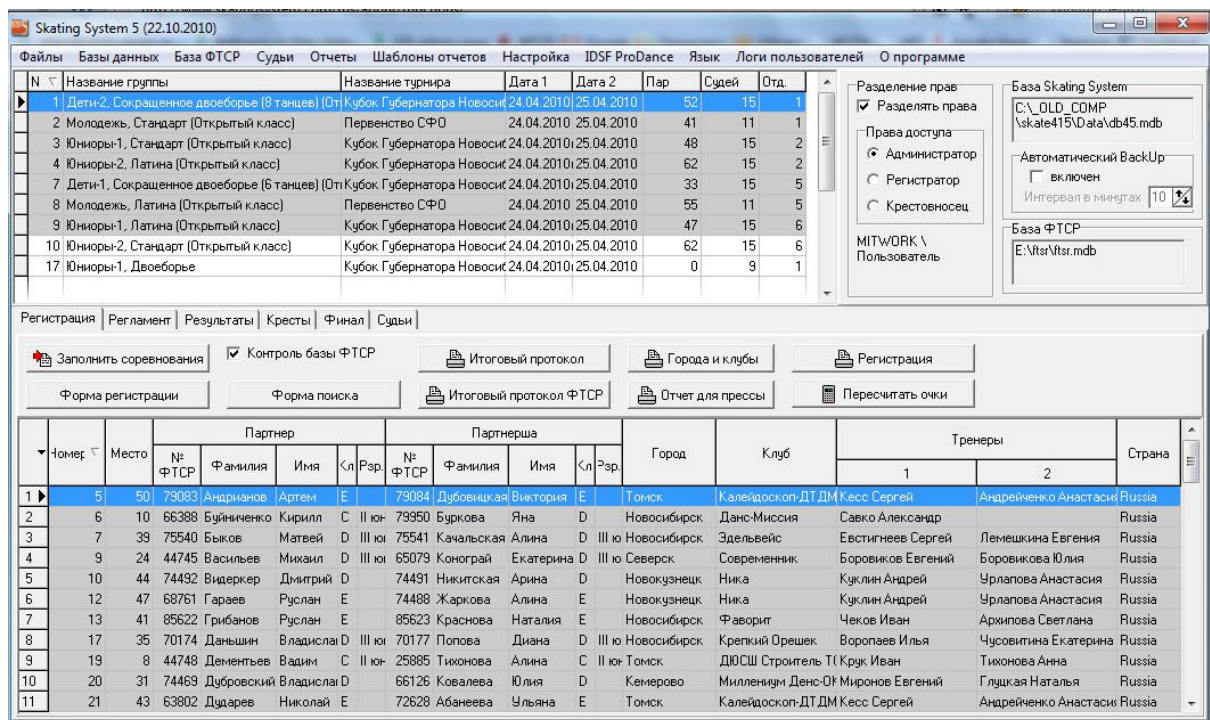


Рисунок 1.2 – Головна форма програми Skating System 5

3. Skating System Software [4]

Дане програмне забезпечення (рис. 1.3) було розроблено для управління змаганнями з танців. Використовується в англомовних країнах.

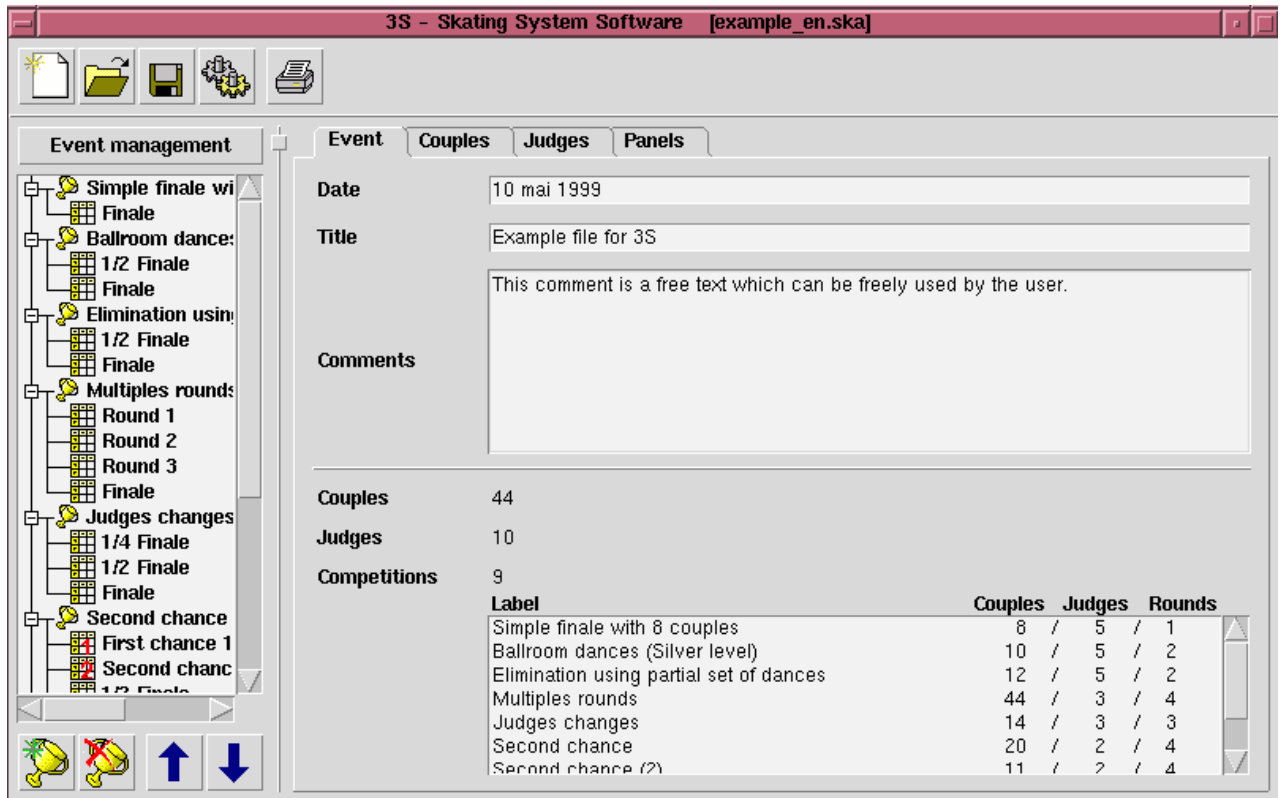


Рисунок 1.3 – Головне вікно програми

Для використання описаних вище програм потрібно запрошувати на свої танцювальні турніри спеціальну лічильну комісію, що має доступ до повної версії ІС для належного проведення змагань зі спортивних бальних танців. Робота комісії не є безкоштовною, тобто організатори змагань мають виділити кошти для її оплати. Так як підтримка проведення змагань за допомогою лічильної комісії є найважливішим фактором, щоб танцювальний турнір пройшов без затримок та автоматизовано, тому бюджет виділяється чималий. Та, якщо комісія приїжджає з іншого міста, організаторам також потрібно покрити витрати за проїзд.

Після пошуку схожих систем на розроблювану ІС було проведено порівняльний аналіз за декількома критеріями, який було занесено у таблицю. Якщо критерій присутній в ІС, то ставиться «1», якщо ні – «0». Результати можна переглянути в таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз інформаційних систем

Критерії	«Турнир»	Skating System 5	Skating System Software	Власна ІС
Зручний інтерфейс	1	0	1	1
Авторизація адміністратора	0	0	0	1
Авторизація суддів	0	0	0	1
Реєстрація учасників	1	1	1	1
Друк списків категорій	0	1	0	1
Доступність за ціною	0	0	0	1
Підсумок	2	2	2	6

Результат проведеного аналізу показав, що оптимальним рішенням є розроблення власної інформаційної системи підтримки проведення змагань зі спортивних бальних танців, власником та користувачем якої буде КСТ Силует, що знаходиться у місті Суми. Завдяки даному рішенням будуть збережені кошти, які у подальшому можна буде використати на організацію змагань, наприклад, для прикраси танцювального майданчику.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі

Головною метою проекту є розроблення інформаційної системи (ІС) підтримки змагань зі спортивних бальних танців для автоматизації реєстрації танцювальних пар, оцінювання їхніх танцювальних навичок та отримання результатів змагань у зручній формі з можливістю друку звітної документації.

Були визначені такі задачі для досягнення мети: аналіз вимог та інформаційних систем підтримки проведення змагань зі спортивних бальних танців, схожих на розроблювану ІС, вибір методів та засобів реалізації, розробка бази даних (БД), проектування і моделювання інтерфейсу, програмна реалізація модулів системи і тестування роботи отриманої ІС методом чорного ящика.

Дуже важливим кроком в розробленні інформаційної системи є підключення декількох ноутбуків, якими будуть користуватися судді, до ноутбуку адміністратора(головного). Тобто система має бути розроблена за клієнт серверною технологією.

Використання розроблюваної інформаційної системи спрямовано на автоматизацію виконуваних бізнес-процесів при організації змагань зі спортивних бальних танців, що усуне можливі затримки та незручності при використанні друкованих бюлетенів для прорахунку результатів.

Для детального опису цілей та задач було сформоване та затверджене технічне завдання, яке знаходиться у додатку А.

2.2 Вибір засобів реалізації

Перед початком реалізації програмної частини потрібно обрати засоби реалізації інформаційної системи, адже існує багато мов програмування, СКБД, середовищ розробки. На основі поставлених мети та задач, описаних вище, оптимальними засобами реалізації є:

- мова програмування C# [5][6];
- середовище розробки Microsoft Visual Studio 2010 Ultimate [7][8];
- локальний веб-сервер Open Server Panel [9];
- веб-додаток phpMyAdmin [10];
- база даних MySQL [11].

1. Мова програмування C#

C# — об'єктно-орієнтована мова програмування з безпечною типізацією для платформи .NET [5]. Дана мова є однією з найпопулярніших мов програмування та відноситься до широко відомому сімейству мов C.

C# відрізняється дуже багатим синтаксисом, який в той же час є простим і зручним. Особливо легко дається вивчення програмістам, які вже володіють знаннями і навичками роботи з мовами C, C++ і Java. C# поєднує в собі всі переваги цих мов, при цьому не має їх недоліків. Однією з основних переваг мови є наявність потужних функцій, і надання прямого доступу до операційної пам'яті[6].

2. Середовище розробки Microsoft Visual Studio 2010 Ultimate

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які містять в собі інтегроване середовище для розроблення програмного забезпечення та багато інших інструментальних засобів[7].

Microsoft Visual Studio 2010 Ultimate забезпечує інтегровану середу інструментів і серверну інфраструктуру, яка спроектує весь процес розробки

додатків. Доставка бізнес-результатів, використовуючи продуктивні, предсказуемые, настільні процеси та підвищує прозорість і простежується на протязі всього життєвого циклу. р-н Microsoft Visual Studio 2010 Ultimate - ефективний, потужний і розумний інструмент, який допоможе вам легко створити нові рішення або підсилювати існуючі додатки, розкривши свій творчий потенціал з потужною прототипуванням[8].

3. Локальний веб-сервер Open Server Panel

Open Server Panel — це портативна серверна платформа і програмне середовище, створена спеціально для веб-розробників з урахуванням їх рекомендацій і побажань. Програмний комплекс має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, має потужні можливості з адміністрування та налаштування компонентів. Платформа широко використовується з метою розробки, налагодження і тестування веб-проектів, а так само для надання веб-сервісів в локальних мережах [9].

4. Веб-додаток phpMyAdmin;

phpMyAdmin є вільним програмним засобом, написаним на PHP, призначений для керування адміністрацією MySQL через Інтернет. phpMyAdmin підтримує широкий спектр операцій на MySQL і MariaDB. Часто використовувані операції (керування базами даних, таблицями, стовпцями, відносинами, індексами, користувачами, дозволами тощо) можна виконувати за допомогою інтерфейсу користувача, в той час як ви все ще маєте можливість безпосередньо виконувати будь-який запит SQL. [10].

5. База даних MySQL

MySQL - найпопулярніша база даних у світі з відкритим кодом. Незалежно від того, чи є ви швидко зростаючою веб-власністю, технологією ISV або великим підприємством, MySQL може економічно ефективно допомогти вам забезпечити високу продуктивність, масштабованість баз даних. [11].

3 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Структурно-функціональне моделювання процесу інформаційної підтримки проведення змагань зі спортивних бальних танців

Перед початком розробки інформаційної системи дуже важливо правильно та зрозуміло побудувати структурно-функціональну модель процесу, який виконує дана ІС, а саме підтримку проведення змагань зі спортивних бальних танців.

Побудова функціональної моделі була здійснена у програмі AllFusion Process Modeler r7.

Вхідними даними до функції «Інформаційна підтримка проведення змагань зі спортивних бальних танців» є :

- Інформація про учасників;
- Інформація про суддів.

Вихідними даними є:

- Протоколи результатів;
- Списки категорій з учасниками.

Обов'язково в кожній функціональній моделі мають бути керуючі елементи процесу, в даному випадку це:

- Правила суддівства(Skating System);
- Регламент змагань.

Та механізмами являються:

- ІС;
- Адміністратор;
- Судді;
- Організатори.

Контекстну діаграму, що називається IDEF0, на якій зображено головний процес функціонування ІС, зображено на рис. 3.1.

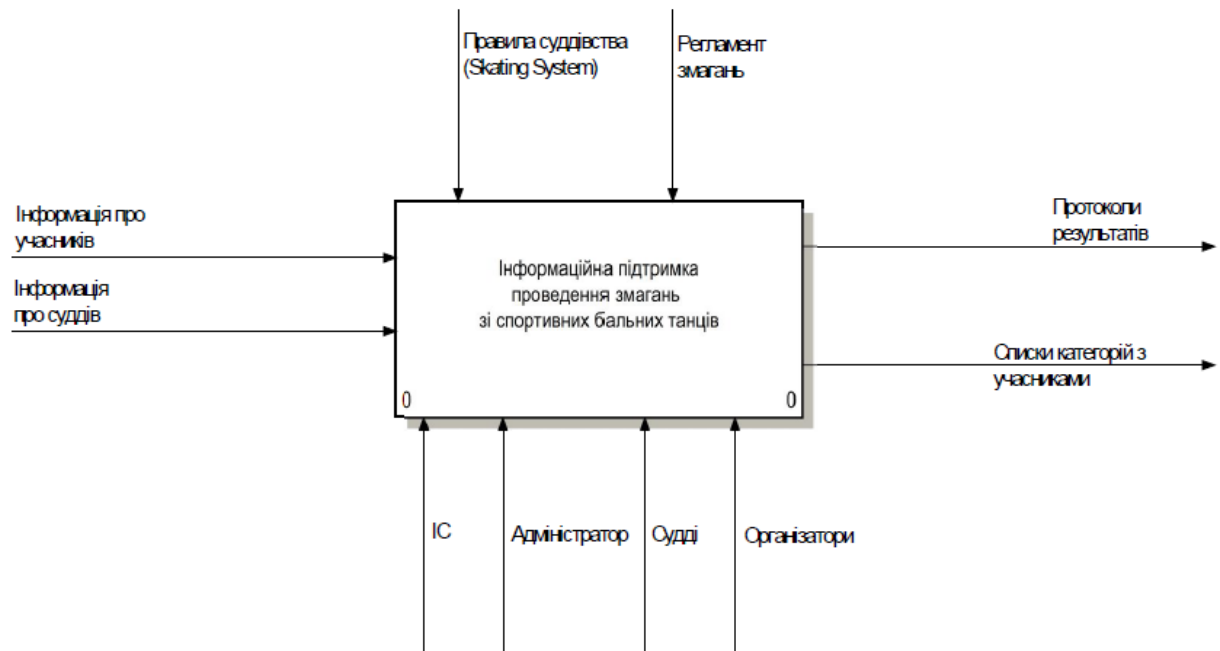


Рисунок 3.1 – Діаграма IDEF0

Задля того, щоб зазирнути глибше до більш розкритого опису роботи ІС потрібно зробити декомпозицію контекстної діаграми.

Діаграму було декомпововано на п'ять блоків. Це говорить про те, що дана система має 5 етапів роботи:

- Реєстрація;
- Формування списків категорій;
- Оцінювання категорій суддями;
- Підрахунок результатів кожної категорії змагань;
- Формування протоколів результатів.

Характеристика першого етапу «Реєстрація»:

- Вхідні дані: Інформація про учасників, інформація про суддів;
- Вихідні дані: Зареєстровані учасники, зареєстровані судді;
- Елементи керування: Регламент змагань;
- Механізми: ІС, адміністратор.

Характеристика другого етапу «Формування списків категорій»:

- Вхідні дані: Зареєстровані учасники;
- Вихідні дані: Списки категорій з учасниками;
- Елементи керування: Регламент змагань;
- Механізми: ІС, адміністратор, організатори.

Характеристика третього етапу «Оцінювання категорій судьями»:

- Вхідні дані: Зареєстровані судді, списки категорій з учасниками;
- Вихідні дані: Оцінки учасників;
- Елементи керування: Правила суддівства(Skating System);
- Механізми: ІС, судді.

Характеристика четвертого етапу «Підрахунок результатів кожної категорії змагань»:

- Вхідні дані: Оцінки учасників;
- Вихідні дані: Результати за категоріями;
- Елементи керування: Правила суддівства(Skating System);
- Механізми: ІС, адміністратор.

Характеристика п'ятого етапу «Формування протоколів результатів»:

- Вхідні дані: Результати за категоріями;
- Вихідні дані: Протоколи результатів;
- Елементи керування: Регламент змагань;
- Механізми: ІС, адміністратор.

Дані етапи зв'язані між собою, отже для того, щоб розроблена інформаційна системи працювала чітко та без будь-яких помилок, потрібно дотримуватися виконання усіх, зазначених вище, підпроцесів.

Для зручного розуміння роботи системи можна переглянути побудовану діаграму другого рівня декомпозиції[], яку представлено на рис. 3.2.

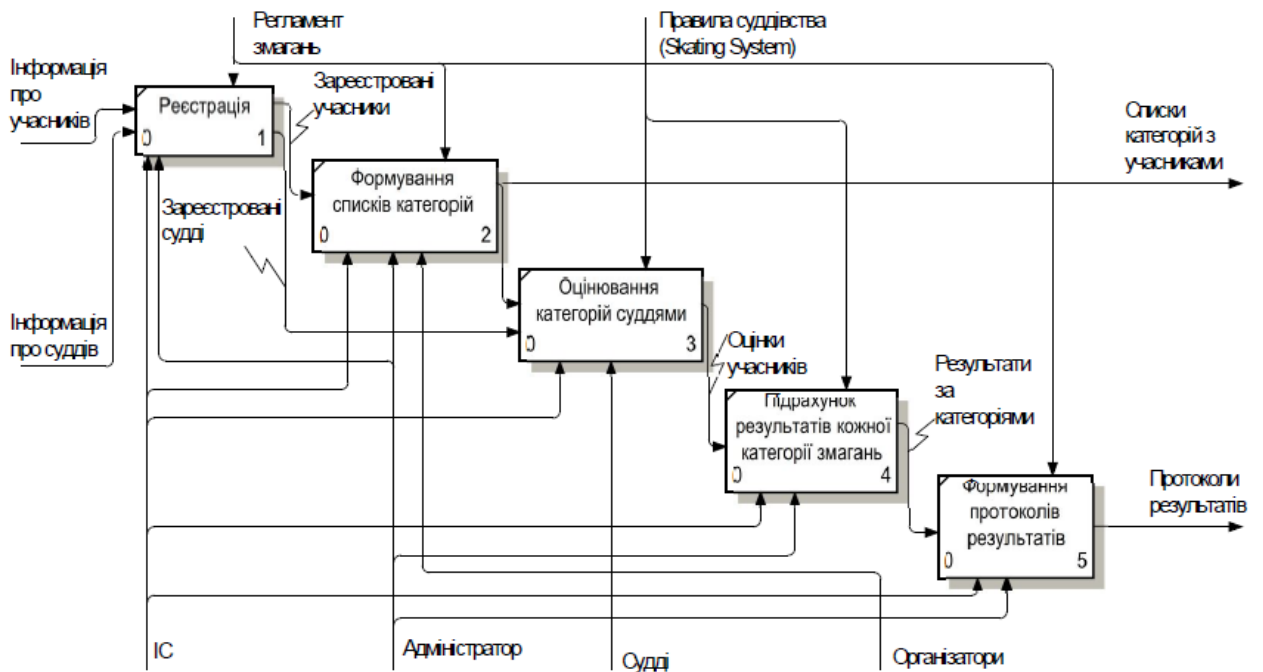


Рисунок 3.2 – Діаграма другого рівня декомпозиції

3.2 Моделювання варіантів використання інформаційної системи підтримки проведення змагань зі спортивних бальних танців

Варіанти використання - це опис послідовності дій, які може здійснювати система у відповідь на зовнішні дії користувачів або інших програмних систем[].

Були виділені наступні актори:

- Адміністратор – користувач, який буде працювати з даною ІС та має доступ до керування усіма варіантами використання;
- Суддя – користувач, який буде працювати з даною ІС та має доступ лише до деяких варіантів використання;;
- База даних(БД) – сховище даних, дані до якої додаються або використовуються впродовж роботи з ІС.

Варіанти використання:

- Реєстрування;
- Формування списків категорій;
- Друк/вивід списків категорій;
- Оцінювання категорій;
- Підрахунок результатів;
- Формування протоколів змагань;
- Друк/вивід протоколів.

На рисунку 3.3 представлена діаграма варіантів використання ІС.

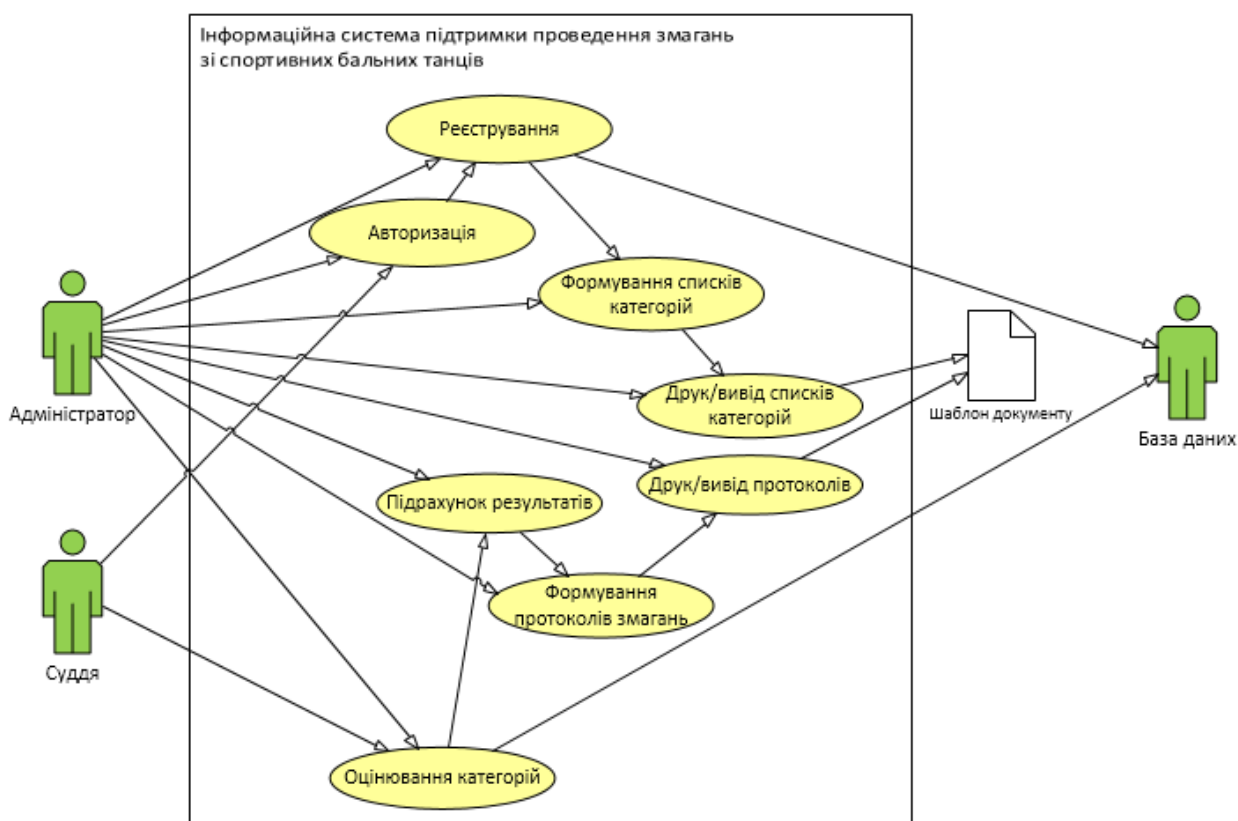


Рисунок 3.3 – Модель варіантів використання ІС

3.3 Проектування бази даних

Виходячи з побудованої моделі варіантів використання дана інформаційна система має зберігати інформацію в базі даних(БД). Отже, було створено базу даних у середовищі MySQL за допомогою веб-додатку phpMyAdmin(рис. 3.4).

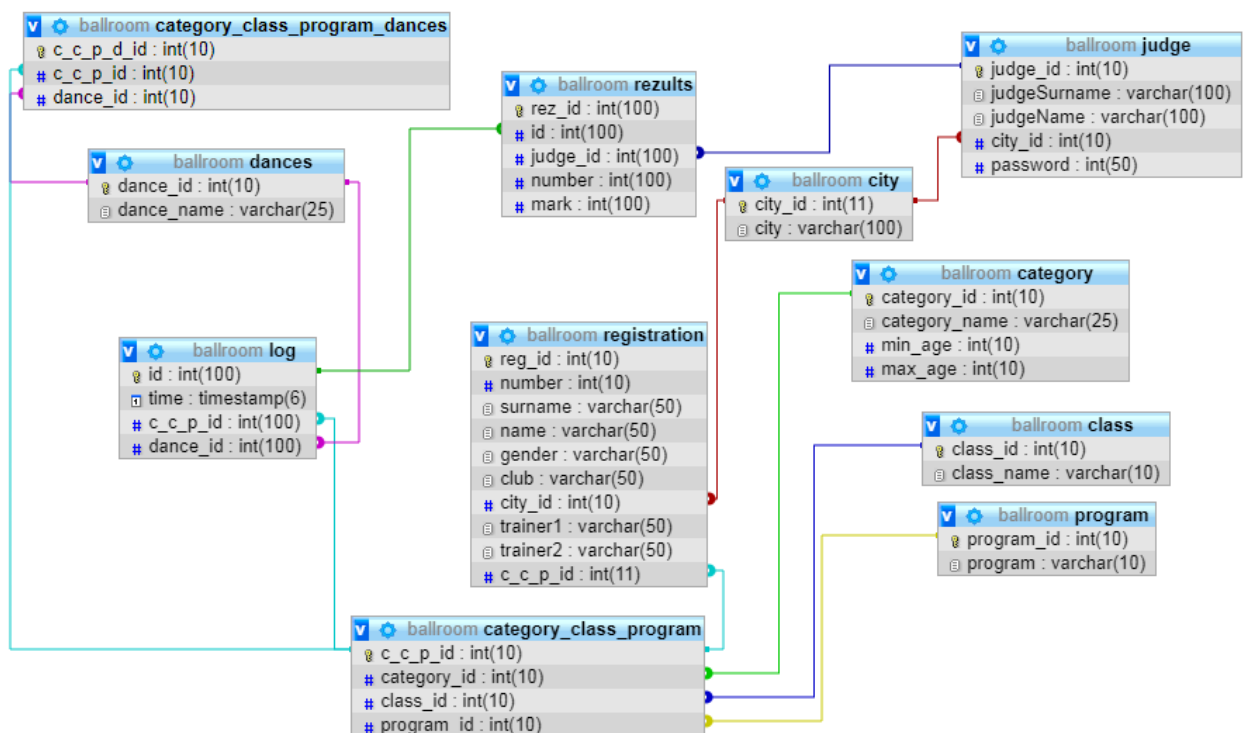


Рисунок 3.4 – База даних ІС

Дана БД має 11 таблиць, які мають певні атрибути:

- Category (категорія). Атрибути: category_name(ім'я категорії), min_age(мінімальний вік), max_age(максимальний вік);
- Class (клас). Атрибут: class_name(ім'я класу);
- Program (програма). Атрибути: program(програма);
- Category_class_program (категорія_клас_програма). Зв'язуюча таблиця;

- Dances (танці). Атрибут: dance_name(ім'я танцю);
- Category_class_program_dances (категорія_клас_програма_танці). Зв'язуюча таблиця;
- Registration (реєстрація). Атрибути: number(номер учасника/пари), surname(прізвище учасника), name(ім'я учасника), gender(стать), club(клуб), trainer1(перший тренер), trainer2(другий тренер);
- City(місто). Атрибут: city(місто);
- Judges(судді). Атрибути: judgeName(ім'я судді), judgeSurname(прізвище судді), password(пароль);
- Log(категорія, танець в даний час). Атрибут: time(час початку);
- Results(результати). Атрибути: number(номер учасника/пари), mark(оцінка).

4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ПРОВЕДЕННЯ ЗМАГАНЬ ЗІ СПОРТИВНИХ БАЛЬНИХ ТАНЦІВ

4.1 Установка та запуск компонентів ІС

Найважливішою функцією розробленої інформаційної системи є функція завантаження та зчитування даних із СУБД MySQL *ballroom.sql*, бо без цих даних не будуть виконуватись майже всі функції ІС. Тому було створено базу даних(рис. 4.1), про яку детально описано у 3 розділі, пункт 3.3.

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/> category	☆ [Icons]	13	InnoDB	utf8_general_ci	16 КИБ	-
<input type="checkbox"/> category_class_program	☆ [Icons]	72	InnoDB	utf8_general_ci	64 КИБ	-
<input type="checkbox"/> category_class_program_dances	☆ [Icons]	282	InnoDB	utf8_general_ci	48 КИБ	-
<input type="checkbox"/> city	☆ [Icons]	489	InnoDB	utf8_general_ci	48 КИБ	-
<input type="checkbox"/> class	☆ [Icons]	8	InnoDB	utf8_general_ci	16 КИБ	-
<input type="checkbox"/> dances	☆ [Icons]	10	InnoDB	utf8_general_ci	16 КИБ	-
<input type="checkbox"/> judge	☆ [Icons]	8	InnoDB	utf8_general_ci	32 КИБ	-
<input type="checkbox"/> log	☆ [Icons]	10	InnoDB	utf8_general_ci	48 КИБ	-
<input type="checkbox"/> program	☆ [Icons]	3	InnoDB	utf8_general_ci	16 КИБ	-
<input type="checkbox"/> registration	☆ [Icons]	6	InnoDB	utf8_general_ci	48 КИБ	-
<input type="checkbox"/> results	☆ [Icons]	35	InnoDB	utf8_general_ci	48 КИБ	-
11 таблиц	Всего	936	InnoDB	utf8_general_ci	400 КИБ	0 Байт

Рисунок 4.1 – База даних у MySQL

Для роботи із БД MySQL необхідно до проекту в середовищі розробки Microsoft Visual Studio 2010 Ultimate підключити бібліотеку типів MySQL, яку було завантажено за допомогою інсталятора *mysql-connector-net-6.9.12.msi* та додано до посилань проекту(рис. 4.2)

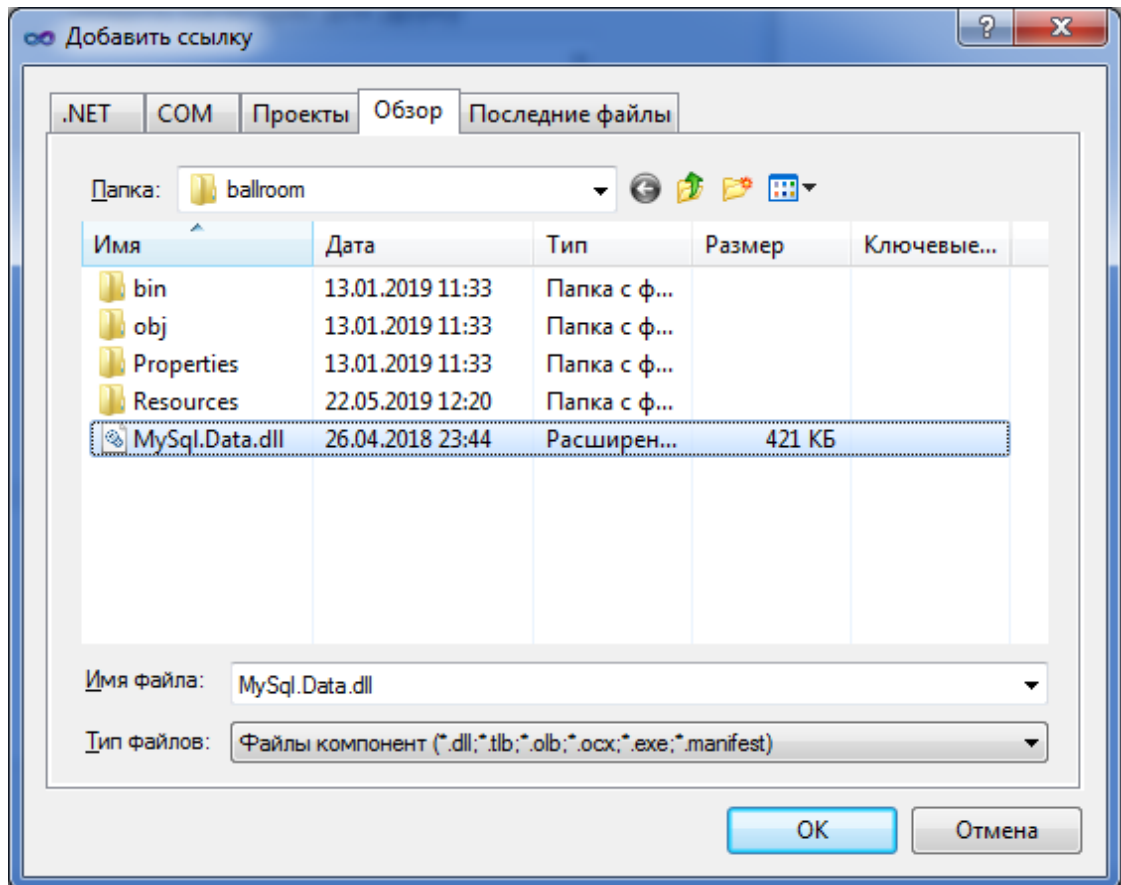


Рисунок 4.2 – Підключення бібліотеки

Для повноцінної роботи із БД MySQL були оголошені модулі MySQL.Data та MySQL.Data.MySqlClient на початку кожного файлу коду IC (рис. 4.3).

```
using MySql.Data;
using MySql.Data.MySqlClient;
```

Рисунок 4.3 – Оголошені модулі

Далі до кожного файлу коду IC було додано рядок для підключення до БД MySQL *ballroom* (рис. 4.4).

```
string connStr = "server=localhost;user=root;password=;database=ballroom;port=3306;";
```

Рисунок 4.4 – Рядок підключення

Після того, як зазначені вище кроки були виконані, вже можна було приступити до виконання простих запитів MySQL. Приклади основних запитів наведені на рис.4.5-4.6

```

MySQLConnection conn = new MySqlConnection(connStr);
MySQLCommand command = new MySQLCommand();

string commandString = "SELECT c.city FROM city c";

command.CommandText = commandString;
command.Connection = conn;
MySQLDataReader reader;
try
{
    command.Connection.Open();
    reader = command.ExecuteReader();
    int i = 0;
    while (reader.Read())
    {
        string city = Convert.ToString(reader["city"]);
        cityy.Items.Add(city);
        i += 1;
    }

    reader.Close();
}
catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{
    command.Connection.Close();
}

```

Рисунок 4.5 – Приклад додавання результатів запиту на форму з БД

```

string strSQL = "INSERT INTO `registration` (`reg_id`, `number`, `surname`, `name`, `gender`, `club`, `city_id`, `trainer1`, `trainer2`, `c_c_p_id`) VA
try
{
    using (MySQLConnection connection = new MySqlConnection(connStr))
    {
        connection.Open();
        MySQLCommand cp = new MySQLCommand(strSQL, connection);

        if (cp.ExecuteNonQuery() == 1)
        {
            MessageBox.Show("Реєстрація за категорією №" + num + " пройшла успішно!");
            MessageBox.Show("Учасника " + girlSurname.Text + " " + girlName.Text + " додано!");
        }
        connection.Close();
    }
}

catch (MySqlException ex)
{
    MessageBox.Show(ex.Message);
}

```

Рисунок 4.6 – Приклад запису даних до БД

Однією з функціональних вимог ІС є те, що система має бути побудована за клієнт-серверною архітектурою(рис. 4.5). Тобто декілька персональних комп'ютерів(ПК) повинні мати доступ до однієї бази даних через локальний сервер.



Рисунок 4.7 – Клієнт-серверна архітектура

Оскільки необхідно запустити додаток з централізованою базою даних, були надані привілеї цьому комп'ютеру в локальній мережі для доступу до конкретної бази даних, розміщеної на ПК локальної мережі.

Для даної операції потрібно було виконати наступні дії:

1. Перейти на сервер MySQL;
2. Ввести наступний код(рис. 4.6), щоб надати доступ для іншого комп'ютера.

```
1 GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'password' WITH GRANT OPTION;
2 FLUSH PRIVILEGES;
```

Рис. 4.8 – Код для надання доступу

4.2 Результат реалізації інформаційної системи

Після встановлення ІС системи на ПК за допомогою настанови з використання(Додаток Г) на робочому столі з'являється ярлик на робочому столі(рис 4.9).

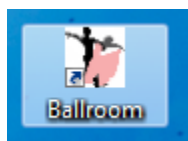


Рисунок 4.9 – Ярлик на робочому столі

По натисканню на дану іконку завантажується інформаційна система. З'являється вікно привітання(рис 4.10), що демонструє назву розробленої ІС, розробників та назву клубу, в роботу якого її впроваджено.



Рис 4.10 – Вікно привітання

Через декілька секунд з'являється форма «Вхід»(рис.4.11). Якщо користувач є адміністратором, то потрібно натиснути кнопку «Адміністратор», якщо суддею – «Суддя».

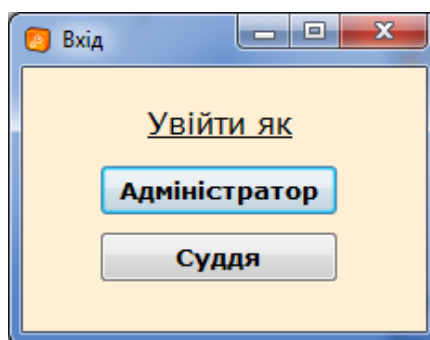


Рисунок 4.11 – Форма «Вхід»

4.2.1 Натиснута кнопка «Адміністратор»

З'являється вікно входу «Admin», як адміністратора(рис. 4.12) для подальшого керування процесами підтримки проведення змагань. Потрібно ввести логін та пароль(рис. 4.13), які відомі лише адміністратору змагань та натиснути кнопку «Увійти». Якщо логін/пароль введено невірно або взагалі дані не введено виникає віконце помилки, де зазначено, що саме зроблено не так(рис. 4.14-4.15). Та в позитивному випадку далі з'являється форма керування «Програма»(4.16).

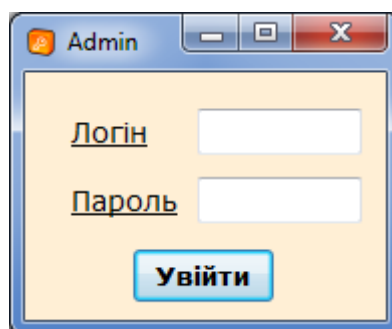


Рисунок 4.12 – Вікно входу «Admin»

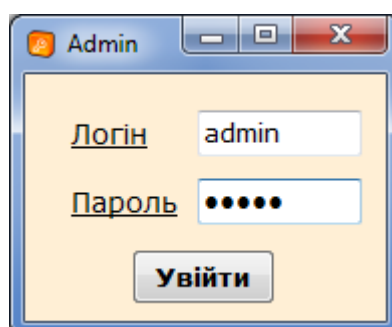


Рисунок 4.13 – Заповнення полів входу

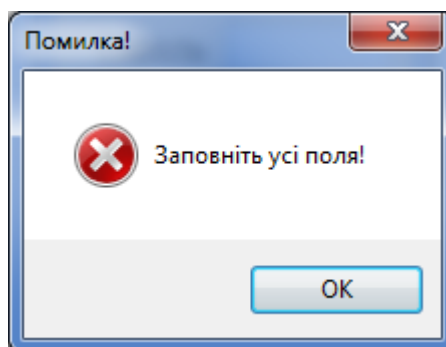


Рисунок 4.14 – Вікно помилки незаповнених даних

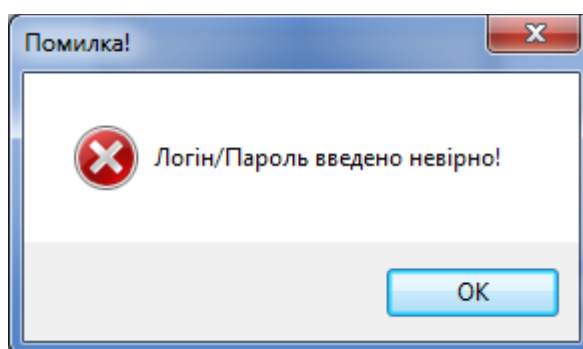


Рисунок 4.15 – Вікно помилки невірно введених даних

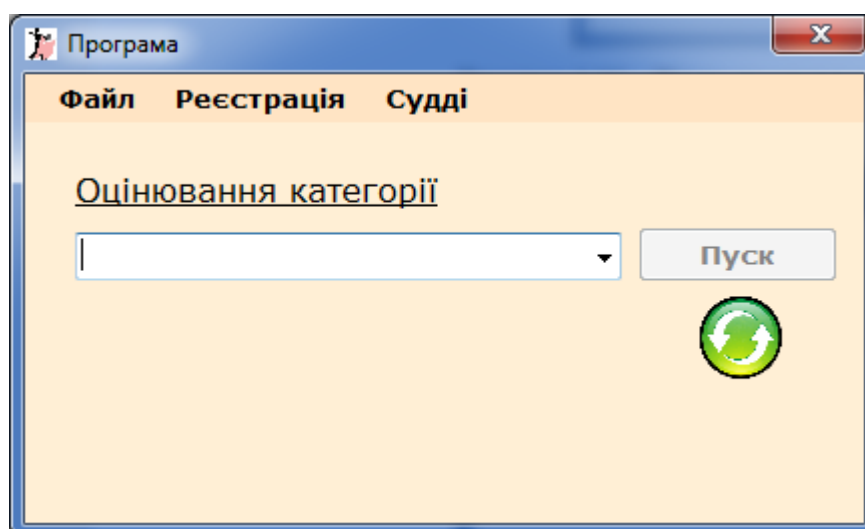


Рисунок 4.16 – Форма керування «Програма»

На формі керування знаходяться вкладки меню «Файл», «Реєстрація», «Судді» та елементи для запуску оцінювання певної категорії на ПК суддів.

Елемент меню «Реєстрація» передбачає реєстрацію учасників по натиску на випадаюче меню «Додати»(рис. 4.17).

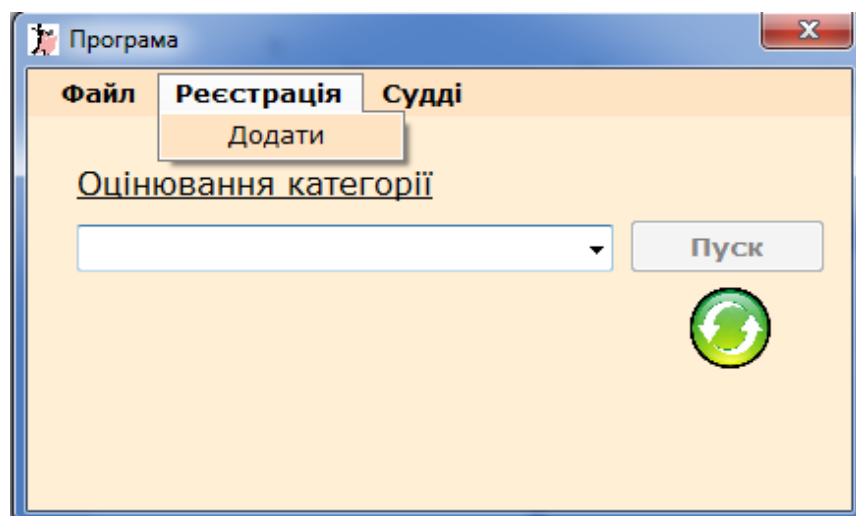


Рисунок 4.17 – Випадаюче меню «Додати»

Далі виникає вікно «Реєстрація», де можна зареєструвати чоло учасника, або пару, що приймає участь у змаганнях зі спортивних бальних танців(рис. 4.18).

Рисунок 4.18 – Вікно «Реєстрація»

Якщо будь-яке обов'язкове поле буде не заповнене, то з'явиться вікно помилки(рис. 4.19). У позитивному випадку з'являться вікна успішної реєстрації(рис. 4.20-4.21)

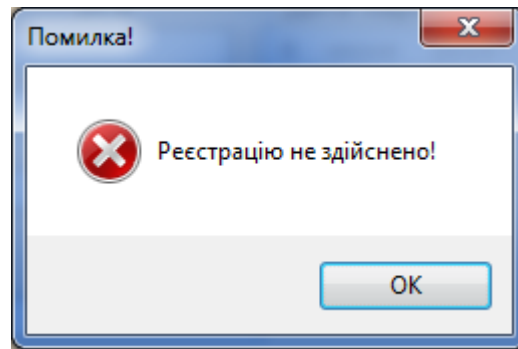


Рисунок 4.19 – Вікно помилки

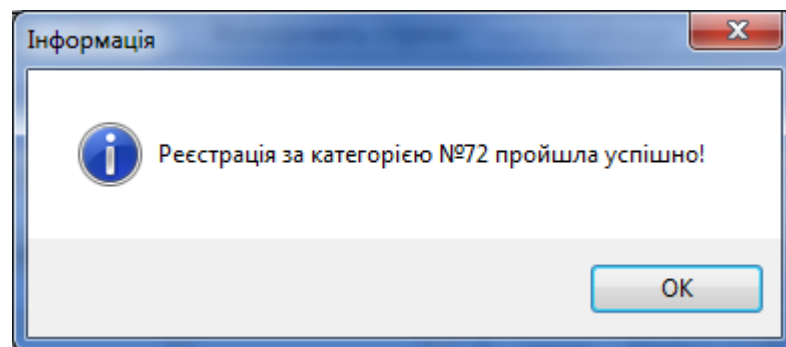


Рисунок 4.20 – Реєстрація пройшла успішно

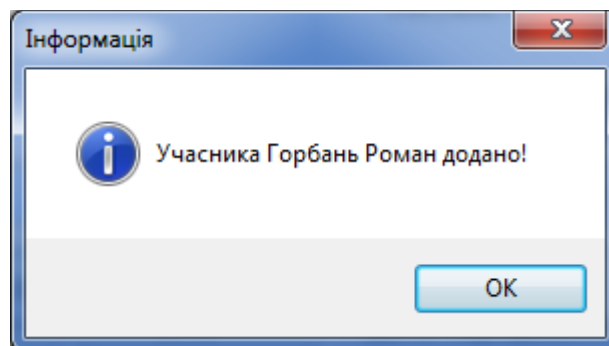


Рисунок 4.21 – Учасника додано

Далі після реєстрації учасників адміністратор має зареєструвати суддів, які були запрошені оцінювати змагання. Дану функція виконується за допомогою елементу меню «Судді» - «Додати»(рис. 4.22), після натискання на який виникає вікно «Новий суддя»(рис. 4.23)

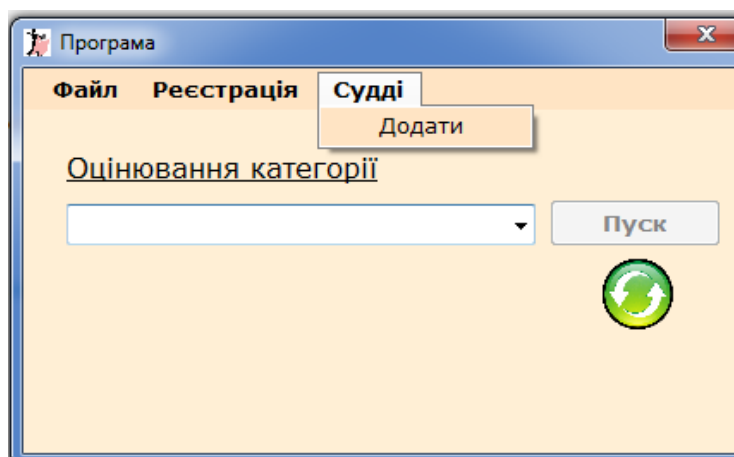


Рисунок 4.22 - Елемент меню «Судді» - «Додати»

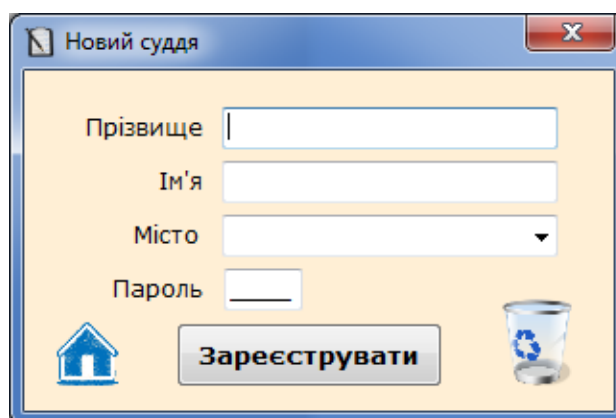


Рисунок 4.23 – Вікно «Новий суддя»

Реєстрація суддів здійснюється за однаковим принципом, як і реєстрація учасників. Відмінною рисою є наявність паролю з 4 цифр(максимально). Також, при заповненні не всіх полів або взагалі жодного – з'являється вікно помилки(рис. 4.24). При успішній реєстрації судді можна побачити вікно, що представлено на рисунку 4.25.

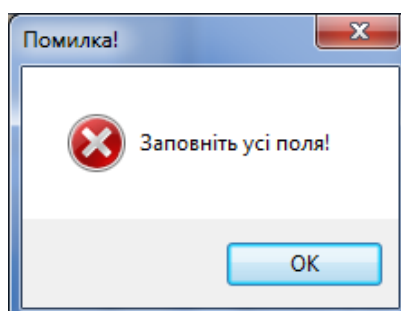


Рисунок 4.24 – Вікно помилки незаповнених даних

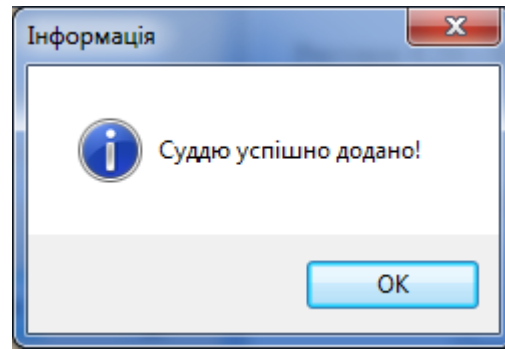


Рисунок 4.25 – Успішна реєстрація судді

Після реєстрації усіх учасників певної категорії є можливість роздрукувати списки зареєстрованих учасників за допомогою функції друку, що реалізована на формі «Друк списків учасників»(рис. 4.27), яка з'являється після вибору «Файл»-«Друк списків учасників»(рис. 4.26)

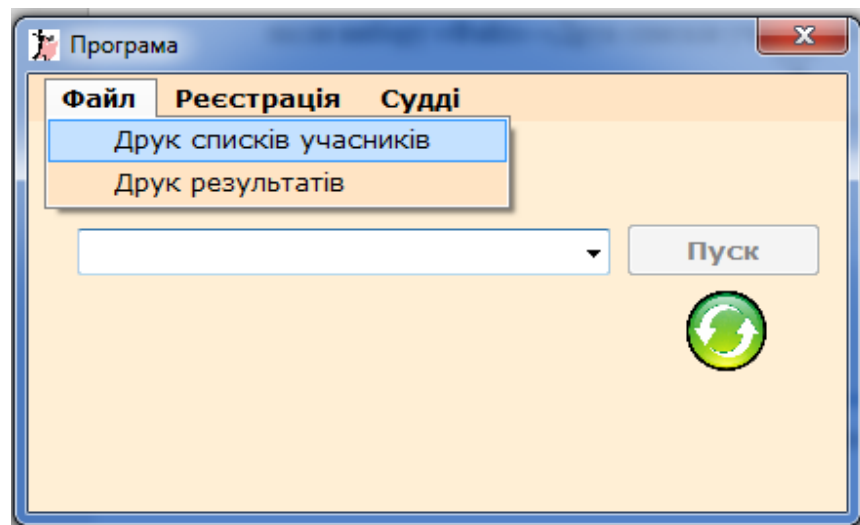


Рисунок 4.26 – Елемент меню «Файл»-«Друк списків учасників»

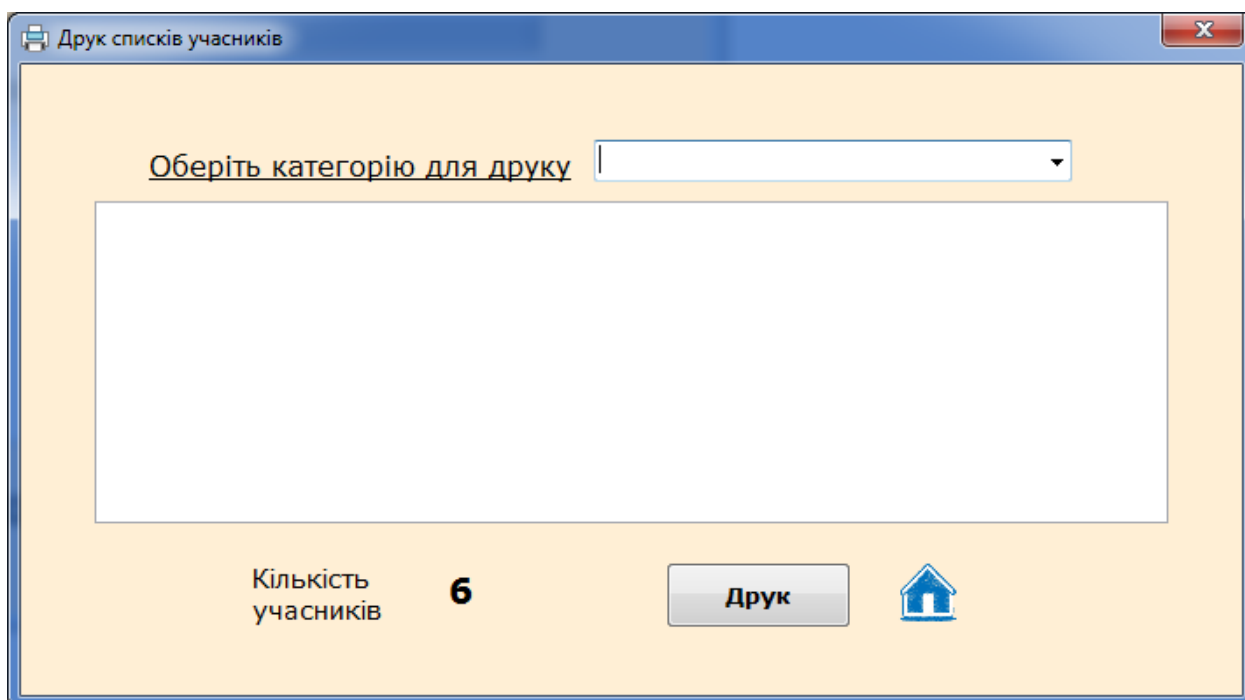


Рисунок 4.27 – Вікно «Друк списків учасників»

У випадяючому списку даної форми потрібно обрати категорію, список учасників якої бажано роздрукувати (рис.4.28).

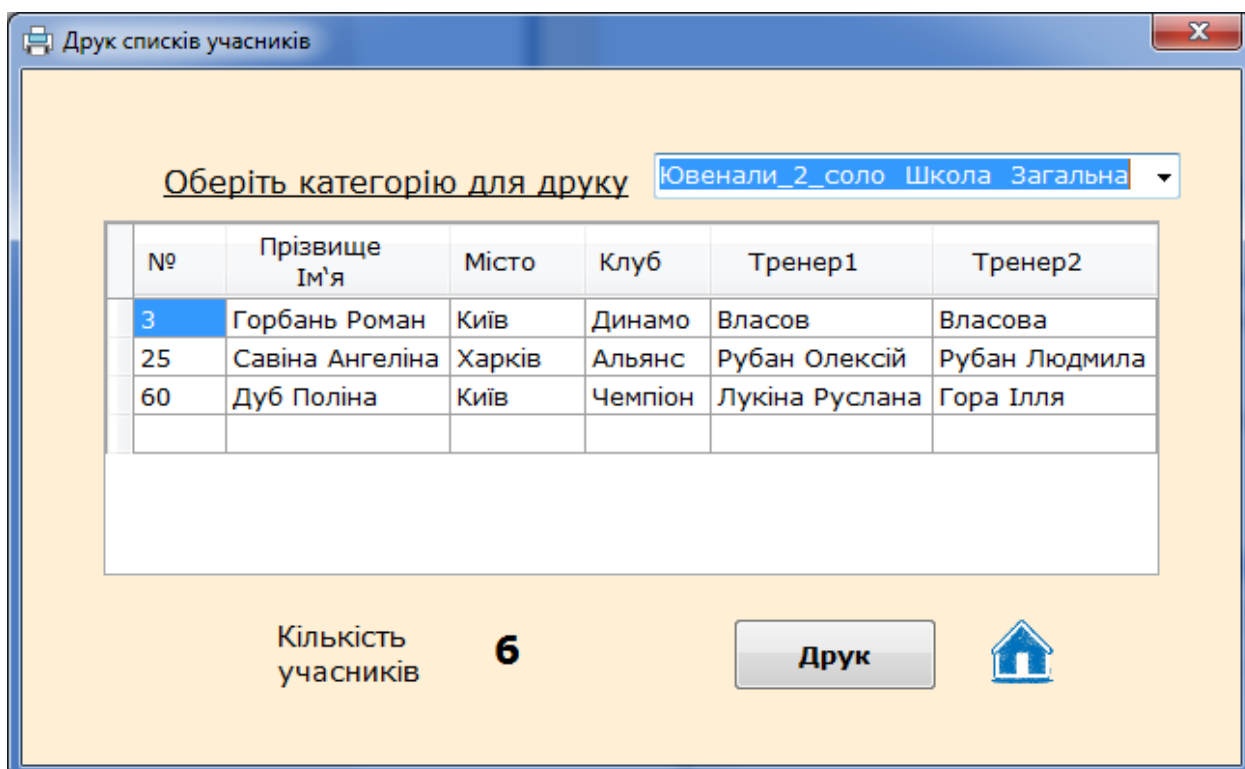


Рисунок 4.28 – Список за обраною категорією

Натиснувши кнопку «Друк» з'явиться вікно для попереднього перегляду та друку отриманого списку(рис. 4.29).

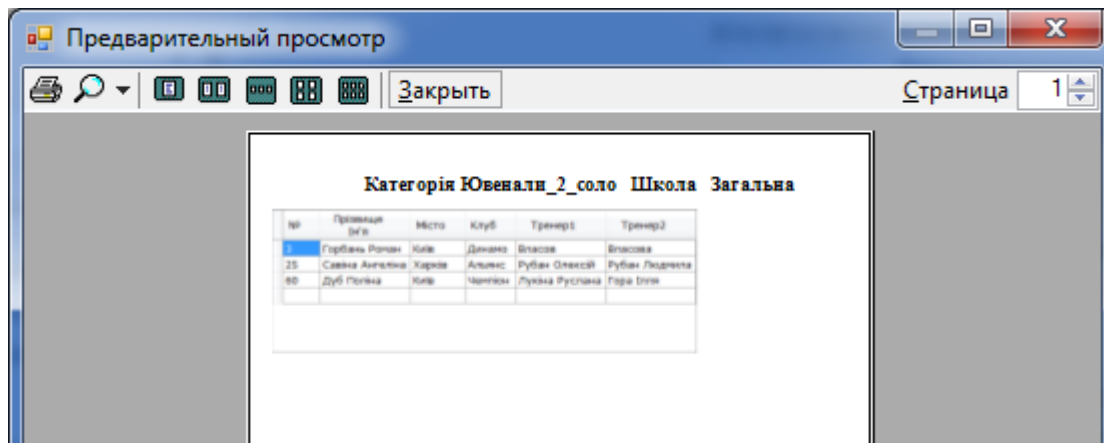


Рисунок 4.29 – Вікно попереднього перегляду та друку

Адміністратор повинен запускати оцінювання кожного танцю кожної категорії на комп'ютерах суддів. Адміністратор обирає категорію з випадаючого списку та танець, який будуть виконувати на паркеті учасники, у вікні «Програма»(рис.4.30). Для оновлення категорій у випадаючому списку адміністратор має натискати кнопку оновлення, яка зображена рисунком на даній формі, після реєстрації учасника/пари. Після натиску на кнопку «Пуск» з'являється вікно, що показує чи почалося оцінювання(рис. 4.31).

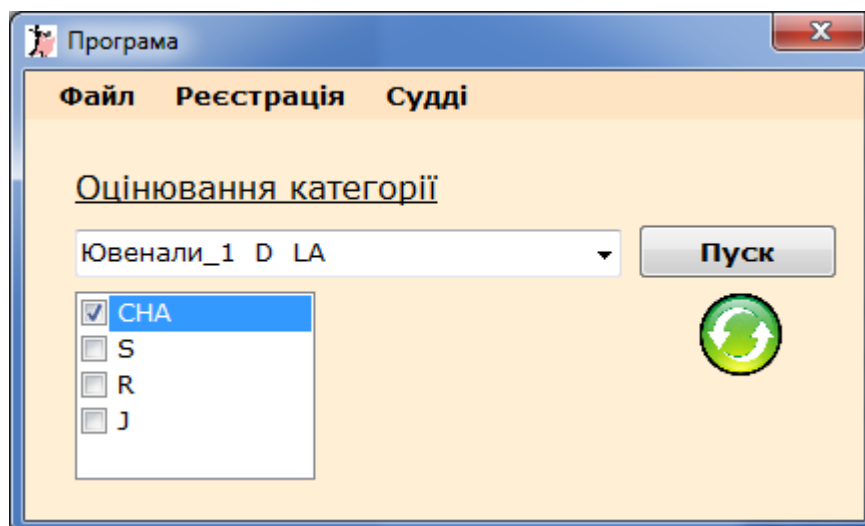


Рисунок 4.30 – Вибір категорії та танцю для оцінювання

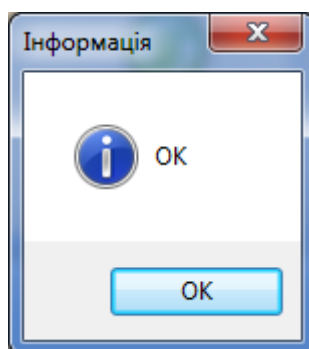


Рисунок 4.31 – Вікно успішності запуску оцінювання

Якщо певна категорія завершила своє змагання, судді оцінили усіх учасників даної категорії адміністратор може підрахувати оцінки, отримати, результати та роздрукувати протоколи результатів, натиснувши «Файл»-«Друк результатів»(рис 4.32). Після чого з'явиться вікно «Друк результатів» (рис.4.33)

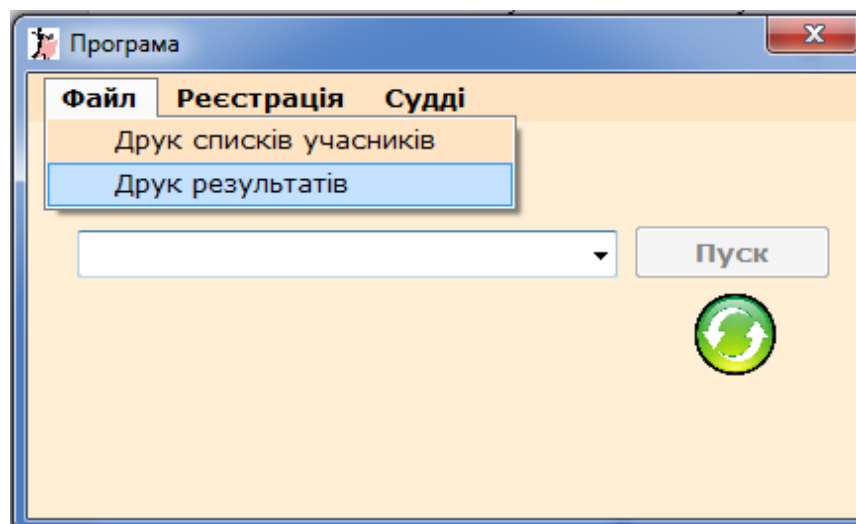


Рисунок 4.32 – Елемент меню «Файл»-«Друк результатів»

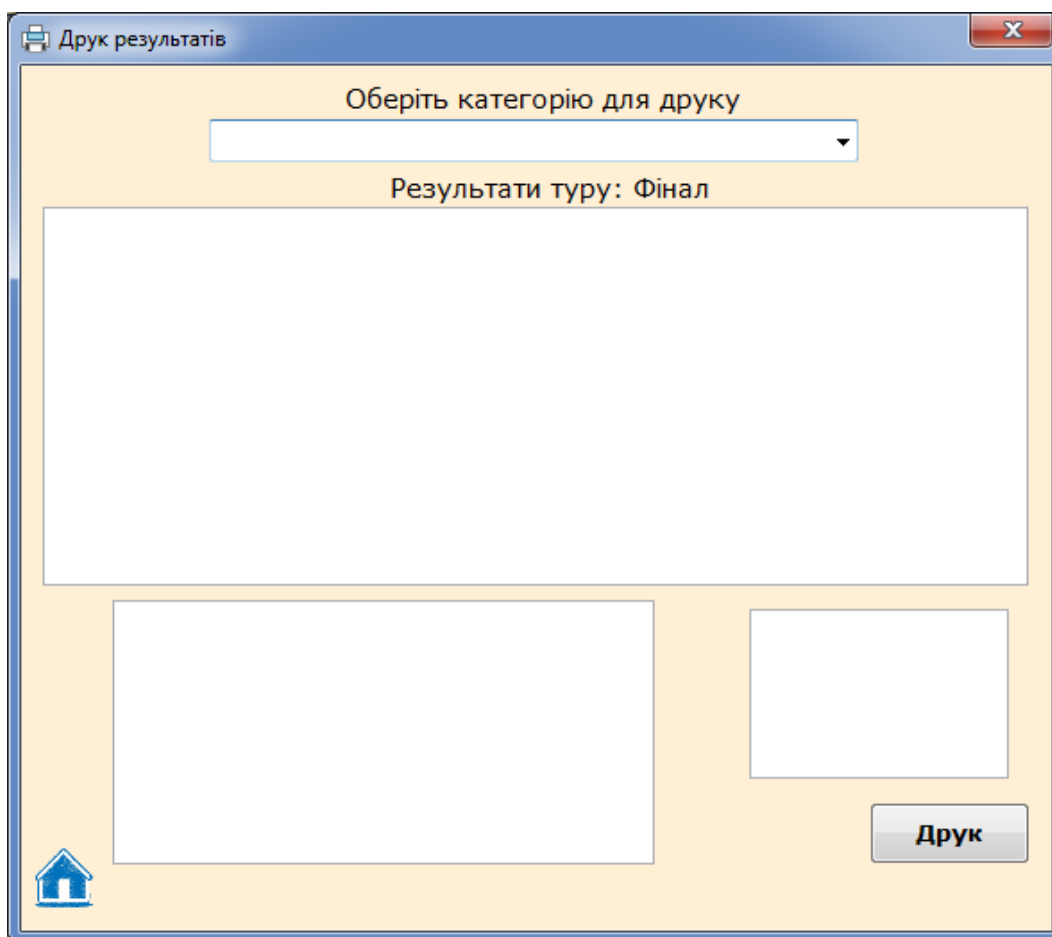


Рисунок 4.33 – Вікно «Друк результатів»

Для друку потрібно обрати категорію, після чого одразу з'являються результати за цією категорією(рисю 4.34) та по натиску на кнопку «Друк» з'явиться вікно для попереднього перегляду та друку протоколів результатів(рис. 4.35)

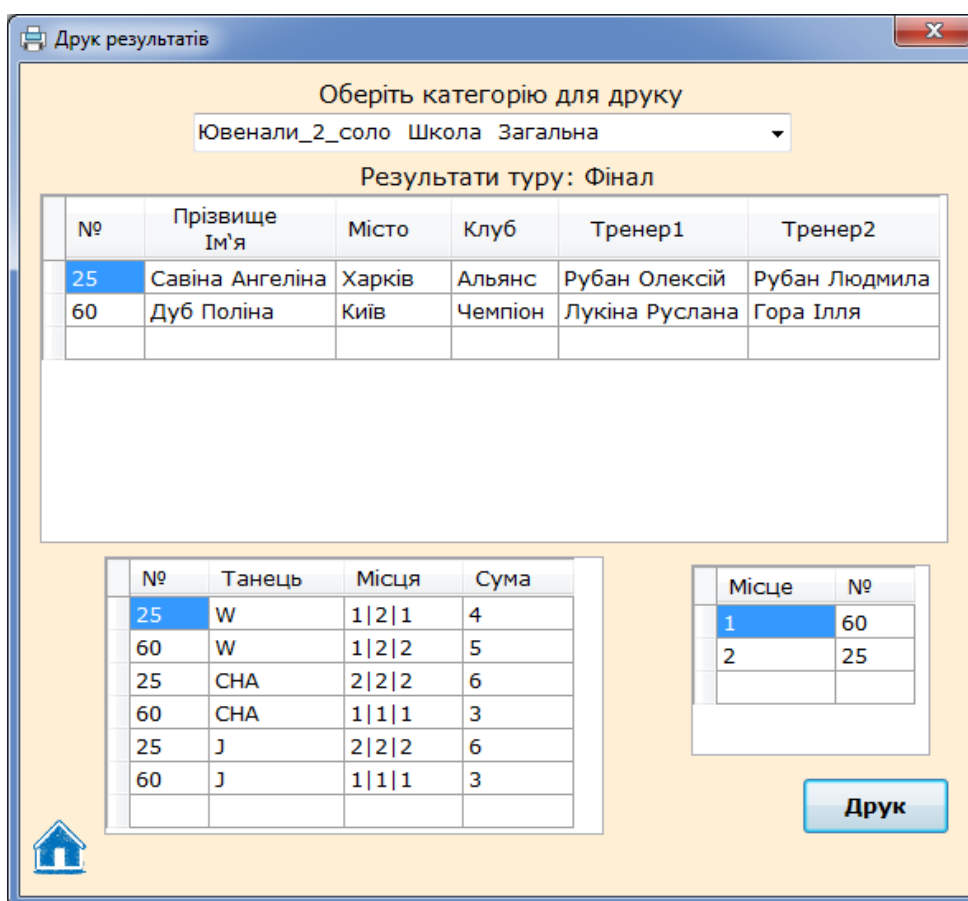


Рисунок 4.34 – Результати за обраною категорією

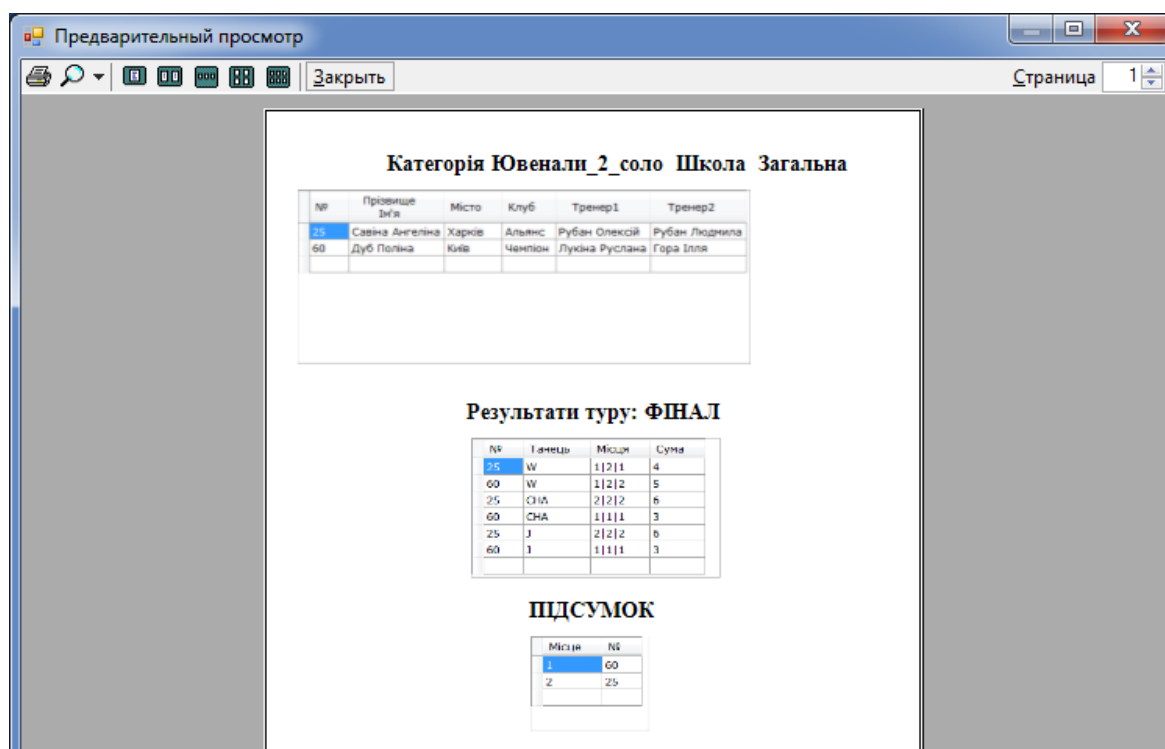


Рисунок 4.35 – Вікно попереднього перегляду та друку протоколів результатів

4.2.1 Натиснута кнопка «Суддя»

З'являється вікно входу «Judge», де суддя повинен обрати своє ім'я з випадаючого списку та ввести пароль, за яким було здійснено реєстрацію(рис. 4.36). Якщо введено невірний пароль або поля не заповнені – виникає вікно помилки(рис. 4.37-4.38)

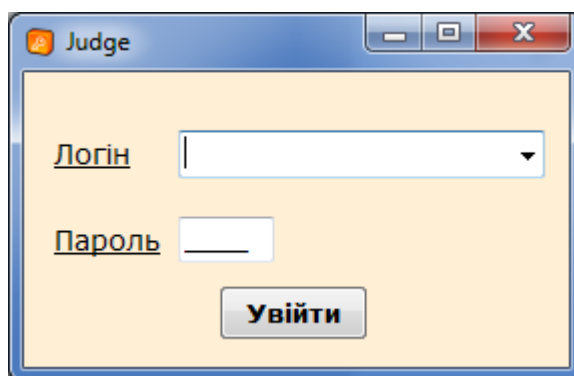


Рисунок 4.36 – Вікно входу «Judge»

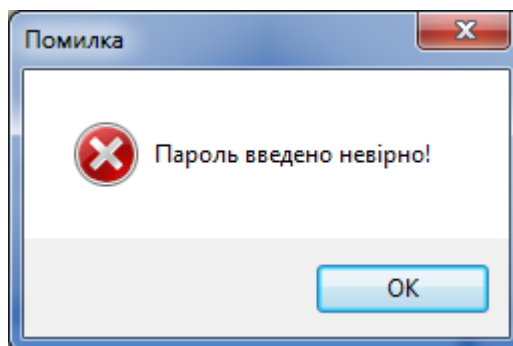


Рисунок 4.37 – Вікно помилки невірно введеного паролю

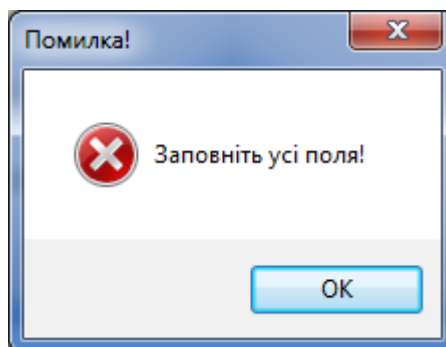


Рисунок 4.38 – Вікно помилки незаповнених даних

При успішному вході з'являється вікно завантаження форми оцінювання категорії та танцю, обраних на ПК адміністратора(рис. 4.39). По завершенню завантаження виникає вікно «Оцінювання фінал», де суддя повинен оцінити пари/учасників, що приймають участь в даній категорії(рис.4.40).

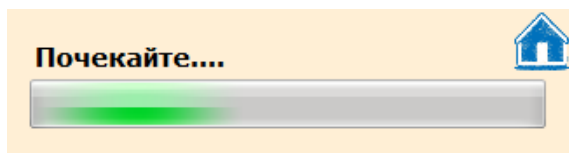


Рисунок 4.39 – Вікно завантаження

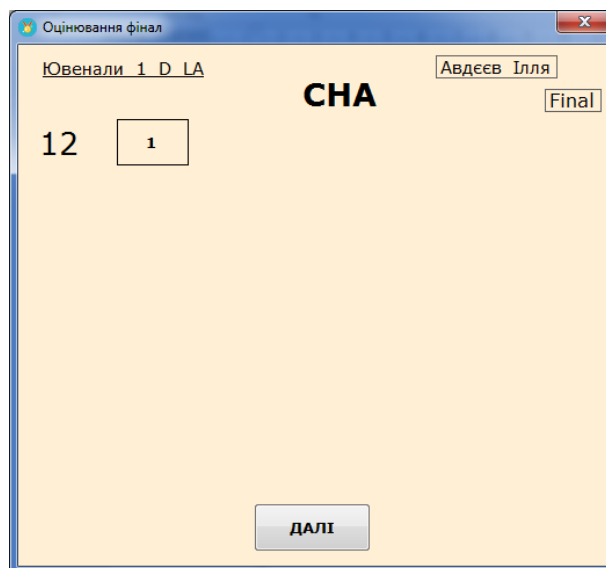


Рисунок 4.40 - Вікно «Оцінювання фінал»

Якщо суддя вважає, що пара/учасник танцює на перше місце, то потрібно натиснути кнопку з цифрою «1» і т.д. Обрана кнопка стає зеленого кольору(рис. 4.41).

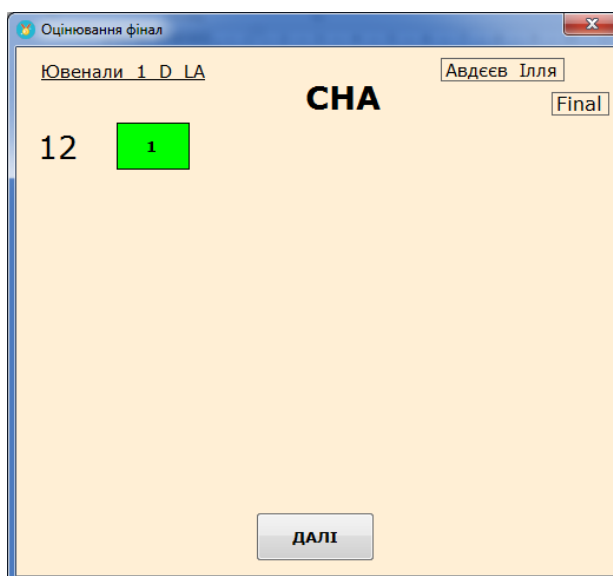


Рисунок 4.41 – Приклад оцінювання суддею

Якщо суддя не обрав у якогось учасника місце - виникає дане повідомлення(рис. 4.42).

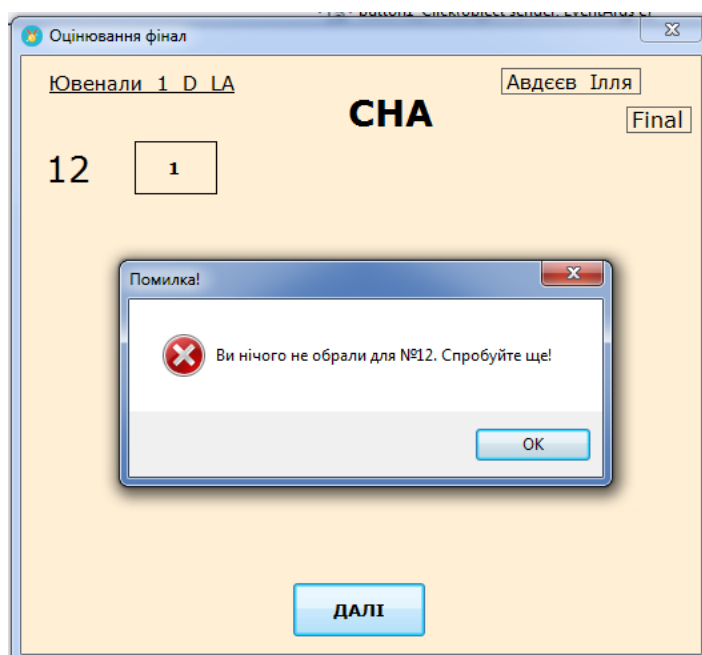


Рисунок 4.42 – Повідомлення

По натисканню на кнопку «Далі» у судді знову з'являється форма завантаження та подальші етапи здійснюються циклічно.

Задля зручного переходу до попередньої сторінки на деяких формах реалізована кнопка(рис. 4.43) та для очищення полів кнопка, що зображена на рис. 4.44.



Рисунок 4.43 – Кнопка повернення на попередню форму

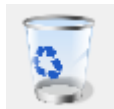


Рисунок 4.44 – Кнопка очищення полів

ВИСНОВОК

З метою розробки інформаційної системи підтримки проведення змагань зі спортивних бальних танців було проведено дослідження актуальності проблеми. За результатами огляду інформаційних систем підтримки проведення змагань зі спортивних бальних танців було прийняте рішення розробити унікальну інформаційну систему, власником та користувачем якої буде КСТ Силует, що знаходиться у місті Суми.

Були визначені такі задачі для досягнення поставленої мети: аналіз вимог та інформаційних систем підтримки проведення змагань зі спортивних бальних танців, схожих на розроблювану ІС, вибір методів та засобів реалізації, проектування і моделювання інтерфейсу, програмна реалізація модулів системи і тестування роботи отриманої ІС методом чорного ящика.

Були обрані засоби реалізації, згідно з отриманого ТЗ.

Програмний код інформаційної системи розроблено на мові С# у середовищі MS Visual Studio 2010 Ultimate через можливість редагування коду під час використання даної ІС.

Для створення бази даних була обрана реляційна система управління базами даних MySQL, що забезпечує потрібний рівень безпеки даних та зручність організації паралельного доступу до даних.

Для реалізації архітектури ІС, як клієнт-серверної, було здійснено підключення та надано доступ комп'ютерам клієнтів до бази даних, що знаходиться на серверному комп'ютері через локальну мережу.

Тестування розробленої ІС розробниками і користувачами методом чорного ящика показало, що система задовольняє визначеним вимогам, зручна у використанні.

У результаті розробки отримано ІС, яка забезпечує автоматизовану підтримку необхідних бізнес-процесів з організації змагань зі спортивних бальних танців.

Інформаційна система підтримки проведення змагань зі спортивних бальних танців було впроваджено в роботу організаційного комітету КСТ Силует.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Скейтинг-система: правила подсчета результатов соревнований [Электронный ресурс] – Режим доступа до ресурсу: <http://ltu.org.ua/skejtynh-systema-pravy-la-podscheta-rezultatov-sorevnovanyj/>
2. Скрутинер Вадим Погорелов :: Счетная бригада из Николаева [Электронный ресурс] – Режим доступа до ресурсу: <http://ballroom.vpogorelova.com/>
3. Skating System 5 [Электронный ресурс] – Режим доступа до ресурсу: <http://www.skatingsystem.com/rus/about/functions/>
4. Skating System Software: The concept [Электронный ресурс] – Режим доступа до ресурсу: <http://laurent.riesterer.free.fr/skating/intro-main.html>
5. C Sharp – Вікіпедія [Электронный ресурс] – Режим доступа до ресурсу: https://uk.wikipedia.org/wiki/C_Sharp.
6. Основы C#. Комментарии и переменные [Электронный ресурс] – Режим доступа до ресурсу: <https://shwanoff.ru/c-sharp-1/>
7. Microsoft Visual Studio – Вікіпедія [Электронный ресурс] – Режим доступа до ресурсу: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio
8. Microsoft Visual Studio 2010 Ultimate [Электронный ресурс] – Режим доступа до ресурсу: <https://ru.freownloadmanager.org/Windows-PC/Microsoft-Visual-Studio-2010-Ultimate.html>
9. Open Server Panel — Локальный веб-сервер для Windows. [Электронный ресурс] – Режим доступа до ресурсу: <https://ospanel.io/>
10. phpMyAdmin [Электронный ресурс] – Режим доступа до ресурсу: <https://www.phpmyadmin.net/>
11. MySQL Editions [Электронный ресурс] – Режим доступа до ресурсу: <https://www.mysql.com/products/>
12. AllFusion Process Modeler 7 (BPwin) [Электронный ресурс] – Режим доступа до ресурсу: <http://www.interface.ru/home.asp?artId=106>

13. Методологія IDEF0) [Електронний ресурс] – Режим доступу до ресурсу: <https://itteach.ru/bpwin/metodologiya-idef0>
14. Построение диаграмм декомпозиции второго уровня в нотации IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: <https://lektsii.org/2-2959.html>
15. Варианты использования (Use Case) IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: http://www.caseclub.ru/articles/use_case.html
16. База данных – Что это такое? [Електронний ресурс] – Режим доступу до ресурсу: <https://hostiq.ua/wiki/database/>
17. Download Connector/NET [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/downloads/connector/net/6.9.html>
18. Архітектура клієнт – сервер [Електронний ресурс] – Режим доступу до ресурсу: <https://homework.net.ua/arkhitektura-klient-server/>
19. Подключение к серверу mysql на другом ПК в локальной сети [Електронний ресурс] – Режим доступу до ресурсу: <http://qaru.site/questions/908318/connecting-to-mysql-server-on-another-pc-in-lan>
20. IconArchive [Електронний ресурс] – Режим доступу до ресурсу: <http://www.iconarchive.com/>

Додаток А. Технічне завдання

ТЕХНІЧНЕ ЗАВДАННЯ на розробку інформаційної системи «Інформаційна система підтримки проведення змагань за спортивних бальних танців»

Суми 2019

1 Призначення й мета створення інформаційної системи

1.1 Призначення інформаційної системи

Інформаційна система(ІС) повинна проводити підтримку проведення змагань зі спортивних змагань, а саме реєстрацію танцювальних пар, оцінювання танцювальної майстерності судьями та підраховувати результати турніру з можливістю друку.

1.2 Мета створення інформаційної системи

Автоматизація бізнес-процесів під час проведення змагань зі спортивних бальних танців.

1.3 Цільова аудиторія

До цільової аудиторії ІС можна віднести такі групи:

1. Адміністратор
2. Організатори турнірів
3. Судді змагань.

2 Вимоги до інформаційної системи

2.1 Вимоги до інформаційної системи в цілому

2.1.1 Вимоги до структури й функціонування інформаційної системи

Розроблювана інформаційна система має бути реалізована у вигляді десктопної програми зі зручним для використання інтерфейсом та функціоналом за клієнт-серверною архітектурою.

2.1.2 Вимоги до користувачів

Для використання даної ІС від користувачів не вимагаються спеціальні технічні навички, за винятком загальних навичок користування персональним комп'ютером.

2.1.3 Вимоги до збереження інформації

Користувач інформаційної системи, а саме адміністратор, має проводити резервне копіювання даних, для того, щоб уникнути зникнення або видалення потрібних для користування даних. Дана процедура буде здійснюватися вручну.

2.1.4 Вимоги до розмежування доступу

Користувачі ІС можна розділити на 3 групи:

1. Організатори турнірів;
2. Адміністратор;
3. Судді.

Організатори турнірів мають змогу переглядати результати змагань на вносити зміни за допомогою адміністратора.

Доступ до адміністративної частини мають користувачі із правами адміністратора. Доступ до адміністративної частини повинен здійснюватися з використанням унікального логіна й пароля.

Адміністратор може:

- проводити реєстрацію учасників танцювальних змагань;
- додавати користувачів із правами «Судді»;
- стежити за оцінюванням учасників судьями;
- підраховувати результати оцінювання;
- друкувати звіти(результати) змагань.

Доступ до суддівської частини мають користувачі із правами адміністратора та судді. Для того, щоб суддя отримав доступ до ІС, потрібно бути доданим адміністратором до суддівської колегії.

Суддя може:

- проводити оцінювання танцювальних пар.

2.2 Вимоги до програмного забезпечення

Головною вимогою є наявність ноутбуку та обов'язкове підключення до локального серверу через Інтернет.

Для роботи інформаційної системи необхідно мати наступні мінімальні системні вимоги:

- Операційна система Windows 7, 8, 10;
- Одноядерний процесор з тактовою частотою 1 ГГц;
- Оперативна пам'ять розміром 512 Мб.

3 Склад і зміст робіт зі створення ІС

Детальний опис етапів розроблення інформаційної системи підтримки проведення змагань зі спортивних бальних танців описано у 6 пункті Додатку Б.

4 Порядок контролю та приймання

Перевірка придатного та належного функціонування інформаційної системи здійснюється керівником дипломного проекту та незалежним користувачем з метою виявлення незручностей у використанні та можливих помилок.

Хід виконання дипломного проекту перевіряється за допомогою календарного плану, зазначеного у 6 пункті Додатку Б.

Додаток Б. Планування робіт

1. Деталізація мети інформаційної системи методом SMART.

Ціллю даної інформаційної системи є отримати інформаційну систему, що реалізує реєстрацію танцювальних пар, оцінювання їхньої майстерності судьями та отримання результатів змагань.

Узагальнений вигляд мети полягає у вивченні, як повинна працювати інформаційна система за загально прийнятою системою оцінювання «Скейтинг».

Загальний вигляд мети полягає в тому, щоб змагання проводилися за підтримки цієї інформаційної системи та, завдяки їй, полегшені в організації.

Реалізація інформаційної системи створена в форматі десктопної програми на основі мови програмування C# та баз даних MySQL за клієнт-серверною архітектурою.

Найбільш важливими результатами є:

- підключення до бази даних MySQL з MS Visual Studio 2010 Ultimate;
- паралельна робота інформаційної системи на ноутбуку суддів та адміністратора з постійним оновленням даних;
- повернення результату.

Інформаційну систему буде виконано вчасно, що підтверджується календарним планом.

2. Планування змісту структури робіт.

WBS-структурою, або структурною декомпозицією робіт (Work Breakdown Structure), є представлення інформаційної системи, яке виконане у вигляді ієрархічної структури робіт та досягається за допомогою послідовної декомпозиції. Дана структура детально планує та проводить оцінку вартості, визначає та розподіляє персональну відповідальність виконавців. На основі

робіт і результатів, що визначають зміст інформаційної системи було побудовано WBS-структуру, яку можна побачити на рис. Б.1.

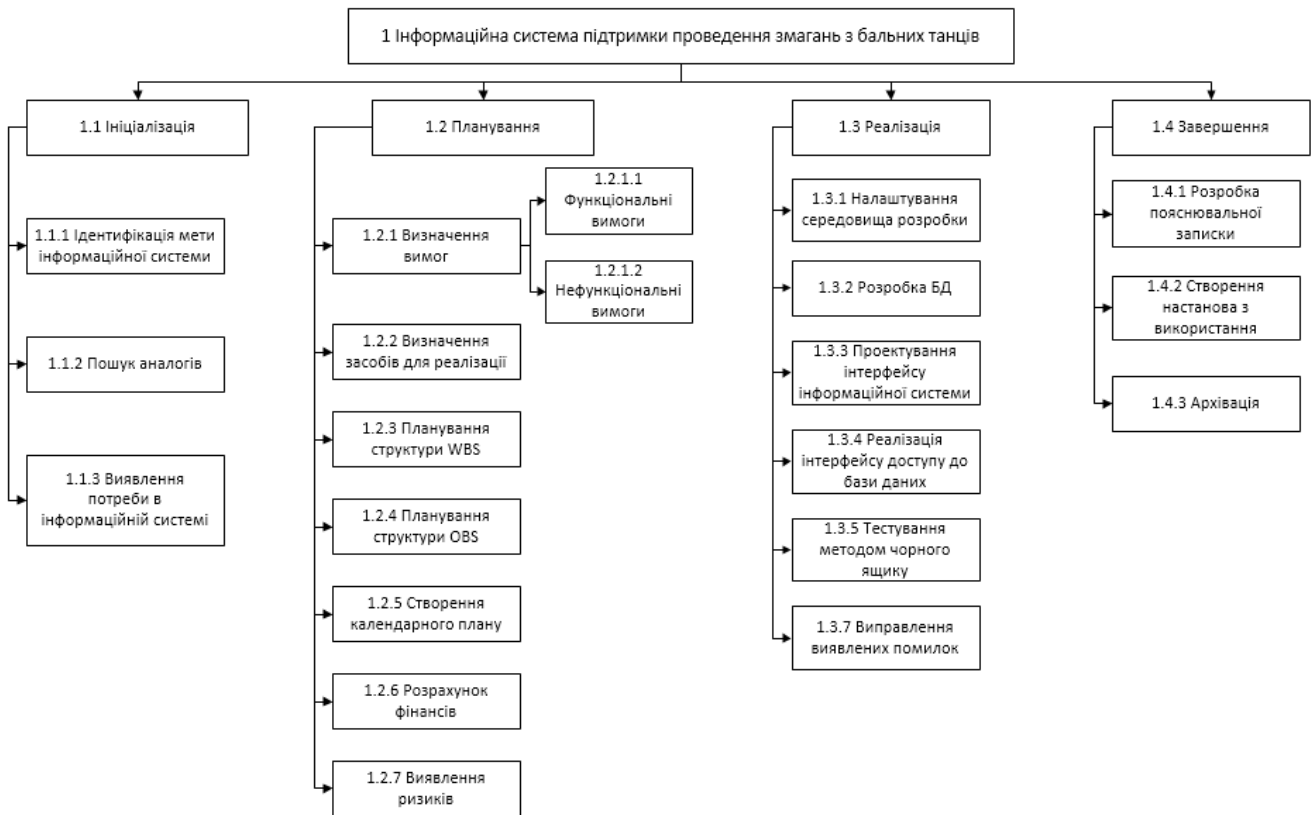


Рисунок Б.1 –WBS-структура системи

3. Планування структури виконавця для впровадження готової інформаційної системи

OBS-структура (рис. Б.2), або організаційна структура проекту (Organizational Breakdown Structure), представляє собою ієрархічну структуру управління інформаційною системою та показує відносини між її учасниками. За допомогою графічного відображення можна визначити учасників інформаційної системи (фізичних та юридичних осіб) та їхніх відповідальних осіб, що залучені до реалізації даної системи. На рис. Б.2 показано OBS-структуру.

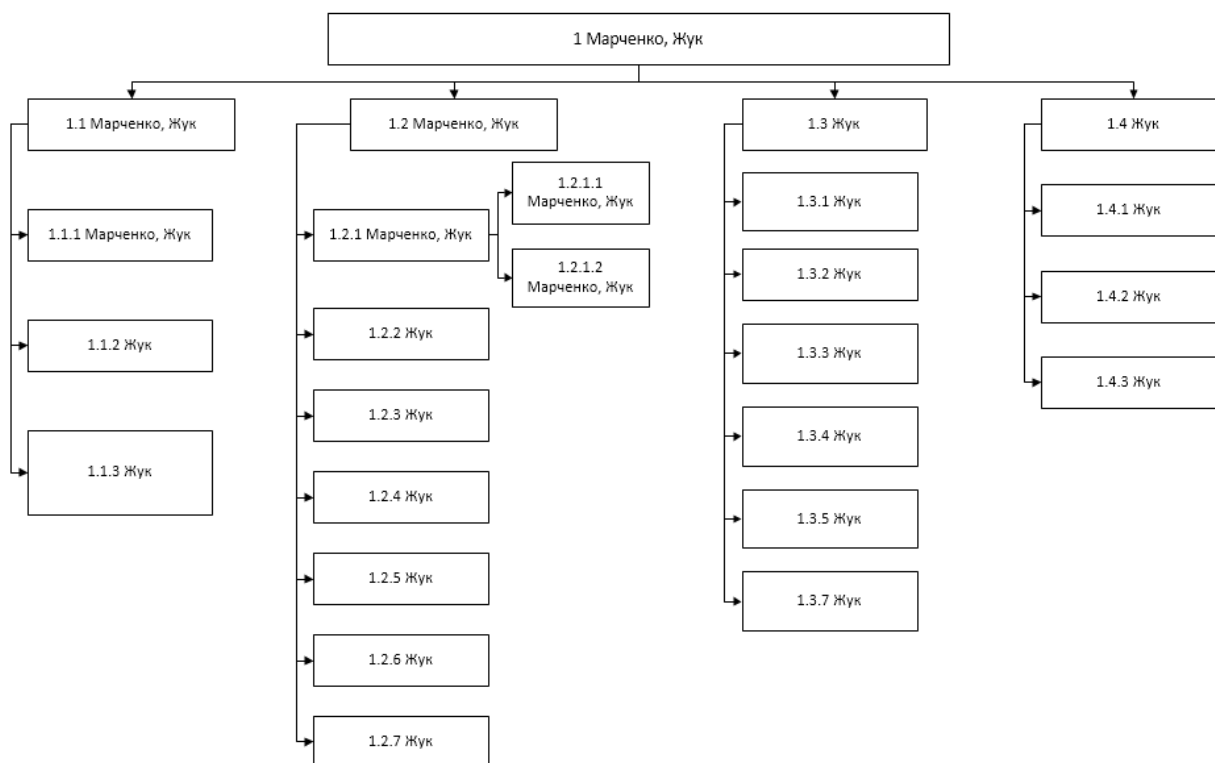


Рисунок Б.2 –OBS-структура системи

4. Побудова матриці відповідальності

Матриця відповідальності використовується для відображення зв'язків між пакетами робіт або операціями та членами команди. Матричний формат показує всі операції, які виконує одна людина або всіх людей, що задіяні в виконанні того чи іншого етапу створення інформаційної системи. А також забезпечує наділення відповідальності одній людині для виконання одного завдання задля попередження різних невідповідностей.

Для побудови матриці відповідальності у вигляді таблиці за основу були взяті OBS та WBS структури. У таблиці Б.1 наведено матрицю відповідальності.

Таблиця Б.1 – Матриця відповідальності

WBS\OBS	Жук	Марченко
1.1.1 Ідентифікація мети інформаційної системи		
1.1.2 Пошук аналогів		

Продовження таблиці Б.1 – Матриця відповідальності

WBS\OBS	Жук	Марченко
1.2.1 Визначення вимог		
1.2.1.1 Функціональні вимоги		
1.2.1.2 Нефункціональні вимоги		
1.2.2 Визначення засобів для реалізації		
1.2.3 Планування структури WBS		
1.2.4 Планування структури OBS		
1.2.5 Створення календарного плану		
1.2.6 Розрахунок фінансів		
1.2.7 Виявлення ризиків		
1.3.1 Налаштування середовища розробки		
1.3.2 Розробка БД		
1.3.3 Проектування інтерфейсу інформаційної системи		
1.3.4 Реалізація інтерфейсу доступу до інформаційної системи		
1.3.5 Тестування методом чорного ящика		
1.3.7 Виправлення виявлених помилок		
1.4.1 Розробка пояснювальної записки		
1.4.2 Створення настанови з використання		
1.4.3 Архівація		

5. Розробка PDM мережі

PDM мережа була побудована за допомогою програми MS Project 2016. Мережу зображено на рис. Б.3. Використання моделі інформаційної системи, побудованої в даному програмному середовищі, дозволяє контролювати і оптимізувати план виконання робіт, наочно відстежувати хід його виконання.

Метод побудови PDM мережі спрямовано на аналіз часу, потрібного для виконання кожного окремого етапу створення інформаційної системи, а також мінімального необхідного часу для виконання інформаційної системи. Мережу зображено на рис. Б.3.

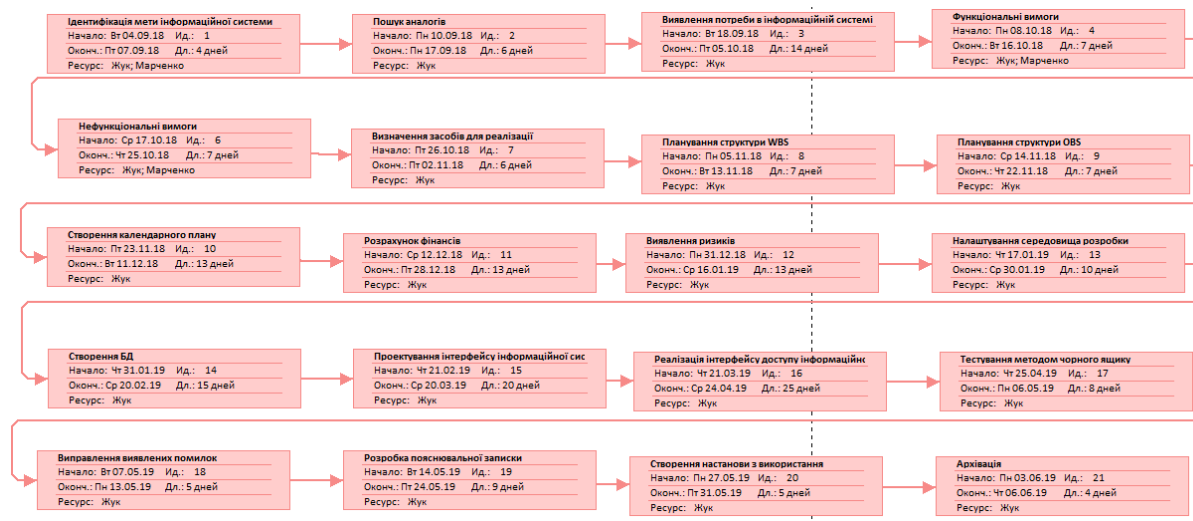


Рисунок Б.3 – PDM-мережа

6. Побудова календарного графіку виконання інформаційної системи

Для того щоб мати реальне уявлення про тривалість виконання окремих робіт з урахуванням обмежень у використанні ресурсів, а також тривалість в цілому з урахуванням вихідних та святкових днів, будують календарний графік робіт, що також називають діаграмою Ганта. Дану діаграму наведено на рисунку Б.4.

Назва задачі	Длительность	Начало	Окончание	Календарний графік робіт																
				Сен	Окт	Ноя	Дек	Янв	Фев	Мар	Апр	Май	Июн							
1 Ідентифікація мети інформаційної системи	4 дні	Вт 04.09.18	Пт 07.09.18	Жук, Марченко																
2 Пошук аналогів	6 днів	Пн 10.09.18	Пн 17.09.18	Жук																
3 Виявлення потреби в інформаційній системі	14 днів	Вт 18.09.18	Пт 05.10.18	Жук																
4 Функціональні вимоги	7 днів	Пн 08.10.18	Вт 16.10.18	Жук, Марченко																
5 Нефункціональні вимоги	7 днів	Ср 17.10.18	Чт 25.10.18	Жук, Марченко																
6 Визначення засобів для реалізації	6 днів	Пт 26.10.18	Пт 02.11.18	Жук																
7 Планування структури WBS	7 днів	Пн 05.11.18	Вт 13.11.18	Жук																
8 Планування структури OBS	7 днів	Ср 14.11.18	Чт 22.11.18	Жук																
9 Створення календарного плану	13 днів	Пт 23.11.18	Вт 11.12.18	Жук																
10 Розрахунок фінансів	13 днів	Ср 12.12.18	Пт 28.12.18	Жук																
11 Виявлення ризиків	13 днів	Пн 31.12.18	Ср 16.01.19	Жук																
12 Налаштування середовища розробки	10 днів	Чт 17.01.19	Ср 30.01.19	Жук																
13 Створення БД	15 днів	Чт 31.01.19	Ср 20.02.19	Жук																
14 Проектування інтерфейсу інформаційної системи	20 днів	Чт 21.02.19	Ср 20.03.19	Жук																
15 Реалізація інтерфейсу доступу інформаційної системи	25 днів	Чт 21.03.19	Ср 24.04.19	Жук																
16 Тестування методом чорного ящика	8 днів	Чт 25.04.19	Пн 06.05.19	Жук																
17 Тестування методом чорного ящика	8 днів	Чт 25.04.19	Пн 06.05.19	Жук																
17 Виправлення виявлених помилок	5 днів	Вт 07.05.19	Пн 13.05.19	Жук																
18 Розробка пояснювальної записки	9 днів	Вт 14.05.19	Пт 24.05.19	Жук																
19 Створення настанови з використанням	5 днів	Пн 27.05.19	Пт 31.05.19	Жук																
20 Архівація	4 днів	Пн 03.06.19	Чт 06.06.19	Жук																

Рисунок Б.4 – Календарний графік робіт

7. Управління ризиками інформаційної системи

Перед початком реалізації інформаційної системи необхідно передбачити усі можливі ризики, що можуть виникнути та вплинути на якість та час розробки даної системи.

Було виявлено такі ризики:

- R1 – зміна ТЗ на етапі розробки;
- R2 – не виявлені помилки на етапі розробки інформаційної системи;
- R3 – порушено дотримання календарного плану;
- R4 – несанкціонований доступ до локального серверу;
- R5 – хвороба розробника;
- R6 – некоректне тестування;
- R7 – некоректна робота інформаційної системи.

За допомогою оцінки були визначені ймовірності появи усіх можливих ризиків. Згідно з цим було побудовано таблицю Б.2.

Таблиця Б.2 – Ймовірність виникнення ризиків

Ймовірність виникнення	R1	R2	R3	R4	R5	R6	R7
Слабо ймовірний							
Малоймовірний							
Ймовірний							
Дуже ймовірний							
Майже можливий							

Наступним кроком було зроблено таблицю тих втрат, які можуть з'явитися в процесі проектування та розробки при виникненні ризиків (таблиця Б.3).

Таблиця Б.3 – Втрати при виникненні ризиків

Значимість впливу	R1	R2	R3	R4	R5	R6	R7
мінімальна							
низька							
середня							

Продовження таблиці Б.3 – Втрати при виникненні ризиків

Значимість впливу	R1	R2	R3	R4	R5	R6	R7
висока							
максимальна							

Після побудови двох таблиць була створена матриця впливу (ймовірностей і наслідків) ризиків, де білим кольором позначено неважливі ризики, темнішим – помірні, дуже темним – критичні.

Таблиця Б.4 – Матриця ймовірність-втрати

Ймовірність		R2	
		R4	R6
	R3		
		R1	R5
	R7		
Вплив			

Виходячи з цього, було визначено три критичних ризики, такі як:

- R2 – не виявлені помилки на етапі розробки інформаційної системи;
- R4 – несанкціонований доступ до локального серверу;
- R6 – некоректне тестування;

Виникнення першого критичного ризику можна уникнути лише за допомогою розробника, а саме через зміну програмного коду. Другий критичний ризик малоймовірний, але при його виникненні потрібне буде втручання розробника, який зможе налагодити доступ до локального серверу. Третій критичний ризик, а саме некоректне тестування, може створити проблеми у подальшому використанні вже реалізованої інформаційної системи.

8. Формування бюджету інформаційної системи

Завершенням планування інформаційної системи є етап розподілу бюджету. Перед тим, як визначити бюджет, необхідно виділити тих виконавців, які брали участь в даній роботі. Були визначені такі працівники як:

- Розробник (Жук);
- Керівник (Марченко);

Бюджет – це загальні майбутні витрати, які необхідні безпосередньо для створення інформаційної системи. Тобто це витрати на фінансування всіх робіт, передбачених WBS-структурою. Витрати розподіляються у часі на основі календарного плану реалізації робіт. Визначення бюджету на витрачання заробітної плати (таблиця Б.5).

Таблиця Б.5 – Бюджет на заробітну плату

Посада	За 1 год	Робочих годин	Сума
Розробник	170	198	33660
Керівник	165	18	2970

Отже, бюджет заробітної плати складає *36630 грн.*

Також на менеджмент інформаційної системи було додано 20% з загального бюджету:

$$36630 + 36630 * 0.2 = 43956$$

Додаток В. Лістинг коду

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace ballroom
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new LogIn());
        }
    }
}
```

LogIn.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using System.Threading;

namespace ballroom
{
    public partial class LogIn : Form
    {
        public LogIn()
        {
            Thread t = new Thread(new ThreadStart(StartForm));
            t.Start();
            Thread.Sleep(5000);
            InitializeComponent();
        }
    }
}
```

```

        t.Abort();
    }

    public void StartForm()
    {
        Application.Run(new Start());
    }

    private void button1_Click(object sender, EventArgs e)
    {
        Form ifrm = new Admin();
        ifrm.StartPosition = FormStartPosition.CenterScreen;
        ifrm.Show();
        this.Hide();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        Form ifrm = new Judge();
        ifrm.StartPosition = FormStartPosition.CenterScreen;
        ifrm.Show();
        this.Hide();
    }

    private void LogIn_FormClosed(object sender, FormClosedEventArgs e)
    {
        Application.Exit();
    }
}

```

Admin.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace ballroom
{
    public partial class Admin : Form
    {
        public Admin()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if ((textBox1.Text == "admin") && (textBox2.Text == "admin"))
            {
                Form form = new Form1();
                form.StartPosition = FormStartPosition.CenterScreen;
            }
        }
    }
}

```

```

        form.Show();
        this.Hide();
    }
    else if ((textBox1.Text == "") || (textBox2.Text == ""))
    {
        MessageBox.Show("Заповніть усі поля!", "Помилка!", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        textBox1.Text = "";
        textBox2.Text = "";
    }

    else
    {
        MessageBox.Show("Логін/Пароль введено невірно!", "Помилка!", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        textBox1.Text = "";
        textBox2.Text = "";
    }
}

private void Admin_FormClosed(object sender, FormClosedEventArgs e)
{
    Form ifrm = Application.OpenForms[0];
    ifrm.StartPosition = FormStartPosition.CenterScreen;
    ifrm.Show();
}
}
}

```

Judge.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using MySql.Data;
using MySql.Data.MySqlClient;

namespace ballroom
{
    public partial class Judge : Form
    {
        public Judge()
        {
            InitializeComponent();
        }

        string connStr = "server=localhost;user=root;password=;database=ballroom;port=3306;";
        int i;
        public static string judge;
        private void button1_Click(object sender, EventArgs e)
        {

```

```

if ((comboBox1.SelectedIndex != -1) && (password.Text != ""))
{
    i = 0;
    MySqlConnection con = new MySqlConnection(connStr);
    con.Open();
    MySqlCommand cmd = con.CreateCommand();
    cmd.CommandType = CommandType.Text;
    string data = comboBox1.Text.ToString();
    string[] words = data.Split(' ');

    cmd.CommandText = "SELECT * FROM judge j WHERE j.judgeSurname=" + words[0] + " AND
j.judgeName=" + words[2] + " AND j.password=" + password.Text + """;

    cmd.ExecuteNonQuery();
    DataTable dt = new DataTable();
    MySqlDataAdapter da = new MySqlDataAdapter(cmd);
    da.Fill(dt);
    i = Convert.ToInt32(dt.Rows.Count.ToString());

    if (i == 0)
    {
        MessageBox.Show("Пароль введено невірно!", "Помилка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        comboBox1.SelectedIndex = -1;
        password.Text = "";
    }

    else
    {
        judge = comboBox1.Text.ToString();
        Form form = new Form2();
        form.StartPosition = FormStartPosition.CenterScreen;
        form.Show();
        this.Hide();
    }

    con.Close();
}
else
{
    MessageBox.Show("Заповніть усі поля!", "Помилка!", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    comboBox1.SelectedIndex = -1;
    password.Text = "";
}

}

private void Judge_Load(object sender, EventArgs e)
{
    MySqlConnection conn = new MySqlConnection(connStr);
    MySqlCommand command = new MySqlCommand();

    string commandString = "SELECT judgeSurname, judgeName FROM judge";

    command.CommandText = commandString;
    command.Connection = conn;
    MySqlDataReader reader;
    try
    {
        command.Connection.Open();
        reader = command.ExecuteReader();
    }
}

```



```

        int i = 0;
        comboBox1.Items.Clear();
        while (reader.Read())
        {
            comboBox1.Items.Add(reader["judgeSurname"] + " " + reader["judgeName"]);
            i += 1;
        }

        reader.Close();
    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.ToString());
    }

    finally
    {
        command.Connection.Close();
    }
}

private void Judge_FormClosed(object sender, FormClosedEventArgs e)
{
    Form ifrm = Application.OpenForms[0];
    ifrm.StartPosition = FormStartPosition.CenterScreen;
    ifrm.Show();
}
}
}

```

Form1.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using MySql.Data;
using MySql.Data.MySqlClient;

namespace ballroom
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        string connStr = "server=localhost;user=root;password=;database=ballroom;port=3306;";
    }
}

```

```

private void Form1_Load(object sender, EventArgs e)
{
    //заповнення списку категорій, де є зареєстровані учасники
    load();
}

private void load()
{
    MySqlConnection conn = new MySqlConnection(connStr);
    MySqlCommand command = new MySqlCommand();

    string commandString = "SELECT cat.category_name, c.class_name, p.program FROM category cat, class
c, program p, category_class_program ccp, registration reg WHERE p.program_id=ccp.program_id AND
cat.category_id=ccp.category_id AND c.class_id=ccp.class_id AND ccp.c_c_p_id=reg.c_c_p_id GROUP BY
ccp.c_c_p_id";

    command.CommandText = commandString;
    command.Connection = conn;
    MySqlDataReader reader;
    try
    {
        command.Connection.Open();
        reader = command.ExecuteReader();
        int i = 0;
        comboBox1.Items.Clear();
        while (reader.Read())
        {
            comboBox1.Items.Add(reader["category_name"] + " " + reader["class_name"] + " " +
reader["program"]);
            i += 1;
        }

        reader.Close();
    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.ToString());
    }

    finally
    {
        command.Connection.Close();
    }
}

private void NewRegToolStripMenuItem_Click(object sender, EventArgs e)
{
    //перехід до форми реєстрації
    Form form = new Registration();
    form.StartPosition = FormStartPosition.CenterScreen;
    form.Show();
    this.Hide();
}

private void Reg_printToolStripMenuItem_Click(object sender, EventArgs e)
{
    //перехід до форми друку зареєстрованих учасників у певній категорії
    Form form = new Reg_print();
}

```

```

        form.StartPosition = FormStartPosition.CenterScreen;
        form.Show();
        this.Hide();
    }

    private void AddToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //перехід до форми додавання суддів
        Form form = new New_judge();
        form.StartPosition = FormStartPosition.CenterScreen;
        form.Show();
        this.Hide();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        MySqlConnection conn2 = new MySqlConnection(connStr);
        MySqlCommand command2 = new MySqlCommand();

        string commandString2 = "SELECT d.dance_id FROM dances d WHERE d.dance_name LIKE '" +
checkedListBox1.SelectedItem.ToString() + "%'";

        command2.CommandText = commandString2;
        command2.Connection = conn2;

        MySqlDataReader reader2;
        int dance;
        try
        {
            command2.Connection.Open();
            reader2 = command2.ExecuteReader();
            while (reader2.Read())
            {
                dance = Convert.ToInt32(reader2["dance_id"]);
            }
        }

        MySqlConnection conn = new MySqlConnection(connStr);
        MySqlCommand command = new MySqlCommand();

        string data = comboBox1.Text.ToString();
        string[] words = data.Split(' ');
        string commandString = "SELECT ccp.c_c_p_id FROM category cat, class c, program p,
category_class_program ccp WHERE p.program_id=ccp.program_id AND cat.category_id=ccp.category_id AND
c.class_id=ccp.class_id AND cat.category_name = '" + words[0] + "' AND c.class_name LIKE '" + words[2] + '%"
AND p.program LIKE '" + words[4] + "%'";
        command.CommandText = commandString;
        command.Connection = conn;

        MySqlDataReader reader;

        try
        {
            command.Connection.Open();
            reader = command.ExecuteReader();

            using (MySqlConnection connect = new MySqlConnection(connStr))
            {
                connect.Open();
                while (reader.Read())
                {

```

```

int num = Convert.ToInt32(reader["c_c_p_id"]);

string strSQL = "INSERT INTO `log` (`id`, `time`, `c_c_p_id`, `dance_id`) VALUES
(NULL, NOW(), " + num + ", " + dance + ")";

try
{
    using (MySqlConnection connection = new MySqlConnection(connStr))
    {
        connection.Open();
        MySqlCommand cp = new MySqlCommand(strSQL, connection);

        if (cp.ExecuteNonQuery() == 1)
        {
            MessageBox.Show("OK", "Інформація", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }
        connection.Close();
    }
}
catch (MySqlException ex)
{
    MessageBox.Show(ex.Message);
}
}
reader.Close();

}

catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{
    command.Connection.Close();
}

}
reader2.Close();

}
catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

}

finally
{
    command2.Connection.Close();
}

}

//кількість суддів, що оцінили танець
A: MySqlConnection conn3 = new MySqlConnection(connStr);

```

```

MySQLCommand command3 = new MySQLCommand();

string commandString3 = "SELECT id FROM log l ORDER BY id DESC limit 1 ";

command3.CommandText = commandString3;
command3.Connection = conn3;
MySQLDataReader reader3;
try
{
    command3.Connection.Open();
    reader3 = command3.ExecuteReader();
    while (reader3.Read())
    {
        int id=Convert.ToInt32(reader3["id"]);

        MySqlConnection connect = new MySqlConnection(connStr);
        MySQLCommand cmd = new MySQLCommand();

        string cmdString = "SELECT COUNT(*) as rez FROM results WHERE id="+id+"";

        cmd.CommandText = cmdString;
        cmd.Connection = connect;

        MySQLDataReader reader4;

        try
        {
            cmd.Connection.Open();
            reader4 = cmd.ExecuteReader();

            while(reader4.Read())
            {
                int rez= Convert.ToInt32(reader4["rez"]);

                MySqlConnection connect2 = new MySqlConnection(connStr);
                MySQLCommand cmd2 = new MySQLCommand();

                string cmdString2 = "SELECT COUNT(*) as judges FROM judge";

                cmd2.CommandText = cmdString2;
                cmd2.Connection = connect2;

                MySQLDataReader reader5;

                try
                {
                    cmd2.Connection.Open();
                    reader5 = cmd2.ExecuteReader();

                    while(reader5.Read())
                    {
                        int judge=Convert.ToInt32(reader5["judges"]);

                        MySqlConnection connect3 = new MySqlConnection(connStr);
                        MySQLCommand cmd3 = new MySQLCommand();

                        string cmdString3 = "SELECT COUNT(*) as members FROM (SELECT COUNT(*) FROM results
r GROUP BY r.number) as rez";

                        cmd3.CommandText = cmdString3;
                        cmd3.Connection = connect3;

```

```

MySQLDataReader reader6;

try
{
    cmd3.Connection.Open();
    reader6 = cmd3.ExecuteReader();

    while(reader6.Read())
    {
        int m_p_j=Convert.ToInt32(reader6["members"]);

        if (rez == judge * m_p_j)
        {
            MessageBox.Show("Оберіть наступний танець/категорію!", "Інформація",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            reader6.Close();
            reader5.Close();
            reader4.Close();
            reader3.Close();
            cmd3.Connection.Close();
            cmd2.Connection.Close();
            cmd.Connection.Close();
            command3.Connection.Close();
            goto A;
        }
    }

    reader6.Close();
}

catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{
    cmd3.Connection.Close();
}

}

reader5.Close();
}
catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

}

finally
{
    cmd2.Connection.Close();
}
}
reader4.Close();
}
catch (MySqlException ex)

```

```

        {
            Console.WriteLine(ex.ToString());
        }

        finally
        {
            cmd.Connection.Close();
        }
    }

    reader3.Close();

}
catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{
    command3.Connection.Close();
}

//перехід до форми оцінювання фіналу
//Form form = new Final();
//form.Left = this.Left;
//form.Top = this.Top;
//form.Show();
//this.Hide();
}

private void comboBox1_SelectedValueChanged(object sender, EventArgs e)
{
    //присвоєння змінній назви категорії

    checkedListBox1.Visible = true;
    MySqlConnection conn = new MySqlConnection(connStr);
    MySqlCommand command = new MySqlCommand();

    string data = comboBox1.Text.ToString();
    string[] words = data.Split(' ');
    string commandString = "SELECT ccp.c_c_p_id FROM category cat, class c, program p,
category_class_program ccp WHERE p.program_id=ccp.program_id AND cat.category_id=ccp.category_id AND
c.class_id=ccp.class_id AND cat.category_name = '" + words[0] + "' AND c.class_name LIKE '" + words[2] + '%"
AND p.program LIKE '" + words[4] + "%'";
    command.CommandText = commandString;
    command.Connection = conn;

    MySqlDataReader reader;

    try
    {
        command.Connection.Open();
        reader = command.ExecuteReader();

        using (MySqlConnection connect = new MySqlConnection(connStr))
        {
            connect.Open();

```

```

        while (reader.Read())
        {
            int num = Convert.ToInt32(reader["c_c_p_id"]);
            MySqlConnection conn1 = new MySqlConnection(connStr);
            MySqlCommand command1 = new MySqlCommand();
            string strSQL="SELECT d.dance_name FROM dances d, category_class_program_dances
ccpd, category_class_program ccp WHERE ccp.c_c_p_id=ccpd.c_c_p_id AND d.dance_id=ccpd.dance_id AND
ccp.c_c_p_id = " + num + """;
            command1.CommandText = strSQL;
            command1.Connection = conn1;

            MySqlDataReader reader1;
            try
            {
                command1.Connection.Open();
                reader1 = command1.ExecuteReader();
                int i = 0;
                checkedListBox1.Items.Clear();
                while (reader1.Read())
                {
                    checkedListBox1.Items.Add(reader1["dance_name"]);
                    i += 1;
                }

                reader1.Close();
            }
            catch (MySqlException ex)
            {
                Console.WriteLine(ex.ToString());
            }

            finally
            {
                command1.Connection.Close();
            }
        }
    }
}
catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{
    command.Connection.Close();
}

}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    Form ifrm = Application.OpenForms[0];
    ifrm.StartPosition = FormStartPosition.CenterScreen;
    ifrm.Show();
}

private void checkedListBox1_ItemCheck(object sender, ItemCheckEventArgs e)
{
    if (e.NewValue == CheckState.Checked)

```



```

        for (int i = 0; i < checkedListBox1.Items.Count; ++i)
            if (e.Index != i) checkedListBox1.SetItemChecked(i, false);
    }

    private void checkedListBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        button1.Enabled = true;
    }

    private void refresh_Click(object sender, EventArgs e)
    {
        load();
    }

    private void PrintToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Form form = new Print();
        form.StartPosition = FormStartPosition.CenterScreen;
        form.Show();
        this.Hide();
    }
}
}

```

Form2.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using System.Threading;

using MySql.Data;
using MySql.Data.MySqlClient;

namespace ballroom
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        string connStr = "server=localhost;user=root;password=;database=ballroom;port=3306;";

        private void Form2_Load(object sender, EventArgs e)
        {

```

```

        timer1.Interval = 1000;
        timer1.Start();
    }

    int flag = 0;
    public void start()
    {

        MySqlConnection con = new MySqlConnection(connStr);
        con.Open();
        MySqlCommand cmd = con.CreateCommand();
        cmd.CommandType = CommandType.Text;

        cmd.CommandText = "SELECT * FROM log";

        cmd.ExecuteNonQuery();
        DataTable dt = new DataTable();
        MySqlDataAdapter da = new MySqlDataAdapter(cmd);
        da.Fill(dt);
        progressBar1.Maximum = Convert.ToInt32(dt.Rows.Count.ToString());
        if (flag < Convert.ToInt32(dt.Rows.Count.ToString()))
        {
            flag++;
            timer1.Start();
            progressBar1.Value++;
        }
        else
        {

            Form form = new Final();
            form.StartPosition = FormStartPosition.CenterScreen;
            form.Show();
            this.Hide();
        }
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        timer1.Stop();
        start();
    }

    private void pictureBox2_Click(object sender, EventArgs e)
    {
        Form ifrm = Application.OpenForms[0];
        ifrm.StartPosition = FormStartPosition.CenterScreen;
        ifrm.Show();
        this.Close();
    }
}
}

```

Registration.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;

```

```

using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using MySql.Data;
using MySql.Data.MySqlClient;

namespace ballroom
{
    public partial class Registration : Form
    {
        public Registration()
        {
            InitializeComponent();
        }

        string connStr = "server=localhost;user=root;password=;database=ballroom;port=3306;";

        private void Form1_Load(object sender, EventArgs e)
        {
            MySqlConnection conn = new MySqlConnection(connStr);
            MySqlCommand command = new MySqlCommand();

            string commandString = "SELECT c.city FROM city c";

            command.CommandText = commandString;
            command.Connection = conn;
            MySqlDataReader reader;
            try
            {
                command.Connection.Open();
                reader = command.ExecuteReader();
                int i = 0;
                while (reader.Read())
                {
                    string city = Convert.ToString(reader["city"]);
                    cityy.Items.Add(city);
                    i += 1;
                }

                reader.Close();
            }
            catch (MySqlException ex)
            {
                Console.WriteLine(ex.ToString());
            }

            finally
            {
                command.Connection.Close();
            }
        }

        private void boyBirth_ValueChanged(object sender, EventArgs e)
        {

```

```

DateTime dateNow = DateTime.Now;
int boyage = dateNow.Year - boyBirth.Value.Year;
if (dateNow.Month < boyBirth.Value.Month ||
    (dateNow.Month == boyBirth.Value.Month && dateNow.Day < boyBirth.Value.Day)) boyage--;

MySQLConnection conn = new MySqlConnection(connStr);
MySQLCommand command = new MySQLCommand();

if (solo.Checked == true)
{
    string commandString = "SELECT cat.category_name, c.class_name, p.program FROM category cat,
class c, program p, category_class_program ccp WHERE p.program_id=ccp.program_id AND
cat.category_id=ccp.category_id AND c.class_id=ccp.class_id AND " + boyage + " BETWEEN cat.min_age AND
cat.max_age AND cat.category_name LIKE '%соло%'";

    command.CommandText = commandString;
    command.Connection = conn;
    MySQLDataReader reader;
    try
    {
        command.Connection.Open();
        reader = command.ExecuteReader();
        int i = 0;
        checkedListBox1.Items.Clear();
        while (reader.Read())
        {
            checkedListBox1.Items.Add(reader["category_name"] + " " + reader["class_name"] + " " +
reader["program"]);
            i += 1;
        }

        reader.Close();
    }
    catch (MySQLException ex)
    {
        Console.WriteLine(ex.ToString());
    }

    finally
    {
        command.Connection.Close();
    }
}
else
{
    string commandString = "SELECT cat.category_name, c.class_name, p.program FROM category cat,
class c, program p, category_class_program ccp WHERE p.program_id=ccp.program_id AND
cat.category_id=ccp.category_id AND c.class_id=ccp.class_id AND " + boyage + " BETWEEN cat.min_age AND
cat.max_age AND cat.category_name NOT LIKE '%соло%'";

    command.CommandText = commandString;
    command.Connection = conn;
    MySQLDataReader reader;
    try
    {
        command.Connection.Open();
        reader = command.ExecuteReader();
        int i = 0;
        checkedListBox1.Items.Clear();
        while (reader.Read())

```

```

        {
            checkedListBox1.Items.Add(reader["category_name"] + " " + reader["class_name"] + " " +
reader["program"]);
            i += 1;
        }

        reader.Close();
    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.ToString());
    }

    finally
    {
        command.Connection.Close();
    }
}
checkedListBox1.Enabled = true;

}

private void girlBirth_ValueChanged(object sender, EventArgs e)
{
    DateTime dateNow = DateTime.Now;
    if (boyBirth.Value.Date != dateNow.Date)
    {
        checkedListBox1.Items.Clear();
        checkedListBox1.Enabled = false;

        int boyage = dateNow.Year - boyBirth.Value.Year;
        if (dateNow.Month < boyBirth.Value.Month ||
            (dateNow.Month == boyBirth.Value.Month && dateNow.Day < boyBirth.Value.Day)) boyage--;
        int girlage = dateNow.Year - girlBirth.Value.Year;
        if (dateNow.Month < girlBirth.Value.Month ||
            (dateNow.Month == girlBirth.Value.Month && dateNow.Day < girlBirth.Value.Day)) girlage--;

        MySqlConnection conn = new MySqlConnection(connStr);
        MySqlCommand command = new MySqlCommand();

        string commandString = "SELECT cat.category_name, c.class_name, p.program FROM category cat,
class c, program p, category_class_program ccp WHERE p.program_id=ccp.program_id AND
cat.category_id=ccp.category_id AND c.class_id=ccp.class_id AND GREATEST(" + boyage + ", " + girlage + ")
BETWEEN cat.min_age AND cat.max_age AND cat.category_name NOT LIKE '%соло%'";

        command.CommandText = commandString;
        command.Connection = conn;
        MySqlDataReader reader;
        try
        {
            command.Connection.Open();
            reader = command.ExecuteReader();
            int i = 0;
            checkedListBox1.Items.Clear();
            while (reader.Read())
            {
                checkedListBox1.Items.Add(reader["category_name"] + " " + reader["class_name"] + " " +
reader["program"]);
                i += 1;
            }
        }
    }
}

```

```

        reader.Close();
    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.ToString());
    }

    finally
    {
        command.Connection.Close();
    }

    checkedListBox1.Enabled = true;
}
else
{
    int girlage = dateNow.Year - girlBirth.Value.Year;
    if (dateNow.Month < girlBirth.Value.Month ||
        (dateNow.Month == girlBirth.Value.Month && dateNow.Day < girlBirth.Value.Day)) girlage--;

    MySqlConnection conn = new MySqlConnection(connStr);
    MySqlCommand command = new MySqlCommand();

    if (solo.Checked == true)
    {
        string commandString = "SELECT cat.category_name, c.class_name, p.program FROM category cat,
class c, program p, category_class_program ccp WHERE p.program_id=ccp.program_id AND
cat.category_id=ccp.category_id AND c.class_id=ccp.class_id AND " + girlage + " BETWEEN cat.min_age AND
cat.max_age AND cat.category_name LIKE '%соло%'";

        command.CommandText = commandString;
        command.Connection = conn;
        MySqlDataReader reader;
        try
        {
            command.Connection.Open();
            reader = command.ExecuteReader();
            int i = 0;
            checkedListBox1.Items.Clear();
            while (reader.Read())
            {
                checkedListBox1.Items.Add(reader["category_name"] + " " + reader["class_name"] + " " +
reader["program"]);
                i += 1;
            }

            reader.Close();
        }
        catch (MySqlException ex)
        {
            Console.WriteLine(ex.ToString());
        }

        finally
        {
            command.Connection.Close();
        }
    }

}
else

```

```

    {
        string commandString = "SELECT cat.category_name, c.class_name, p.program FROM category cat,
class c, program p, category_class_program ccp WHERE p.program_id=ccp.program_id AND
cat.category_id=ccp.category_id AND c.class_id=ccp.class_id AND '" + girlage + "' BETWEEN cat.min_age AND
cat.max_age AND cat.category_name NOT LIKE '%соло%'";

        command.CommandText = commandString;
        command.Connection = conn;
        MySqlDataReader reader;
        try
        {
            command.Connection.Open();
            reader = command.ExecuteReader();
            int i = 0;
            checkedListBox1.Items.Clear();
            while (reader.Read())
            {
                checkedListBox1.Items.Add(reader["category_name"] + " " + reader["class_name"] + " " +
reader["program"]);
                i += 1;
            }

            reader.Close();
        }
        catch (MySqlException ex)
        {
            Console.WriteLine(ex.ToString());
        }

        finally
        {
            command.Connection.Close();
        }

        checkedListBox1.Enabled = true;
    }
}

private void addButton_Click(object sender, EventArgs e)
{
    if (numberBox.Text != "" && club.Text != "" && cityy.Text != "" && trainer1.Text != "" && trainer2.Text
!= "" && checkedListBox1.CheckedItems != null && (boyBirth.Enabled == true || girlBirth.Enabled ==true))
    {

        DateTime dateNow = DateTime.Now;
        MySqlConnection conn2 = new MySqlConnection(connStr);
        MySqlCommand command2 = new MySqlCommand();

        string commandString2 = "SELECT c.city_id FROM city c WHERE c.city LIKE '" + cityy.Text + "%'";

        command2.CommandText = commandString2;
        command2.Connection = conn2;

        MySqlDataReader reader2;
        int id;
        try
        {
            command2.Connection.Open();

```

```

reader2 = command2.ExecuteReader();
while (reader2.Read())
{
    id = Convert.ToInt32(reader2["city_id"]);

    MySqlConnection conn = new MySqlConnection(connStr);
    MySqlCommand command = new MySqlCommand();

    foreach (object itemChecked in checkedListBox1.CheckedItems)
    {
        string data = itemChecked.ToString();
        string[] words = data.Split(' ');
        string commandString = "SELECT ccp.c_c_p_id FROM category cat, class c, program p,
category_class_program ccp WHERE p.program_id=ccp.program_id AND cat.category_id=ccp.category_id AND
c.class_id=ccp.class_id AND cat.category_name LIKE '" + words[0] + "%' AND c.class_name LIKE '" + words[2]
+ "%' AND p.program LIKE '" + words[4] + "%'";
        command.CommandText = commandString;
        command.Connection = conn;

        MySqlDataReader reader;

        try
        {
            command.Connection.Open();
            reader = command.ExecuteReader();

            using (MySqlConnection connect = new MySqlConnection(connStr))
            {
                connect.Open();

                while (reader.Read())
                {
                    int num = Convert.ToInt32(reader["c_c_p_id"]);

                    if (solo.Checked == true)
                    {
                        if (boyName.Text != null && boySurname.Text != null && boyBirth.Value.Date !=
dateNow.Date)
                            {
                                string strSQL = "INSERT INTO `registration` (`reg_id`, `number`, `surname`, `name`,
`gender`, `club`, `city_id`, `trainer1`, `trainer2`, `c_c_p_id`) VALUES (NULL, '" +
Convert.ToInt32(numberBox.Text) + "', '" + Convert.ToString(boySurname.Text) + "', '" +
Convert.ToString(boyName.Text) + "', 'партнер', '" + Convert.ToString(club.Text) + "', '" + id + "', '" +
Convert.ToString(trainer1.Text) + "', '" + Convert.ToString(trainer2.Text) + "', '" + num + "')";

                                try
                                {
                                    using (MySqlConnection connection = new MySqlConnection(connStr))
                                    {
                                        connection.Open();
                                        MySqlCommand cp = new MySqlCommand(strSQL, connection);

                                        if (cp.ExecuteNonQuery() == 1)
                                        {
                                            MessageBox.Show("Реєстрація за категорією №" + num + " пройшла
успішно!", "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);

```



```

        MessageBox.Show("Учасника " + boySurname.Text + " " + boyName.Text +
" додано!", "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    connection.Close();
}
}

catch (MySqlException ex)
{
    MessageBox.Show(ex.Message);
}

else if (girlName.Text != null && girlSurname.Text != null && girlBirth.Value.Date !=
dateNow.Date)
{
    string strSQL = "INSERT INTO `registration` (`reg_id`, `number`, `surname`, `name`,
`gender`, `club`, `city_id`, `trainer1`, `trainer2`, `c_c_p_id`) VALUES (NULL, "" +
Convert.ToInt32(numberBox.Text) + "", "" + Convert.ToString(girlSurname.Text) + "", "" +
Convert.ToString(girlName.Text) + "", 'партнерка', "" + Convert.ToString(club.Text) + "", "" + id + "", "" +
Convert.ToString(trainer1.Text) + "", "" + Convert.ToString(trainer2.Text) + "", "" + num + "");

    try
    {

        using (MySqlConnection connection = new MySqlConnection(connStr))
        {
            connection.Open();
            MySqlCommand cp = new MySqlCommand(strSQL, connection);

            if (cp.ExecuteNonQuery() == 1)
            {
                MessageBox.Show("Реєстрація за категорією №" + num + " пройшла
успішно!", "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);
                MessageBox.Show("Учасника " + girlSurname.Text + " " + girlName.Text +
" додано!", "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            connection.Close();
        }
    }

    catch (MySqlException ex)
    {
        MessageBox.Show(ex.Message);
    }

}
}
else
{
    string strSQL = "INSERT INTO `registration` (`reg_id`, `number`, `surname`, `name`,
`gender`, `club`, `city_id`, `trainer1`, `trainer2`, `c_c_p_id`) VALUES (NULL, "" +
Convert.ToInt32(numberBox.Text) + "", "" + Convert.ToString(boySurname.Text) + "", "" +
Convert.ToString(boyName.Text) + "", 'партнер', "" + Convert.ToString(club.Text) + "", "" + id + "", "" +
Convert.ToString(trainer1.Text) + "", "" + Convert.ToString(trainer2.Text) + "", "" + num + ""), (NULL, "" +
Convert.ToInt32(numberBox.Text) + "", "" + Convert.ToString(girlSurname.Text) + "", "" +
Convert.ToString(girlName.Text) + "", 'партнерка', "" + Convert.ToString(club.Text) + "", "" + id + "", "" +
Convert.ToString(trainer1.Text) + "", "" + Convert.ToString(trainer2.Text) + "", "" + num + "");

    try

```

```

    {
        using (SqlConnection connection = new SqlConnection(connStr))
        {
            connection.Open();
            MySqlCommand cp = new MySqlCommand(strSQL, connection);

            if (cp.ExecuteNonQuery() == 2)
            {
                MessageBox.Show("Реєстрація за категорією №" + num + " пройшла
                успішно!", "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);
                MessageBox.Show("Пару " + boySurname.Text + " " + boyName.Text + " - " +
                girlSurname.Text + " " + girlName.Text + " додано!", "Інформація", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
            }
            connection.Close();
        }
    }

    catch (MySqlException ex)
    {
        MessageBox.Show(ex.Message);
    }
}
}
reader.Close();
}
}

catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{
    command.Connection.Close();
}
}

}

reader2.Close();

}
catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

}

finally
{
    command2.Connection.Close();
}
}

}
else

```

```
        MessageBox.Show("Реєстрацію не здійснено!", "Помилка!", MessageBoxButtons.OK,  
        MessageBoxIcon.Error);
```

```
    }
```

```
private void girlSurname_TextChanged(object sender, EventArgs e)  
{  
    girlBirth.Enabled = true;  
}
```

```
private void boySurname_TextChanged(object sender, EventArgs e)  
{  
    boyBirth.Enabled = true;  
}
```

```
private void Registration_FormClosed(object sender, FormClosedEventArgs e)  
{  
    Form ifrm = Application.OpenForms[2];  
    ifrm.StartPosition = FormStartPosition.CenterScreen;  
    ifrm.Show();  
}
```

```
private void pictureBox1_Click(object sender, EventArgs e)  
{  
    numberBox.Clear();  
    boyName.Clear();  
    boySurname.Clear();  
    boyBirth.ResetText();  
    girlName.Clear();  
    girlSurname.Clear();  
    girlBirth.ResetText();  
    boyBirth.Enabled = false;  
    girlBirth.Enabled = false;  
    checkedListBox1.Items.Clear();  
    checkedListBox1.Enabled = false;  
    club.Clear();  
    cityy.SelectedIndex = -1;  
    trainer1.Clear();  
    trainer2.Clear();  
    solo.Checked = false;  
}
```

```
private void pictureBox2_Click(object sender, EventArgs e)  
{  
    Form ifrm = Application.OpenForms[2];  
    ifrm.StartPosition = FormStartPosition.CenterScreen;  
    ifrm.Show();  
    this.Close();  
}
```

```
    }
```

```
}
```

New_judge.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using MySql.Data;
using MySql.Data.MySqlClient;

namespace ballroom
{
    public partial class New_judge : Form
    {
        public New_judge()
        {
            InitializeComponent();
        }

        string connStr = "server=localhost;user=root;password=;database=ballroom;port=3306;";

        private void New_judge_Load(object sender, EventArgs e)
        {
            MySqlConnection conn = new MySqlConnection(connStr);
            MySqlCommand command = new MySqlCommand();

            string commandString = "SELECT c.city FROM city c";

            command.CommandText = commandString;
            command.Connection = conn;
            MySqlDataReader reader;
            try
            {
                command.Connection.Open();
                reader = command.ExecuteReader();
                int i = 0;
                while (reader.Read())
                {
                    string cityy = Convert.ToString(reader["city"]);
                    city.Items.Add(cityy);

                    i += 1;
                }

                reader.Close();
            }
            catch (MySqlException ex)
            {
                Console.WriteLine(ex.ToString());
            }

            finally
            {
                command.Connection.Close();
            }
        }
    }
}

```

```

    }

}

private void addButton_Click(object sender, EventArgs e)
{
    if (judgeSurname.Text != "" && judgeName.Text != "" && city.Text != "" && password.Text != "")
    {
        MySqlConnection conn1 = new MySqlConnection(connStr);
        MySqlCommand command1 = new MySqlCommand();

        string commandString1 = "SELECT c.city_id FROM city c WHERE c.city LIKE '" + city.Text + "%'";

        command1.CommandText = commandString1;
        command1.Connection = conn1;

        MySqlDataReader reader1;

        try
        {
            command1.Connection.Open();
            reader1 = command1.ExecuteReader();
            while (reader1.Read())
            {
                int id = Convert.ToInt32(reader1["city_id"]);

                string strSQL1 = "INSERT INTO `judge` (`judge_id`, `judgeSurname`, `judgeName`, `city_id`,
`password`) VALUES (NULL, '" + Convert.ToString(judgeSurname.Text) + "', '" +
Convert.ToString(judgeName.Text) + "', '" + id + "', '" + Convert.ToInt32(password.Text) + "')";

                try
                {
                    using (MySqlConnection connection1 = new MySqlConnection(connStr))
                    {
                        connection1.Open();

                        MySqlCommand cp1 = new MySqlCommand(strSQL1, connection1);

                        if (cp1.ExecuteNonQuery() == 1) MessageBox.Show("Суддю успішно додано!",
"Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);

                        connection1.Close();
                    }
                }

                catch (MySqlException ex)
                {
                    MessageBox.Show(ex.Message);
                }
            }

            reader1.Close();
        }
    }
}

```

```

    }

    catch (MySqlException ex)
    {
        Console.WriteLine(ex.ToString());
    }

    finally
    {
        command1.Connection.Close();
    }

    judgeName.Clear();
    judgeSurname.Clear();

    password.Clear();
    city.SelectedIndex = -1;
}

else
{
    MessageBox.Show("Заповніть усі поля!", "Помилка!", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
    judgeName.Clear();
    judgeSurname.Clear();

    password.Clear();
    city.SelectedIndex = -1;
}

}

private void New_judge_FormClosed(object sender, FormClosedEventArgs e)
{
    Form ifrm = Application.OpenForms[2];
    ifrm.StartPosition = FormStartPosition.CenterScreen;
    ifrm.Show();
}

private void pictureBox1_Click(object sender, EventArgs e)
{
    judgeName.Clear();
    judgeSurname.Clear();
    password.Clear();
    city.SelectedIndex = -1;
}

private void pictureBox2_Click(object sender, EventArgs e)
{
    Form ifrm = Application.OpenForms[2];
    ifrm.StartPosition = FormStartPosition.CenterScreen;
    ifrm.Show();
    this.Close();
}

}
}

```

Reg_print.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using MySql.Data;
using MySql.Data.MySqlClient;
using System.Threading;

namespace ballroom
{
    public partial class Reg_print : Form
    {
        public Reg_print()
        {
            InitializeComponent();
        }

        string connStr = "server=localhost;user=root;password=;database=ballroom;port=3306;";

        private void Reg_print_Load(object sender, EventArgs e)
        {
            MySqlConnection conn = new MySqlConnection(connStr);
            MySqlCommand command = new MySqlCommand();

            string commandString = "SELECT cat.category_name, c.class_name, p.program FROM category cat, class
c, program p, category_class_program ccp, registration reg WHERE p.program_id=ccp.program_id AND
cat.category_id=ccp.category_id AND c.class_id=ccp.class_id AND ccp.c_c_p_id=reg.c_c_p_id GROUP BY
ccp.c_c_p_id";

            command.CommandText = commandString;
            command.Connection = conn;
            MySqlDataReader reader;
            try
            {
                command.Connection.Open();
                reader = command.ExecuteReader();
                int i = 0;
                comboBox1.Items.Clear();
                while (reader.Read())
                {
                    comboBox1.Items.Add(reader["category_name"] + " " + reader["class_name"] + " " +
reader["program"]);
                    i += 1;
                }

                reader.Close();
            }
            catch (MySqlException ex)
            {
                Console.WriteLine(ex.ToString());
            }

            finally

```

```

    {
        command.Connection.Close();
    }

    MySqlConnection conn1 = new MySqlConnection(connStr);
    MySqlCommand command1 = new MySqlCommand();

    string commandString1 = "SELECT COUNT(*) as kol FROM (SELECT COUNT(*) as 'count' FROM
registration GROUP BY number) as reg";

    command1.CommandText = commandString1;
    command1.Connection = conn1;
    MySqlDataReader reader1;
    try
    {
        command1.Connection.Open();
        reader1 = command1.ExecuteReader();
        while (reader1.Read())
        {
            label3.Text=reader1["kol"].ToString();
        }

        reader1.Close();
    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.ToString());
    }

    finally
    {
        command1.Connection.Close();
    }
}

private DataTable GetComments()
{
    DataTable dt = new DataTable();

    MySqlConnection conn = new MySqlConnection(connStr);
    MySqlCommand command = new MySqlCommand();

    string data = comboBox1.SelectedItem.ToString();
    string[] words = data.Split(' ');
    string commandString = "SELECT ccp.c_c_p_id FROM category cat, class c, program p,
category_class_program ccp WHERE p.program_id=ccp.program_id AND cat.category_id=ccp.category_id AND
c.class_id=ccp.class_id AND cat.category_name LIKE '" + words[0] + "%' AND c.class_name LIKE '" + words[2]
+ "%' AND p.program LIKE '" + words[4] + "%'";
    command.CommandText = commandString;
    command.Connection = conn;

    MySqlDataReader reader;

    try
    {
        command.Connection.Open();
        reader = command.ExecuteReader();

        using (MySqlConnection connect = new MySqlConnection(connStr))
        {
            connect.Open();

```



```

MySQLConnection conn1 = new MySqlConnection(connStr);
MySQLCommand command1 = new MySQLCommand();
while (reader.Read())
{
    int num = Convert.ToInt32(reader["c_c_p_id"]);
    string commandString1 = "SELECT reg.number as '№', GROUP_CONCAT(reg.surname, '',
reg.name SEPARATOR '-') as 'Прізвище Ім'я', c.city as 'Місто', reg.club AS 'Клуб', reg.trainer1 AS 'Тренер1',
reg.trainer2 AS 'Тренер2' FROM category cat, class cl, program p, category_class_program ccp, registration reg,
city c WHERE p.program_id=ccp.program_id AND cat.category_id=ccp.category_id AND cl.class_id=ccp.class_id
AND c.city_id=reg.city_id AND ccp.c_c_p_id=reg.c_c_p_id AND reg.c_c_p_id LIKE '" + num + "%' GROUP BY
number";

    command1.CommandText = commandString1;
    command1.Connection = conn1;
    MySQLDataReader reader1;
    try
    {
        command1.Connection.Open();
        using (reader1 = command1.ExecuteReader())
        {
            if (reader1.HasRows)
            {
                dt.Load(reader1);
            }
        }
    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.ToString());
    }

    finally
    {
        command1.Connection.Close();
    }
}

}
catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{
    command.Connection.Close();
}
return dt;
}

private void button1_Click(object sender, EventArgs e)
{
    printPreviewDialog1.ShowDialog();
}

private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
{
    Bitmap bmp = new Bitmap(dataGridView1.Width, dataGridView1.Height);
    dataGridView1.DrawToBitmap(bmp, new Rectangle(0, 0, dataGridView1.Width, dataGridView1.Height));
    e.Graphics.DrawImage(bmp, 30, 100);
}

```

```

        e.Graphics.DrawString("Категорія "+comboBox1.Text.ToString(),new Font("Times New
Roman",20,FontStyle.Bold),Brushes.Black, new Point(150,50));
    }

    private void Reg_print_FormClosed(object sender, FormClosedEventArgs e)
    {
        Form ifrm = Application.OpenForms[2];
        ifrm.StartPosition = FormStartPosition.CenterScreen;
        ifrm.Show();
    }

    private void comboBox1_SelectedValueChanged(object sender, EventArgs e)
    {
        progressBar1.Visible = true;
        progressBar1.Minimum = 0;
        progressBar1.Maximum = 5;
        for (int i = 0; i < 5; i++)
        {
            Thread.Sleep(1000);
            progressBar1.Value++;
        }
        progressBar1.Value = 0;
        dataGridView1.DataSource = GetComments();
        progressBar1.Visible = false;
    }

    private void pictureBox2_Click(object sender, EventArgs e)
    {
        Form ifrm = Application.OpenForms[2];
        ifrm.StartPosition = FormStartPosition.CenterScreen;
        ifrm.Show();
        this.Close();
    }
}
}
}

```

Print.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using MySql.Data;
using MySql.Data.MySqlClient;

namespace ballroom
{
    public partial class Print : Form
    {
        public Print()
        {
            InitializeComponent();

```

```

}

string connStr = "server=localhost;user=root;password=;database=ballroom;port=3306;";

private void Print_Load(object sender, EventArgs e)
{
    MySqlConnection conn = new MySqlConnection(connStr);
    MySqlCommand command = new MySqlCommand();

    string commandString = "SELECT cat.category_name, c.class_name, p.program FROM category cat, class
c, program p, category_class_program ccp, registration reg WHERE p.program_id=ccp.program_id AND
cat.category_id=ccp.category_id AND c.class_id=ccp.class_id AND ccp.c_c_p_id=reg.c_c_p_id GROUP BY
ccp.c_c_p_id";

    command.CommandText = commandString;
    command.Connection = conn;
    MySqlDataReader reader;
    try
    {
        command.Connection.Open();
        reader = command.ExecuteReader();
        int i = 0;
        comboBox1.Items.Clear();
        while (reader.Read())
        {
            comboBox1.Items.Add(reader["category_name"] + " " + reader["class_name"] + " " +
reader["program"]);
            i += 1;
        }

        reader.Close();
    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.ToString());
    }

    finally
    {
        command.Connection.Close();
    }
}

private DataTable GetMembers()
{
    DataTable dt = new DataTable();

    MySqlConnection conn = new MySqlConnection(connStr);
    MySqlCommand command = new MySqlCommand();

    string data = comboBox1.SelectedItem.ToString();
    string[] words = data.Split(' ');
    string commandString = "SELECT ccp.c_c_p_id FROM category cat, class c, program p,
category_class_program ccp WHERE p.program_id=ccp.program_id AND cat.category_id=ccp.category_id AND
c.class_id=ccp.class_id AND cat.category_name LIKE '" + words[0] + "%' AND c.class_name LIKE '" + words[2]
+ "%' AND p.program LIKE '" + words[4] + "%'";
    command.CommandText = commandString;
    command.Connection = conn;

    MySqlDataReader reader;

```

```

try
{
    command.Connection.Open();
    reader = command.ExecuteReader();

    using ( MySqlConnection connect = new MySqlConnection(connStr)
    {
        connect.Open();
        MySqlConnection conn1 = new MySqlConnection(connStr);
        MySqlCommand command1 = new MySqlCommand();
        while (reader.Read())
        {
            int num = Convert.ToInt32(reader["c_c_p_id"]);
            string commandString1 = "SELECT reg.number as '№', GROUP_CONCAT(reg.surname, ' ',
reg.name SEPARATOR '-') as 'Прізвище Ім'я', c.city as 'Місто', reg.club AS 'Клуб', reg.trainer1 AS 'Тренер1',
reg.trainer2 AS 'Тренер2' FROM category cat, class cl, program p, category_class_program ccp, registration reg,
city c WHERE p.program_id=ccp.program_id AND cat.category_id=ccp.category_id AND cl.class_id=ccp.class_id
AND c.city_id=reg.city_id AND ccp.c_c_p_id=reg.c_c_p_id AND reg.c_c_p_id LIKE '" + num + "%' GROUP BY
number";

            command1.CommandText = commandString1;
            command1.Connection = conn1;
            MySqlDataReader reader1;
            try
            {
                command1.Connection.Open();
                using (reader1 = command1.ExecuteReader())
                {
                    if (reader1.HasRows)
                    {
                        dt.Load(reader1);
                    }
                }
            }
            catch (MySqlException ex)
            {
                Console.WriteLine(ex.ToString());
            }

            finally
            {
                command1.Connection.Close();
            }
        }
    }
}
catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{
    command.Connection.Close();
}
return dt;
}

```

```

private DataTable GetMarks()
{
    DataTable dt = new DataTable();

    MySqlConnection conn = new MySqlConnection(connStr);
    MySqlCommand command = new MySqlCommand();

    string data = comboBox1.SelectedItem.ToString();
    string[] words = data.Split(' ');
    string commandString = "SELECT ccp.c_c_p_id FROM category cat, class c, program p,
category_class_program ccp WHERE p.program_id=ccp.program_id AND cat.category_id=ccp.category_id AND
c.class_id=ccp.class_id AND cat.category_name LIKE '" + words[0] + "%' AND c.class_name LIKE '" + words[2]
+ "%' AND p.program LIKE '" + words[4] + "%'";
    command.CommandText = commandString;
    command.Connection = conn;

    MySqlDataReader reader;

    try
    {
        command.Connection.Open();
        reader = command.ExecuteReader();

        using (MySqlConnection connect = new MySqlConnection(connStr))
        {
            connect.Open();
            MySqlConnection conn1 = new MySqlConnection(connStr);
            MySqlCommand command1 = new MySqlCommand();
            while (reader.Read())
            {
                int num = Convert.ToInt32(reader["c_c_p_id"]);
                string commandString1 = "SELECT r.number as '№', d.dance_name as 'Танець',
GROUP_CONCAT(r.mark,' SEPARATOR '|') as 'Місія', SUM(r.mark) as 'Сума' FROM results r, log l, dances d,
category_class_program ccp WHERE l.c_c_p_id="+num+" AND ccp.c_c_p_id=l.c_c_p_id AND l.id=r.id AND
l.dance_id=d.dance_id GROUP BY r.id,r.number";

                command1.CommandText = commandString1;
                command1.Connection = conn1;
                MySqlDataReader reader1;
                try
                {
                    command1.Connection.Open();
                    using (reader1 = command1.ExecuteReader())
                    {
                        if (reader1.HasRows)
                        {
                            dt.Load(reader1);
                        }
                    }
                }
                catch (MySqlException ex)
                {
                    Console.WriteLine(ex.ToString());
                }

                finally
                {
                    command1.Connection.Close();
                }
            }
        }
    }
}

```

```

    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.ToString());
    }

    finally
    {
        command.Connection.Close();
    }
    return dt;
}

private DataTable GetResults()
{
    DataTable dt = new DataTable();

    MySqlConnection conn = new MySqlConnection(connStr);
    MySqlCommand command = new MySqlCommand();

    string data = comboBox1.SelectedItem.ToString();
    string[] words = data.Split(' ');
    string commandString = "SELECT ccp.c_c_p_id FROM category cat, class c, program p,
category_class_program ccp WHERE p.program_id=ccp.program_id AND cat.category_id=ccp.category_id AND
c.class_id=ccp.class_id AND cat.category_name LIKE '" + words[0] + "%' AND c.class_name LIKE '" + words[2]
+ "%' AND p.program LIKE '" + words[4] + "%'";
    command.CommandText = commandString;
    command.Connection = conn;

    MySqlDataReader reader;

    try
    {
        command.Connection.Open();
        reader = command.ExecuteReader();

        using (MySqlConnection connect = new MySqlConnection(connStr))
        {
            connect.Open();
            MySqlConnection conn1 = new MySqlConnection(connStr);
            MySqlCommand command1 = new MySqlCommand();
            while (reader.Read())
            {
                int num = Convert.ToInt32(reader["c_c_p_id"]);
                MySqlConnection con = new MySqlConnection(connStr);
                con.Open();
                MySqlCommand cmd = con.CreateCommand();
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = "SELECT IF(SUM(r.mark)=(SELECT MIN(summ) as summa FROM
(SELECT SUM(r.mark) as summ FROM rezults r, log l, category_class_program ccp WHERE
l.c_c_p_id="+num+" AND ccp.c_c_p_id=l.c_c_p_id AND l.id=r.id GROUP BY r.number) r), 1, @i+1) as 'Micue',
r.number as '№' FROM rezults r, log l, dances d, category_class_program ccp WHERE l.c_c_p_id="+num+" AND
ccp.c_c_p_id=l.c_c_p_id AND l.id=r.id AND l.dance_id=d.dance_id GROUP BY r.number ORDER BY
SUM(r.mark)";
                MySqlDataAdapter adap = new MySqlDataAdapter();
                cmd.Parameters.AddWithValue("@i", 1);
                adap.SelectCommand = cmd;
                adap.Fill(dt);
                con.Close();
            }
        }
    }
}

```

```

    }

    }
catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{
    command.Connection.Close();
}
return dt;
}

private void comboBox1_SelectedValueChanged(object sender, EventArgs e)
{
    dataGridView1.DataSource = GetMembers();
    dataGridView2.DataSource = GetMarks();
    dataGridView3.DataSource = GetResults();

}

private void pictureBox2_Click(object sender, EventArgs e)
{
    Form ifrm = Application.OpenForms[2];
    ifrm.StartPosition = FormStartPosition.CenterScreen;
    ifrm.Show();
    this.Close();
}

private void button1_Click(object sender, EventArgs e)
{
    printPreviewDialog1.ShowDialog();
}

private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
{
    Bitmap bmp1 = new Bitmap(dataGridView1.Width, dataGridView1.Height);
    Bitmap bmp2 = new Bitmap(dataGridView2.Width, dataGridView2.Height);
    Bitmap bmp3 = new Bitmap(dataGridView3.Width, dataGridView3.Height);
    dataGridView1.DrawToBitmap(bmp1, new Rectangle(0, 0, dataGridView1.Width, dataGridView1.Height));
    dataGridView2.DrawToBitmap(bmp2, new Rectangle(0, 0, dataGridView2.Width, dataGridView2.Height));
    dataGridView3.DrawToBitmap(bmp3, new Rectangle(0, 0, dataGridView3.Width, dataGridView3.Height));
    e.Graphics.DrawImage(bmp1, 40, 100);
    e.Graphics.DrawImage(bmp2, 260, dataGridView1.Height+200);
    e.Graphics.DrawImage(bmp3, 335, dataGridView1.Height+dataGridView2.Height+280);
    e.Graphics.DrawString("Категорія " + comboBox1.Text.ToString(), new Font("Times New Roman", 20,
    FontStyle.Bold), Brushes.Black, new Point(150, 50));
    e.Graphics.DrawString("Результати туру: ФІНАЛ", new Font("Times New Roman", 20, FontStyle.Bold),
    Brushes.Black, new Point(250, dataGridView1.Height + 150));
    e.Graphics.DrawString("ПІДСУМОК", new Font("Times New Roman", 20, FontStyle.Bold),
    Brushes.Black, new Point(330, dataGridView1.Height+dataGridView2.Height + 230));
}

private void Print_FormClosed(object sender, FormClosedEventArgs e)
{
    Form ifrm = Application.OpenForms[2];
    ifrm.StartPosition = FormStartPosition.CenterScreen;
    ifrm.Show();
}

```

```

    }

}
}

```

Final.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using System.Threading;

using MySql.Data;
using MySql.Data.MySqlClient;

namespace ballroom
{
    public partial class Final : Form
    {
        public Final()
        {
            InitializeComponent();
        }

        string connStr = "server=localhost;user=root;password=;database=ballroom;port=3306;";

        private void Final_Load(object sender, EventArgs e)
        {
            start();
        }

        private void start()
        {
            label2.Text = Convert.ToString(Judge.judge);
            MySqlConnection conn = new MySqlConnection(connStr);
            MySqlCommand command = new MySqlCommand();
            string commandString = "SELECT cat.category_name, c.class_name, p.program, d.dance_name,
ccp.c_c_p_id FROM category cat, class c, program p, category_class_program ccp, dances d, (SELECT * FROM
log l ORDER BY id DESC limit 1) l WHERE p.program_id=ccp.program_id AND cat.category_id=ccp.category_id
AND c.class_id=ccp.class_id AND ccp.c_c_p_id=l.c_c_p_id AND d.dance_id=l.dance_id";

            command.CommandText = commandString;
            command.Connection = conn;
            MySqlDataReader reader;
            try
            {
                command.Connection.Open();
                reader = command.ExecuteReader();
            }

```



```

while (reader.Read())
{
    label1.Text = (reader["category_name"] + " " + reader["class_name"] + " " + reader["program"]);
    label3.Text = (reader["dance_name"]).ToString();

    int num = Convert.ToInt32(reader["c_c_p_id"]);

    MySqlConnection conn1 = new MySqlConnection(connStr);
    MySqlCommand command1 = new MySqlCommand();

    string commandString1 = "SELECT COUNT(*) as kol FROM (SELECT COUNT(*) as 'count',
c_c_p_id FROM registration GROUP BY number) as reg WHERE c_c_p_id=" + num + """;

    command1.CommandText = commandString1;
    command1.Connection = conn1;
    MySqlDataReader reader1;
    try
    {
        command1.Connection.Open();
        reader1 = command1.ExecuteReader();

        while (reader1.Read())
        {
            string s = Convert.ToString(reader1["kol"]);
            int n = int.Parse(s) + 1; //кількість кнопок(місць)

            if (n == 7)
            {
                for (int i = 1; i < n; i++)
                {
                    checkBox1.Visible = true;
                    checkBox2.Visible = true;
                    checkBox3.Visible = true;
                    checkBox4.Visible = true;
                    checkBox5.Visible = true;
                    checkBox6.Visible = true;
                    checkBox7.Visible = true;
                    checkBox8.Visible = true;
                    checkBox9.Visible = true;
                    checkBox10.Visible = true;
                    checkBox11.Visible = true;
                    checkBox12.Visible = true;
                    checkBox13.Visible = true;
                    checkBox14.Visible = true;
                    checkBox15.Visible = true;
                    checkBox16.Visible = true;
                    checkBox17.Visible = true;
                    checkBox18.Visible = true;
                    checkBox19.Visible = true;
                    checkBox20.Visible = true;
                    checkBox21.Visible = true;
                    checkBox22.Visible = true;
                    checkBox23.Visible = true;
                    checkBox24.Visible = true;
                    checkBox25.Visible = true;
                    checkBox26.Visible = true;
                    checkBox27.Visible = true;
                    checkBox28.Visible = true;
                    checkBox29.Visible = true;
                    checkBox30.Visible = true;
                }
            }
        }
    }
}

```

```
        checkBox31.Visible = true;
        checkBox32.Visible = true;
        checkBox33.Visible = true;
        checkBox34.Visible = true;
        checkBox35.Visible = true;
        checkBox36.Visible = true;
    }
}
else if (n == 6)
{
    for (int i = 1; i < n; i++)
    {
        checkBox1.Visible = true;
        checkBox2.Visible = true;
        checkBox3.Visible = true;
        checkBox4.Visible = true;
        checkBox5.Visible = true;
        checkBox7.Visible = true;
        checkBox8.Visible = true;
        checkBox9.Visible = true;
        checkBox10.Visible = true;
        checkBox11.Visible = true;
        checkBox13.Visible = true;
        checkBox14.Visible = true;
        checkBox15.Visible = true;
        checkBox16.Visible = true;
        checkBox17.Visible = true;
        checkBox19.Visible = true;
        checkBox20.Visible = true;
        checkBox21.Visible = true;
        checkBox22.Visible = true;
        checkBox23.Visible = true;
        checkBox25.Visible = true;
        checkBox26.Visible = true;
        checkBox27.Visible = true;
        checkBox28.Visible = true;
        checkBox29.Visible = true;
    }
}
else if (n == 5)
{
    for (int i = 1; i < n; i++)
    {
        checkBox1.Visible = true;
        checkBox2.Visible = true;
        checkBox3.Visible = true;
        checkBox4.Visible = true;
        checkBox7.Visible = true;
        checkBox8.Visible = true;
        checkBox9.Visible = true;
        checkBox10.Visible = true;
        checkBox13.Visible = true;
        checkBox14.Visible = true;
        checkBox15.Visible = true;
        checkBox16.Visible = true;
        checkBox19.Visible = true;
        checkBox20.Visible = true;
        checkBox21.Visible = true;
        checkBox22.Visible = true;
    }
}
```

```

}
else if (n == 4)
{
    for (int i = 1; i < n; i++)
    {
        checkBox1.Visible = true;
        checkBox2.Visible = true;
        checkBox3.Visible = true;
        checkBox7.Visible = true;
        checkBox8.Visible = true;
        checkBox9.Visible = true;
        checkBox13.Visible = true;
        checkBox14.Visible = true;
        checkBox15.Visible = true;
    }
}
else if (n == 3)
{
    for (int i = 1; i < n; i++)
    {
        checkBox1.Visible = true;
        checkBox2.Visible = true;
        checkBox7.Visible = true;
        checkBox8.Visible = true;
    }
}
else if (n == 2)
{
    for (int i = 1; i < n; i++)
    {
        checkBox1.Visible = true;
    }
}

```

```

MySQLConnection conn2 = new MySQLConnection(connStr);
MySQLCommand command2 = new MySQLCommand();

```

```

string commandString2 = "SELECT number FROM registration WHERE c_c_p_id=" + num + "
GROUP BY number";

```

```

command2.CommandText = commandString2;
command2.Connection = conn2;
MySQLDataReader reader2;

```

```

string[] items = new string[20];
int j = 0;
try
{
    command2.Connection.Open();
    reader2 = command2.ExecuteReader();

    while (reader2.Read())
    {
        items[j] = Convert.ToString(reader2["number"]);
        j++;
    }
    reader2.Close();
    if (j == 5)

```

```
{
    label4.Visible = true;
    label5.Visible = true;
    label6.Visible = true;
    label7.Visible = true;
    label8.Visible = true;
    label9.Visible = true;

    label4.Text = items[0];
    label5.Text = items[1];
    label6.Text = items[2];
    label7.Text = items[3];
    label8.Text = items[4];
    label9.Text = items[5];
}
else if (j == 4)
{
    label4.Visible = true;
    label5.Visible = true;
    label6.Visible = true;
    label7.Visible = true;
    label8.Visible = true;

    label4.Text = items[0];
    label5.Text = items[1];
    label6.Text = items[2];
    label7.Text = items[3];
    label8.Text = items[4];
}
else if (j == 3)
{
    label4.Visible = true;
    label5.Visible = true;
    label6.Visible = true;
    label7.Visible = true;

    label4.Text = items[0];
    label5.Text = items[1];
    label6.Text = items[2];
    label7.Text = items[3];
}
else if (j == 2)
{
    label4.Visible = true;
    label5.Visible = true;
    label6.Visible = true;

    label4.Text = items[0];
    label5.Text = items[1];
    label6.Text = items[2];
}
else if (j == 1)
{
    label4.Visible = true;
    label5.Visible = true;

    label4.Text = items[0];
    label5.Text = items[1];
}
else
{
    label4.Visible = true;
```

```

        label4.Text = items[0];
    }

    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.ToString());
    }

    finally
    {
        command.Connection.Close();
    }
}

reader1.Close();

}
catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{
    command.Connection.Close();
}
}

reader.Close();
}
catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{
    command.Connection.Close();
}
}

private void button1_Click(object sender, EventArgs e)
{
    for (int i = 1; i < 37; i++)
    {
        if (i >= 1 && i < 7)
        {
            if (((CheckBox)(this.Controls["checkBox" + i.ToString])).Checked == true)
            {
                MySqlConnection conn = new MySqlConnection(connStr);
                MySqlCommand command = new MySqlCommand();
                string commandString = "SELECT id FROM log 1 ORDER BY id DESC limit 1";

                command.CommandText = commandString;
                command.Connection = conn;
                MySqlDataReader reader;
                try
                {
                    command.Connection.Open();
                    reader = command.ExecuteReader();
                }
            }
        }
    }
}

```

```

while (reader.Read())
{
    string id = Convert.ToString(reader["id"]);

    string strSQL = "INSERT INTO `rezults` (`rez_id`, `id`, `judge_id`, `number`, `mark`)
VALUES (NULL, '" + id + "', '"+label2.Text+"', '" + label4.Text + "', '" + ((CheckBox)(this.Controls["checkBox" +
i.ToString()])).Text + "'";

    try
    {

        using (MySQLConnection connection = new MySQLConnection(connStr))
        {
            connection.Open();
            MySqlCommand cp = new MySqlCommand(strSQL, connection);

            if (cp.ExecuteNonQuery() == 1)
            {
                MessageBox.Show("Результат збережено!", "Інформація", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }
            connection.Close();
        }

        catch (MySQLException ex)
        {
            MessageBox.Show(ex.Message);
        }
        reader.Close();
    }

    catch (MySQLException ex)
    {
        Console.WriteLine(ex.ToString());
    }

    finally
    {
        command.Connection.Close();
    }
}
else
{
    MessageBox.Show("Ви нічого не обрали для №"+label4.Text+". Спробуйте ще!", "Помилка!",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
else if (i >= 7 && i < 13)
{
    if (((CheckBox)(this.Controls["checkBox" + i.ToString()])).Checked == true)
    {
        MySQLConnection conn = new MySQLConnection(connStr);
        MySqlCommand command = new MySqlCommand();
        string commandString = "SELECT id FROM log l ORDER BY id DESC limit 1";

        command.CommandText = commandString;
        command.Connection = conn;
    }
}
}

```

```

MySQLDataReader reader;
try
{
    command.Connection.Open();
    reader = command.ExecuteReader();

    while (reader.Read())
    {
        string id = Convert.ToString(reader["id"]);

        string strSQL = "INSERT INTO `results` (`rez_id`, `id`, `judge_id`, `number`, `mark`)
VALUES (NULL, '" + id + "', '" + label2.Text + "', '" + label5.Text + "', '" + ((CheckBox)(this.Controls["checkBox" +
i.ToString()])).Text + "')";

        try
        {
            using (MySQLConnection connection = new MySQLConnection(connStr))
            {
                connection.Open();
                MySqlCommand cp = new MySqlCommand(strSQL, connection);

                if (cp.ExecuteNonQuery() == 1)
                {
                    MessageBox.Show("Результат збережено!", "Інформація", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                }
                connection.Close();
            }
        }
        catch (MySQLException ex)
        {
            MessageBox.Show(ex.Message);
        }
        reader.Close();
    }
}

catch (MySQLException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{
    command.Connection.Close();
}

else
{
    MessageBox.Show("Ви нічого не обрали для №" + label5.Text + ". Спробуйте ще!", "Помилка!",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
else if (i >= 13 && i < 19)
{
    if (((CheckBox)(this.Controls["checkBox" + i.ToString()])).Checked == true)
    {
        MySQLConnection conn = new MySQLConnection(connStr);
    }
}

```

```

MySQLCommand command = new MySQLCommand();
string commandString = "SELECT id FROM log l ORDER BY id DESC limit 1";

command.CommandText = commandString;
command.Connection = conn;
MySQLDataReader reader;
try
{
    command.Connection.Open();
    reader = command.ExecuteReader();

    while (reader.Read())
    {
        string id = Convert.ToString(reader["id"]);

        string strSQL = "INSERT INTO `results` (`rez_id`, `id`, `judge_id`, `number`, `mark`)
VALUES (NULL, '" + id + "', '"+label2.Text+"', '"+ label6.Text + "', '" + ((CheckBox)(this.Controls["checkBox"] +
i.ToString()))).Text + "'";

        try
        {

            using (MySQLConnection connection = new MySQLConnection(connStr))
            {
                connection.Open();
                MySQLCommand cp = new MySQLCommand(strSQL, connection);

                if (cp.ExecuteNonQuery() == 1)
                {
                    MessageBox.Show("Результат збережено!", "Інформація", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                }
                connection.Close();
            }
        }

        catch (MySQLException ex)
        {
            MessageBox.Show(ex.Message);
        }
        reader.Close();
    }
}

catch (MySQLException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{
    command.Connection.Close();
}
}

else
{
    MessageBox.Show("Ви нічого не обрали для №"+label6.Text+" . Спробуйте ще!", "Помилка!",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

```



```

else if (i >= 19 && i < 25)
{
    if (((CheckBox)(this.Controls["checkBox" + i.ToString()])).Checked == true)
    {
        MySqlConnection conn = new MySqlConnection(connStr);
        MySqlCommand command = new MySqlCommand();
        string commandString = "SELECT id FROM log l ORDER BY id DESC limit 1";

        command.CommandText = commandString;
        command.Connection = conn;
        MySqlDataReader reader;
        try
        {
            command.Connection.Open();
            reader = command.ExecuteReader();

            while (reader.Read())
            {
                string id = Convert.ToString(reader["id"]);

                string strSQL = "INSERT INTO `results` (`rez_id`, `id`, `judge_id`, `number`, `mark`)
VALUES (NULL, '" + id + "', '"+label2.Text+"', '" + label7.Text + "', '" + ((CheckBox)(this.Controls["checkBox" +
i.ToString()])).Text + "')";

                try
                {
                    using (MySqlConnection connection = new MySqlConnection(connStr))
                    {
                        connection.Open();
                        MySqlCommand cp = new MySqlCommand(strSQL, connection);

                        if (cp.ExecuteNonQuery() == 1)
                        {
                            MessageBox.Show("Результат збережено!", "Інформація", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                        }
                    }
                    connection.Close();
                }
            }

            catch (MySqlException ex)
            {
                MessageBox.Show(ex.Message);
            }
            reader.Close();
        }
    }

    catch (MySqlException ex)
    {
        Console.WriteLine(ex.ToString());
    }

    finally
    {
        command.Connection.Close();
    }
}
else

```

```

    {
        MessageBox.Show("Ви нічого не обрали для №"+label7.Text+". Спробуйте ще!", "Помилка!",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
else if (i >= 25 && i < 31)
{
    if (((CheckBox)(this.Controls["checkBox" + i.ToString()])).Checked == true)
    {
        MySqlConnection conn = new MySqlConnection(connStr);
        MySqlCommand command = new MySqlCommand();
        string commandString = "SELECT id FROM log l ORDER BY id DESC limit 1";

        command.CommandText = commandString;
        command.Connection = conn;
        MySqlDataReader reader;
        try
        {
            command.Connection.Open();
            reader = command.ExecuteReader();

            while (reader.Read())
            {
                string id = Convert.ToString(reader["id"]);

                string strSQL = "INSERT INTO `results` (`rez_id`, `id`, `judge_id`, `number`, `mark`)
VALUES (NULL, '" + id + "', '" + label2.Text + "', '" + label8.Text + "', '" + ((CheckBox)(this.Controls["checkBox" +
i.ToString()])).Text + "')";

                try
                {
                    using (MySqlConnection connection = new MySqlConnection(connStr))
                    {
                        connection.Open();
                        MySqlCommand cp = new MySqlCommand(strSQL, connection);

                        if (cp.ExecuteNonQuery() == 1)
                        {
                            MessageBox.Show("Результат збережено!", "Інформація", MessageBoxButtons.OK,
                            MessageBoxIcon.Information);
                        }
                    }
                    connection.Close();
                }
            }
        }
        catch (MySqlException ex)
        {
            MessageBox.Show(ex.Message);
        }
        reader.Close();
    }
}

catch (MySqlException ex)
{
    Console.WriteLine(ex.ToString());
}

finally
{

```

```

        command.Connection.Close();
    }
}

else
{
    MessageBox.Show("Ви нічого не обрали для №"+label8.Text+". Спробуйте ще!", "Помилка!",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
else if (i >= 31 && i < 37)
{
    if (((CheckBox)(this.Controls["checkBox" + i.ToString()])).Checked == true)
    {
        MySqlConnection conn = new MySqlConnection(connStr);
        MySqlCommand command = new MySqlCommand();
        string commandString = "SELECT id FROM log l ORDER BY id DESC limit 1";

        command.CommandText = commandString;
        command.Connection = conn;
        MySqlDataReader reader;
        try
        {
            command.Connection.Open();
            reader = command.ExecuteReader();

            while (reader.Read())
            {
                string id = Convert.ToString(reader["id"]);

                string strSQL = "INSERT INTO `results` (`rez_id`, `id`, `judge_id`, `number`, `mark`)
VALUES (NULL, '" + id + "', '" + label2.Text + "', '" + label9.Text + "', '" + ((CheckBox)(this.Controls["checkBox" +
i.ToString()])).Text + "')";

                try
                {
                    using (MySqlConnection connection = new MySqlConnection(connStr))
                    {
                        connection.Open();
                        MySqlCommand cp = new MySqlCommand(strSQL, connection);

                        if (cp.ExecuteNonQuery() == 1)
                        {
                            MessageBox.Show("Результат збережено!", "Інформація", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
                        }
                    }
                    connection.Close();
                }
            }
        }
        catch (MySqlException ex)
        {
            MessageBox.Show(ex.Message);
        }
        reader.Close();
    }
}

catch (MySqlException ex)
{

```


Додаток Г. Настанова з використання

Для інсталяції та подальшого використання розробленої інформаційної системи необхідно запустити майстер установки «Ballroom». Для цього треба зайти в папку з інсталятором Setup.exe і запустити його.

На рисунках Г.1 – Г.4 представлені пошагові дії по встановленню програми Ballroom.

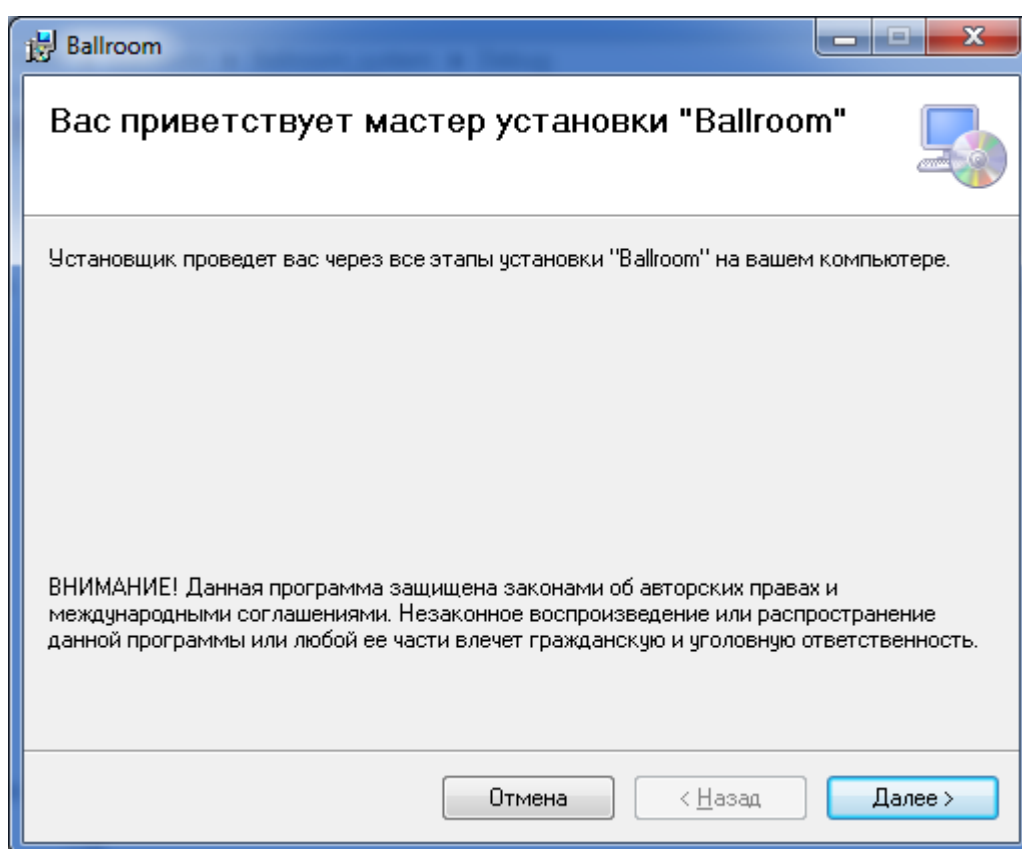


Рисунок Г.1 – Початкове вікно встановлення

Для коректної роботи програми на OS Windows 10 і нижче рекомендовано встановлювати у місце, де доступні повні права адміністратора (рис. Г.2).

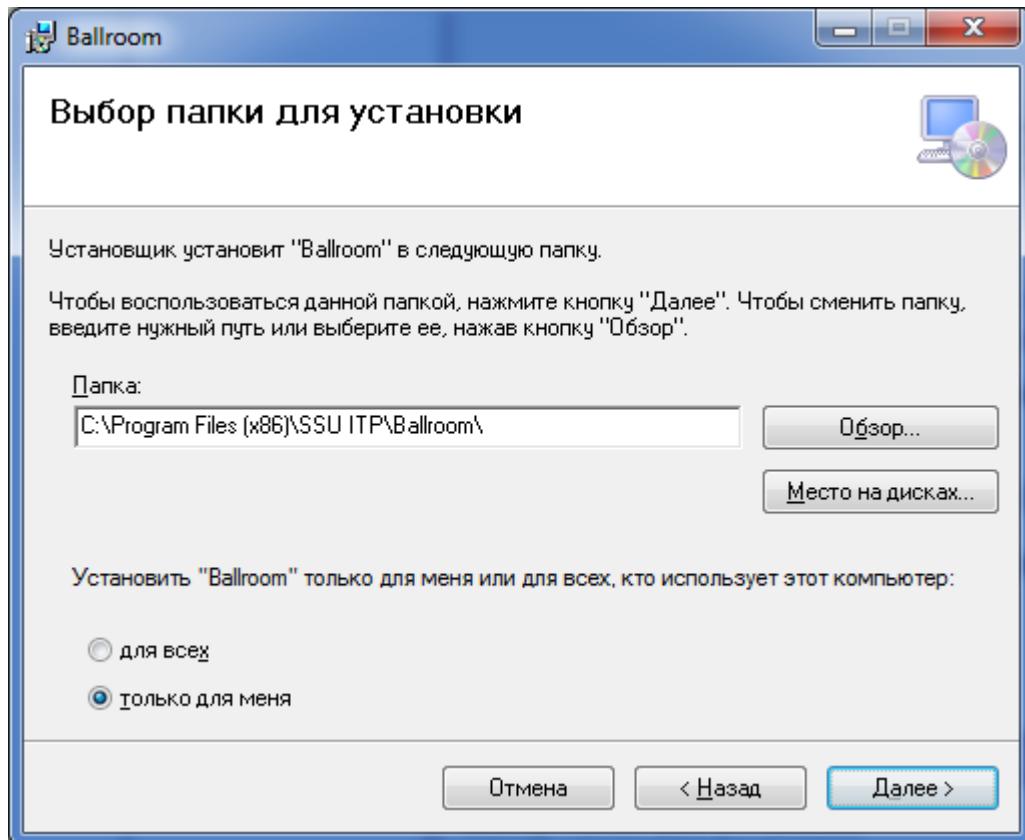


Рисунок Г.2 – Вибір папки для встановлення

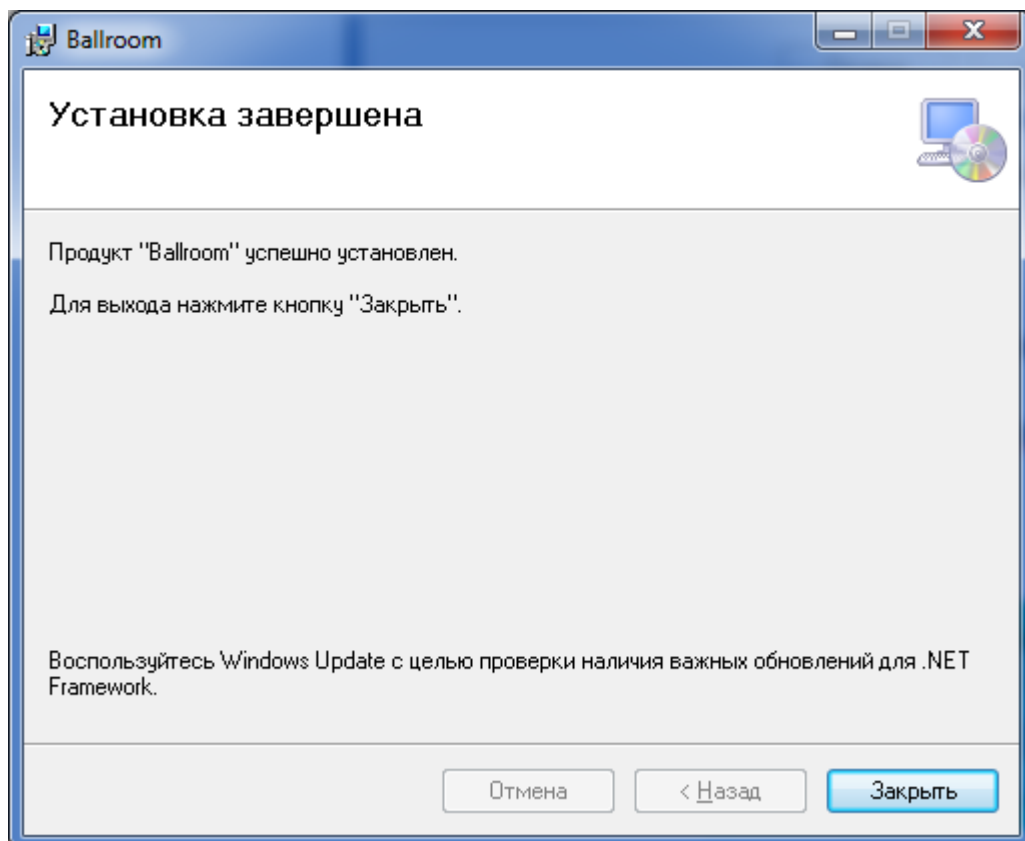


Рисунок Г.3 – Встановлення завершено

Після встановлення завершення встановлення програми, на робочому столі з'являється ярлик (рис. Г.4)



Рисунок Г.4 – Ярлик на робочому столі

Запуск ІС можна виконувати як подвійним кліком по ярлику на робочому столі так і запустити .exe файл у папці, куди була встановлена програма.

Додаток Д.
Копії акту впровадження ІС

Президент КСТ Силует

Тарасівка О.І.

«07» серпня 2019 р.

Акт

впровадження результатів дипломного проекту
студента Сумського державного університету
Жук Олени Костянтинівни

Даним актом підтверджується, що результати роботи студента Жук О. К. на тему «Інформаційна система підтримки проведення змагань зі спортивних бальних танців» впроваджена в роботу клубу спортивного танцю Силует.

Дана інформаційна система реалізує функції ведення звітної документації проведених змагань, проведення танцювальних змагань, ведення бази даних.

Впровадження системи в роботу відділу дозволило підвищити продуктивність виконання основних типів робіт, зменшити людські фактори помилок, підвищити якість формування звітної документації. Розроблена база даних забезпечує цілісність та надійність збереження даних, швидке формування вибірок за різними критеріями.

Президент КСТ Силует

Заслужений працівник культури України

