

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ФАХОВИЙ КОЛЕДЖ РАКЕТНО-КОСМІЧНОГО МАШИНОБУДУВАННЯ
ДНІПРОВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ім. О. ГОНЧАРА

Циклова комісія програмної інженерії

КУРСОВИЙ ПРОЕКТ
з навчальної дисципліни
«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»

на тему: «Програма ведення обліку задач НМТ»

(вказати тему курсового проекту)

Студента ІVкурсу ПЗ-21-1 групи
галузь знань 12 «Інформаційні технології»
спеціальності 121 «Інженерія програмного
забезпечення»

Гуненко Я.М.

(прізвище та ініціали студента)

Керівник

викладач Гапоненко Н.В.

Національна шкала _____

Кількість балів: _____ Оцінка ECTS: _____

Члени комісії _____

(підпис)

Любохинець В.М.

(прізвище та ініціали)

Ланська С.С.

(підпис)

(прізвище та ініціали)

Гапоненко Н.В.

(підпис)

(прізвище та ініціали)

м. Дніпро - 2024 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ФАХОВИЙ КОЛЕДЖ РАКЕТНО-КОСМІЧНОГО МАШИНОБУДУВАННЯ
ДНІПРОВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ім. О. ГОНЧАРА

Циклова комісія програмної інженерії

ЗАТВЕРДЖУЮ

Голова комісії ПІ

С.С.Ланська

«__» _____ 2024 р.

ЗАВДАННЯ
на виконання курсового проекту

з дисципліни Об'єктно-орієнтоване програмування
студенту Гуненко Ярославу Максимовичу
(прізвище, ім'я та по батькові)
Відділення Комп'ютерної та програмної інженерії
Спеціальність 121 Інженерія програмного забезпечення
Курс IV Група (шифр) ПЗ-21-1
1 Тема проекту «Програма ведення обліку здачі НМТ»

2 Початкові дані Перелік даних про учня (ID учня, Номер паспорту, Тип ,
Паспорту, ПІБ, Дата народження, Стать, Е-mai, Номер телефону, Номер
свідоства про освіту, ПІН, Примітка), перелік навчальних закладів(місто,
область, тип, Е-mail, ПІБ відповідального, Шифр навчального закладу)
перелік предметів(Код предмету, Назва предмету, Опис, Зразок завдань)
термін дії, статус) перелік сертифікатів(Номер сертифікату, ID учня, PIN,
Дата створення, Статус) перелік результатів тесту(Шифр результату,
ID умови, Отриманий бал, Статус здачі, Дата проходження, Шифр Статус
навчального закладу) Умови проходження(ID умови, Код предмету,
Максимальний бал, Мінімальний бал, Мінімальний бал для проходження,
предмету, Дата ухвалення умов)

Розглянуто і ухвалено на засіданні циклової комісії програмної інженерії
Протокол № 2 від 21.09.2021 р.

Керівник КП

Н.В.Гапоненко

(підпис)

(ініціали та прізвище)

Завдання до виконання
одержав студент

(підпис)

(ініціали та прізвище)

Дата видачі

21 вересня 2021 р.

Термін виконання

22 листопада 2021 р.

ЗМІСТ

ВСТУП.....	4
1 ПОСТАНОВКА ЗАДАЧІ.....	5
2 ХАРАКТЕРИСТИКА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОЕКТУ	7
2.1 Опис середовища програмування	7
2.2 Опис мови програмування	8
2.3 Опис СКБД	8
2.4 Опис основних принципів ООП.....	9
2.5 Опис подібних програмних продуктів.....	13
3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ	14
3.1 Опис бази даних.....	14
3.2 ПРОЕКТУВАННЯ КОРИСТУВАЧЬКОГО ІНТЕРФЕЙСУ ПРОГРАМИ	16

					КП.ПЗ.211.06.ПЗ				
Змн.	Лист	№ докум.	Підпис	Дата	Програма ведення обліку здачі НМТ	Літ.	Арк.	Аркуші	
Розроб.	Гуненко Я.М.						3	24	
Перевір.	Гапоненко Н.В.								
Реценз.						ВСП" ФКРКМ ДНУ" ім. Олеся Гончара			
Н. контр.									
Затверд.									

ВСТУП

У сучасному світі точне зберігання даних та автоматизоване управління документацією потрібні скрізь. Цифрова епоха поширюється і на сферу освіти. Щороку в Україні учні, які закінчують 11-й клас, складають іспит, що відображає рівень знань, які вони здобули за цей час. Наразі, у зв'язку з воєнним станом, Зовнішнє незалежне оцінювання, скорочено ЗНО, замінено на Національний мультимедійний тест, скорочено НМТ.

Для центрів оцінювання доступно багато програмного забезпечення, але ці програми складні у вивченні та незручні у використанні. Більшість цих програм базуються на залежних від мережі інтернет WEB технологіях, що не завжди зручно. Програми повинні бути простими у використанні, швидкими, легкими в освоєнні інструментами та універсальними для центрів оцінювання будь-якого розміру. Програмне забезпечення повинно бути доступним для використання навіть недосвідченими користувачами комп'ютерів. Воно може використовуватися як однією особою (наприклад, у невеликому центрі оцінювання), так і групою осіб (наприклад, у великому центрі оцінювання з кількома посадами).

Основна мета курсового проекту - створити корисний і простий додаток для великих і малих центрів оцінювання.

	Вик.	Гуненко Я.М.			КП.ПЗ.211.06.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ПОСТАНОВКА ЗАДАЧІ

Тема курсового передбачає створення програмного забезпечення для обліку та зберігання інформації про осіб, які складали НМТ, їх результатів та створення сертифікатів та включає в себе такий функціонал:

1. Ведення обліку інформації про осіб які складали НМТ:

- Додавання інформації про учасника НМТ, шляхом введення типу паспорта, номеру паспорта, ПІН, номера свідоцтва про освіту, статі, дати народження, електронної пошти, контактного номеру телефону та необов'язкової помітки для екстрених випадків.

- Редагування інформації про учасника НМТ.

- Видалення інформації про учасника НМТ.

2. Ведення обліку інформації про умови тестування затверджені МОН:

- Додавання інформації про умови проведення тестування, шляхом введення статусу обов'язковості, дати укладання умов, прохідного балу, максимального можливого балу, мінімального можливого балу.

- Редагування інформації про умови проведення тестування.

- Видалення інформації умови проведення тестування.

3. Ведення обліку інформації про результати тестування:

- Додавання інформації про результати тестування, шляхом введення номеру паспорту учасника, навчального закладу, дати укладання умов тестування, назви навчальної дисципліни, результат тестування у балах, дата складання тестування.

- Редагування інформації про умови проведення тестування.

4. Ведення обліку інформації про навчальні дисципліни:

- Додавання інформації про навчальні дисципліни, шляхом введення назви предмету, короткого опису предмету, зразка тестування.

- Редагування інформації про умови навчальні дисципліни.

- Видалення інформації про умови навчальні дисципліни

5. Ведення обліку інформації про сертифікати НМТ:

	Вик.	Гуненко Я.М.			КП.ПЗ.211.06.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

- Додавання інформації про сертифікати НМТ, шляхом введення номеру паспорту учня, PIN-коду сертифікату, терміну дії сертифікату, статусу дії сертифікату, дати створення сертифікату, дати укладання умов тестування.

- Редагування інформації про сертифікати НМТ.

- Генерація сертифікатів.

6. Вимоги до операційної системи:

- Операційна система: Windows 10 x64.

- Процесор: Intel Celeron або еквівалент.

- Графічна підсистема: DirectX 10 і вище.

Таблиця 3.1 – Виділення інформаційних об'єктів предметної області

Особа	ID учня, номер паспорту, тип паспорту, ПІБ, дата народження, стать, E-mail, контактний номер телефону, Номер свідоцтва про освіту, примітка, ІПН
Навчальний заклад	Шифр навчального закладу, місто, область, тип, E-mail, ПІБ відповідального за проведення
Предмет	Код предмету, назва предмету, опис, зразок завдань
Умови проходження тестування з предмету	ID умови, код предмету, максимальний бал, мінімальний бал, мінімальний бал для проходження, статус, дата укладання вимог
Результат тестування	Шифр результату, ID умови, ID учня, отриманий бал, статус здачі, дата проходження, шифр навчального закладу
Сертифікат учасника НМТ	Номер сертифікату, ID учня, PIN-код, дата створення, термін дії, статус дійсності

2 ХАРАКТЕРИСТИКА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОЕКТУ

2.1 Опис середовища програмування

C++ Builder - це програмний продукт, інструмент швидкої розробки додатків (RAD), інтегроване середовище розробки (IDE) та система, що використовується програмістами для розробки програмного забезпечення мовою програмування C++. Visual C++ Builder дозволяє створювати графічні інтерфейси користувача шляхом перетягування компонентів з палітри інструментів на форми. За допомогою Builder можна створювати додатки для Windows, які використовують велику бібліотеку візуальних компонентів (VCL). C++ Builder автоматично генерує більшу частину коду програми, як тільки ви починаєте працювати над проектом. Для завершення решти логіки програми використовується текстовий редактор коду, який надає такі функції, як рефакторинг, паралельне редагування, завершення коду, збережені макроси натискання клавіш і власні комбінації клавіш. C++ Builder інтегрований в MSBuild як середовище збірки з командами збірки та компіляції, які викликають MSBuild; середовище RAD Studio надає мови програмування Delphi та C++ для розробки. Для курсових проектів корисним є C++ builder, оскільки він повністю підтримує мову програмування C++; середовище C++ builder підтримує найновіші стандарти C++, що дозволяє розробляти сучасне, ефективне програмне забезпечення. Вбудований компілятор ефективно оптимізує код і збільшує швидкість виконання програми.

Ще однією важливою особливістю цього середовища є підтримка високопродуктивних бібліотек: Visual Component Library (VCL) - це набір компонентів для побудови багатих графічних інтерфейсів користувача, VCL забезпечує швидкість і надійність. FireDAC - сучасна бібліотека доступу до даних, що підтримує широкий спектр баз даних; RTL (Runtime Library) - рутинна бібліотека часу виконання, що містить безліч корисних функцій для роботи з рядками, файлами, математичними операціями тощо.

	Вик.	Гуненко Я.М.			КП.ПЗ.211.06.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

C++ Builder дозволяє створювати додатки для Windows, macOS, Android та iOS з єдиної кодової бази; додатки, створені за допомогою C++ Builder, використовують нативні компоненти платформи, що забезпечує високу продуктивність та інтеграцію з операційною системою.

2.2 Опис мови програмування

Для виконання курсового проєкту було обрано мову програмування C++ з кількох вагомих причин. По-перше, C++ – це універсальна мова загального призначення, яка дозволяє створювати високопродуктивні, ефективні та надійні програмні системи. Її потужність полягає у можливості низькорівневого доступу до пам'яті та апаратних ресурсів, що робить її ідеальним вибором для розробки критичних до продуктивності додатків.

Однією з ключових особливостей C++ є підтримка об'єктно-орієнтованого програмування. Ця парадигма дозволяє моделювати реальний світ за допомогою класів та об'єктів. Класи визначають структуру даних та поведінку об'єктів, а об'єкти є екземплярами цих класів. Такий підхід сприяє модульності, повторюваності коду та легкості підтримки великих програмних систем. Як зазначає Б'ярн Страуструп, творець C++, "ООП дозволяє розробникам створювати більш складні та надійні системи, ніж це було б можливо за допомогою процедурного програмування".

Шаблони (templates) – це ще одна потужна особливість C++, яка дозволяє створювати універсальні алгоритми та структури даних. За допомогою шаблонів можна писати код, який працює з різними типами даних, не дублюючи його. Це значно зменшує обсяг коду та підвищує його надійність. Як зазначається в стандарті C++ (ISO/IEC 14882:2011), "шаблони є фундаментальною частиною мови C++ і дозволяють розробникам писати більш загальний та ефективний код".

STL – це багата колекція контейнерів, алгоритмів та ітераторів, яка є невід'ємною частиною сучасного C++. STL забезпечує ефективні та зручні засоби для роботи з даними, що значно спрощує розробку складних програмних систем.

	Вик.	Гуненко Я.М.			КП.ПЗ.211.06.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

Як зазначає Александре Степанов, один із розробників STL, "STL – це інструмент для створення ефективних та елегантних програм".

У курсовому проєкті мова програмування C++ дозволила структурувати програму за допомогою класів. Класи були створені для представлення різних доменних сутностей, що полегшує розуміння та підтримку коду. Для взаємодії з базою даних MySQL було використано функціонал C++, що забезпечує швидкий та надійний доступ до даних. Для роботи з динамічною пам'яттю використовувалися вказівники мови C++, низькорівневий доступ до пам'яті якої дозволив додатку працювати швидше.

2.3Опис СКБД

Для виконання курсового проєкту було використано СУБД MySQL. MySQL – це одна з найпопулярніших у світі систем управління реляційними базами даних з відкритим кодом. Її широко застосовують для створення веб-додатків, корпоративних систем та в багатьох інших сферах. MySQL здатна обробляти великі обсяги даних і одночасно обслуговувати тисячі запитів. Це досягається завдяки оптимізації запитів, ефективним алгоритмам сортування та індексування. MySQL забезпечує високий рівень доступності даних завдяки підтримці транзакцій, реплікації та механізмам відновлення після збоїв. Вона підтримує різні типи таблиць і механізми зберігання даних, що дозволяє адаптувати систему до конкретних потреб проєкту. СУБД сумісна зі стандартом SQL, що значно спрощує роботу з даними та дозволяє використовувати широкий спектр інструментів і бібліотек.

Вирішальними факторами для вибору СУБД MySQL для курсового проєкту стали її технології та можливості. InnoDB – основний рушій зберігання даних у MySQL, який забезпечує високу швидкість роботи, підтримку транзакцій, цілісність даних і масштабованість. MyISAM – ще один рушій зберігання даних, оптимізований для швидкого читання. Реплікація дозволяє створювати копії бази даних на інших серверах для підвищення доступності й резервування. Партіціювання – це можливість розбивати великі таблиці на менші частини для

	Вик.	Гуненко Я.М.			КП.ПЗ.211.06.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

ефективнішої обробки даних і покращення швидкодії. Тригери й збережені процедури автоматизують виконання певних дій при зміні даних у таблицях.

Для реалізації курсового проєкту було використано такі можливості MySQL: збережені процедури для автоматизації часто виконуваних завдань, тригери для автоматичного виконання дій у відповідь на зміну даних, представлення (віртуальні таблиці) для об'єднання даних з кількох таблиць, індекси для прискорення пошуку, транзакції для забезпечення цілісності даних, реплікація для підвищення доступності та резервування, а також кластеризація для масштабування бази даних.

2.4 Опис основних принципів ООП

Об'єктно-орієнтоване програмування (ООП) – це парадигма програмування, яка фокусується на створенні програм за допомогою об'єктів, що взаємодіють між собою. Основні принципи ООП – інкапсуляція, наслідування, поліморфізм та абстракція.

Інкапсуляція – це об'єднання даних і методів, що працюють з ними, в єдиний об'єкт. Це приховує внутрішню структуру об'єкта від зовнішнього світу, забезпечуючи безпеку даних та спрощуючи підтримку коду. В C++ інкапсуляція досягається за допомогою модифікаторів доступу: `private`, `protected` та `public`. Приклад фрагменту коду що демонструє інкапсуляцію продемонстровано у лістингу 2.1.

Лістинг 2.1 – Фрагмент коду

```
class Neuron {
private:
    double output;
    std::vector<double> weights;
public:
    double calculateOutput(const std::vector<double>& inputs) {
        // Розрахунок виходу нейрона
        // ...
        return output;
    }
};
```

	Вик.	Гуненко Я.М.			КП.ПЗ.211.06.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Наслідування дозволяє створювати нові класи (похідні) на основі існуючих (базових), успадковуючи їхні властивості та методи. Це забезпечує повторне використання коду та створення ієрархій класів. Типи наслідування в C++: `public`, `protected` та `private`. При публічному наслідуванні публічні члени базового класу стають публічними членами похідного класу, а захищені – захищеними. Це означає, що об'єкти похідного класу можуть безпосередньо звертатися до цих членів. Публічне наслідування зазвичай використовується для створення ієрархій класів, де похідні класи є більш специфічними видами базового класу. При захищеному наслідуванні як публічні, так і захищені члени базового класу стають захищеними членами похідного класу. Це означає, що до них можуть звертатися лише члени похідного класу та інших похідних класів. Захищене наслідування часто використовується для створення ієрархій класів, де похідні класи можуть розширювати базовий клас, але приховувати деякі деталі реалізації від зовнішнього світу. При приватному наслідуванні публічні і захищені члени базового класу стають приватними членами похідного класу. Це означає, що до них можуть звертатися лише члени самого похідного класу. Приватне наслідування рідко використовується, оскільки обмежує можливості повторного використання коду. Зазвичай використовується для повного приховування реалізації базового класу і створення нового, більш абстрактного класу. Приклад фрагменту коду що демонструє наслідування продемонстровано у лістингу 2.2.

Лістинг 2.2 – Фрагмент коду

```
class NeuralNetwork {
public: virtual void train(const
std::vector<std::pair<std::vector<double>, double>>& data) = 0;
};
class ConvolutionalNeuralNetwork : public NeuralNetwork { public:
void train(const std::vector<std::pair<std::vector<double>,
double>>& data) override {
    // Тренування згорткової нейронної мережі // ...
}
};
```

	Вик.	Гуненко Я.М.			КП.ПЗ.211.06.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

Поліморфізм – це здатність об'єктів різних типів відповідати на один і той самий запит по-різному. Це досягається за допомогою перевантаження функцій та віртуальних функцій. Поліморфізм забезпечує гнучкість та розширюваність програм. Приклад фрагменту коду що демонструє поліморфізм продемонстровано у лістингу 2.3.

Лістинг 2.3 – Фрагмент коду

```
class ActivationFunction {
public:
    virtual double activate(double z) = 0;
};

class Sigmoid : public ActivationFunction {
public:
    double activate(double z) override {
        // Функція сигмоїди
        // ...
    }
};

class ReLU : public ActivationFunction {
public:
    double activate(double z) override {
        // Функція ReLU
        // ...
    }
};
```

Абстракція – це процес фокусування на суттєвих характеристиках об'єкта, приховуючи деталі реалізації. Абстрактні класи та чисті віртуальні функції в C++ дозволяють створювати абстрактні інтерфейси, які не можуть бути інстанційовані безпосередньо. Приклад фрагменту коду що демонструє абстракцію продемонстровано у лістингу 2.4.

Лістинг 2.4 – Фрагмент коду

```
class Layer {
public:
    virtual void forward(const std::vector<double>& input) = 0;
    virtual void backward(const std::vector<double>& output_error) =
0;
};
```

	Вик.	Гуненко Я.М.			КП.ПЗ.211.06.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

2.5 Опис подібних програмних продуктів

Схожою за функціоналом є система my.testportal.gov.ua.

my.testportal.gov.ua — це веб-портал, призначений для організації та обліку результатів національних тестів. Система працює виключно онлайн, що означає, що будь-яка перерва в доступі до інтернету може спричинити затримку в роботі, оскільки програма не підтримує роботу в автономному режимі.

Платформа використовує власну базу даних для збереження інформації про учнів, навчальні заклади та результати тестувань. Проте, для більш гнучкої обробки даних немає можливості прямої інтеграції з СКБД на стороні користувача. Дані можна експортувати у форматі PDF, але це обмежує можливості подальшого аналізу та структуризації інформації.

Платформа також вимагає значного часу на налаштування, особливо для нових користувачів або навчальних закладів. Потрібен спеціаліст для розподілу ролей та доступів між адміністраторами, вчителями та учнями, оскільки самостійно це зробити складно без відповідного досвіду.

Висновок: my.testportal.gov.ua підходить для великих організацій, що мають відповідні ресурси для підтримки та адміністрування системи. Однак, для невеликих закладів або окремих користувачів платформа може виявитися занадто складною та вимагати значних зусиль для ефективного використання.

	Вик.	Гуненко Я.М.			КП.ПЗ.211.06.ПЗ	Арк.
	Пер.	Гапоненко Н.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

3.1 Опис бази даних

Для збереження великих об'ємів інформації потрібно розробити низку правил та відношень між цими даними. Відношення між таблицями показані на ER-діаграмі. Скрипт створення БД показаний у додатку Б. Зміст таблиць бази даних показаний у таблицях 3.2-3.9.

Таблиця 3.2 – Поля таблиці «Студент» (Student)

Назва поля	Опис	Тип	Розмір	Ключ
Student_id	Унікальний ідентифікатор студента	Числовий		Первинний
Passport_num	Номер паспорта	Символьний	До 20 символів	
Passport_type	Тип паспорта	Символьний	До 10 символів	
PIB	Прізвище, ім'я, по-батькові	Символьний	До 100 символів	
Birth_date	Дата народження	DATE	Стандарт ISO	
Gender	Стать	Символьний	1 символ	
E-mail	Електронна пошта	Символьний	До 100 символів	
Phone_num	Номер телефону	Символьний	До 15 символів	
EduCerf_num	Номер освітнього сертифіката	Символьний	До 30 символів	
PN	Ідентифікаційний код	Символьний	10 символів	
Additional	Додаткова інформація	Текстовий		

Таблиця 3.3 – Поля таблиці «Користувачі» (Users)

Назва поля	Опис	Тип	Розмір	Ключ
Login	Логін користувача	Символьний	До 50 символів	Первинний
Password	Пароль користувача	Символьний	До 50 символів	
Role	Роль користувача	Символьний	До 20 символів	

Таблиця 3.4 – Поля таблиці «Результат» (Result)

Назва поля	Опис	Тип	Розмір	Ключ
Res_id	Унікальний код результату	Числовий	Макс. значення – 2147483647	Первинний
Subj_id	Код предмета	Числовий		Зовнішній
Condition_id	Код умов	Числовий		Зовнішній
Student_id	Код студента	Числовий		Зовнішній
Reached_score	Досягнутий бал	Числовий		
Status	Статус	Логічний		
School_id	Код навчального закладу	Числовий		Зовнішній
Attemp_date	Дата спроби	DATE	Стандарт ISO	

Таблиця 3.5 – Поля таблиці «Сертифікат» (Certificate)

Назва поля	Опис	Тип	Розмір	Ключ
Cerf_num	Номер сертифіката	Числовий	Макс. значення – 2147483647	Первинний
Student_id	ID студента	Числовий		Зовнішній
PIN	PIN-код	Символьний	До 15 символів	
Creation_date	Дата створення	DATE	Стандарт ISO	
Efect_time	Термін дії	DATE	Стандарт ISO	
Status	Статус сертифіката	Логічний		

Таблиця 3.6 – Поля таблиці «Умови тестування» (Condition)

Назва поля	Опис	Тип	Розмір	Ключ
Condition_id	Унікальний код умови	Числовий	Макс. значення – 2147483647	Первинний
Subject_id	Код предмета	Числовий		Зовнішній
Max_point	Максимальний бал	Числовий		
Min_r_point	Мінімальний бал для проходження	Числовий		
Min_point	Мінімальний бал	Числовий		
Status	Обов'язкова умова	Логічний		
Date	Дата	DATE	Стандарт ISO	

Таблиця 3.7 – Поля таблиці «Навчальний заклад» ()

Назва поля	Опис	Тип	Розмір	Ключ
School_id	Унікальний код навчального закладу	Числовий	Макс. значення – 2147483647	Первинний
City	Місто	Символьний	До 50 символів	
Region	Область	Символьний	До 50 символів	
Type	Тип навчального закладу	Символьний	До 30 символів	
E-mail	Електронна пошта	Символьний	До 100 символів	
Teacher_PIB	Прізвище, ім'я, по-батькові відповідального	Символьний	До 100 символів	

Таблиця 3.8– Поля таблиці «Предмети» (Subject)

Назва поля	Опис	Тип	Розмір	Ключ
Subject_id	Унікальний код предмета	Числовий	Макс. значення – 2147483647	Первинний
Name	Назва предмета	Символьний	До 100 символів	
Description	Опис предмета	Текстовий		

ER-діаграма зв'язку таблиць бази даних наведена нижче на рисунку 3.1.

Вик.	Гуненко Я.М.				КП.ПЗ.211.06.ПЗ	Арк.
Пер.	Гапоненко Н.В.					15
Змн.	Арк.	№ докум.	Підпис	Дата		

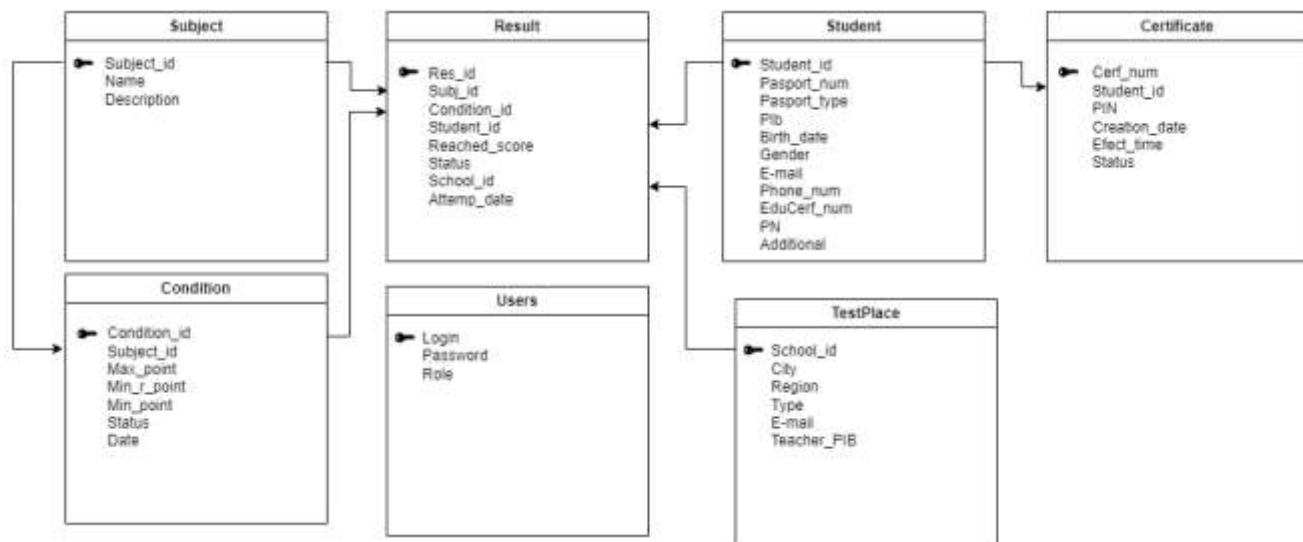


Рисунок 3.1 – ER діаграма бази даних

3.2 Проектування користувацького інтерфейсу програми

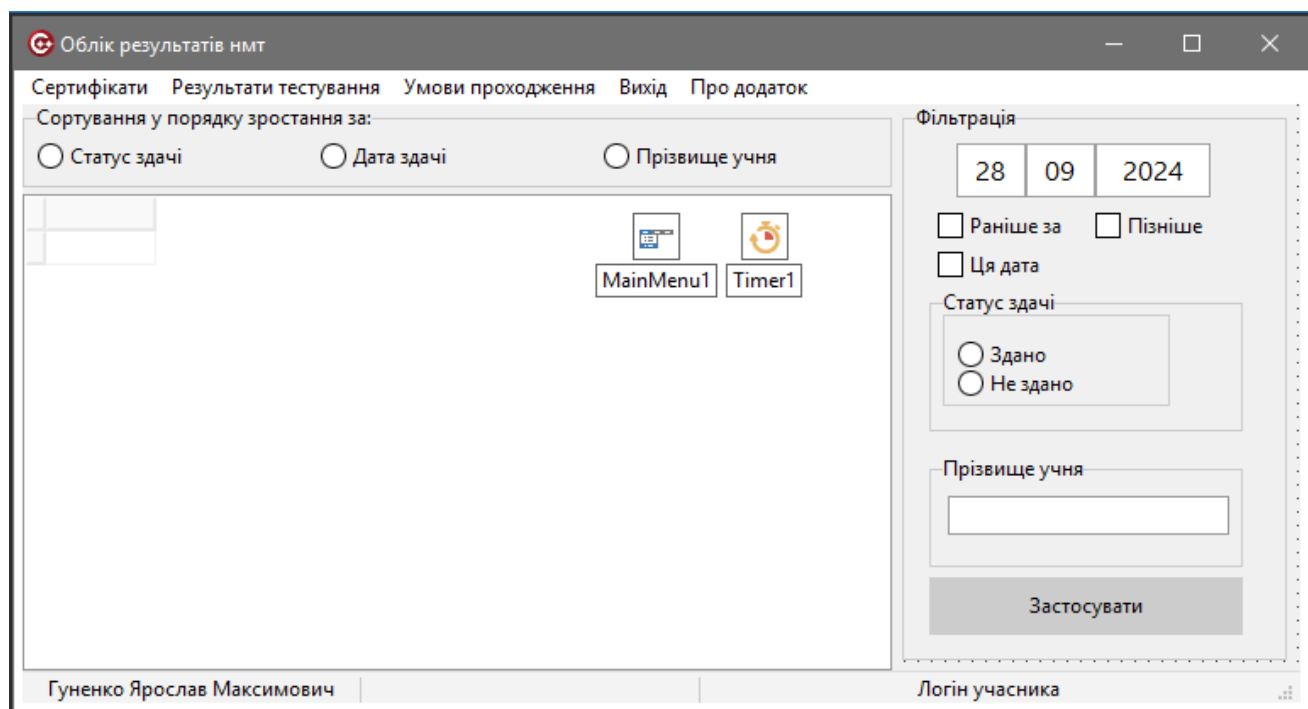


Рисунок 3.2 – Дизайн головного вікна

Вик.	Гуненко Я.М.			
Пер.	Гапоненко Н.В.			
Змн.	Арк.	№ докум.	Підпис	Дата

КП.ПЗ.211.06.ПЗ

Арк.

16

Додавання інформації про учня

Сертифікати Учні Умови проходження Предмети Результати тестування Про додаток

Поля для введення інформації про студента

Номер паспорта

Стать
☐ Чоловіча ☐ Жіноча

ІПН

Номер свідоцтва про освіту

Дата народження
 28 09 2024

Е-mail

Контактний телефон

Помітка(*)

Додати

Рисунок 3.3 – Робота з інформацією про учнів

Додавання інформації про наказ з умовами для пр...

Сертифікати Предмети Результати тестування Про додаток Учні Умови проходження

Поля для введення інформації про умови

☐ Обов'язково
☐ Не обов'язково

Максимальний бал

Мінімальний бал

Дата створення
 28 09 2024

Прохідний бал

Додати

Рисунок 3.4 – Робота з інформацією про умови тестування

Вик.	Гуненко Я.М.				КП.ПЗ.211.06.ПЗ	Арк.
Пер.	Гапоненко Н.В.					
Змн.	Арк.	№ докум.	Підпис	Дата		17

Додавання інформації про результати

Сертифікати Предмети Результати тестування Про додаток Учні Умови проходження

Поля для введення інформації про результати

Номер паспорту учня

Навчальний заклад

Навчальна дисципліна

Дата укладання умов 28 09 2024

Результат у балах

☐ Здано ☐ Не здано

Дата складання 28 09 2024

Додати

MainMenu1

Рисунок 3.5 – Робота з інформацією про результати тестування

Додавання інформації про предмети

Сертифікати Предмети Результати тестування Про додаток Учні Умови проходження

Поля для введення інформації про студента

Назва предмету

Опис предмету

Додати

MainMenu1

Рисунок 3.6 – Робота з інформацією про навчальні дисципліни

Сертифікати

СертифікатиПредметиРезультати тестуванняПро додатокУчніУмови проходження

MainMenu1

Поля для введення інформації про сертифікати

Номер паспорту учня

PIN-код

Термін дії
☐ Дійсний ☐ Не дійсний

Дата укладання умов
03102024

Дата створення
28092024

Додати

Про программу

— □ ×

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi sed nunc ornare, faucibus leo vel, eleifend dolor. Praesent est nunc, bibendum quis ipsum et, faucibus feugiat enim. Pellentesque gravida fringilla ante eu varius. Proin vel neque iaculis lacus ultricies aliquet. Ut fermentum est et massa sodales, non congue purus auctor. Ut a rhoncus quam. Aenean et nunc ligula. Quisque mattis magna eu mauris ornare, eget tincidunt risus imperdiet.


```

CREATE DATABASE NMT_results;
USE NMT_results;

-- Створення таблиці "Student"
CREATE TABLE Student (
    Student_id INT PRIMARY KEY AUTO_INCREMENT,
    Passport_num VARCHAR(20),
    Passport_type VARCHAR(10),
    PIB VARCHAR(100),
    Birth_date DATE,
    Gender CHAR(1),
    Email VARCHAR(100),
    Phone_num VARCHAR(15),
    EduCerf_num VARCHAR(30),
    PN VARCHAR(10),
    Additional TEXT
);

-- Створення таблиці "Users"
CREATE TABLE Users (
    Login VARCHAR(50) PRIMARY KEY,
    Password VARCHAR(50),
    Role VARCHAR(20)
);

-- Створення таблиці "Result"
CREATE TABLE Result (
    Res_id INT PRIMARY KEY AUTO_INCREMENT,
    Subj_id INT,
    Condition_id INT,
    Student_id INT,
    Reached_score INT,
    Status BOOLEAN,
    School_id INT,
    Attemp_date DATE,
    FOREIGN KEY (Subj_id) REFERENCES Subject(Subject_id),
    FOREIGN KEY (Condition_id) REFERENCES Condition(Condition_id),
    FOREIGN KEY (Student_id) REFERENCES Student(Student_id),
    FOREIGN KEY (School_id) REFERENCES School(School_id)
);

-- Створення таблиці "Certificate"
CREATE TABLE Certificate (
    Cerf_num INT PRIMARY KEY AUTO_INCREMENT,
    Student_id INT,
    PIN VARCHAR(15),
    Creation_date DATE,
    Effect_time DATE,
    Status BOOLEAN,
    FOREIGN KEY (Student_id) REFERENCES Student(Student_id)
);

-- Створення таблиці "Condition"
CREATE TABLE Condition (
    Condition_id INT PRIMARY KEY AUTO_INCREMENT,
    Subject_id INT,

```

```

    Max_point INT,
    Min_r_point INT,
    Min_point INT,
    Status BOOLEAN,
    Date DATE,
    FOREIGN KEY (Subject_id) REFERENCES Subject(Subject_id)
);

-- Створення таблиці "School"
CREATE TABLE School (
    School_id INT PRIMARY KEY AUTO_INCREMENT,
    City VARCHAR(50),
    Region VARCHAR(50),
    Type VARCHAR(30),
    Email VARCHAR(100),
    Teacher_PIB VARCHAR(100)
);

-- Створення таблиці "Subject"
CREATE TABLE Subject (
    Subject_id INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(100),
    Description TEXT
);

```