



# НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем

Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

Освітній ступінь бакалавр

Спеціальність 122 «Комп'ютерні науки»  
(код і назва)

Освітньо-професійна програма «Комп'ютерні науки»  
(назва)

## ЗАТВЕРДЖУЮ

Завідувач кафедри Грибков С.В.

«15» квітня 2024 року

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Шкарупа Андрій Вікторович

(прізвище, ім'я, по батькові)

Тема роботи: Розроблення інформаційної системи для підтримки навчального процесу ІТ-школи Robocode ФОП «ЮШКОВ Дмитро Сергійович»

1. керівник роботи Литвинов Валерій Андроникович, проф., док. технічних наук,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 15 квітня 2024 року № 279-кс

2. Строк подання здобувачем роботи: 03.06.2024

3. Вихідні дані до роботи: дані про організаційну структуру ІТ-школи Robocode ФОП «ЮШКОВ Дмитро Сергійович» та відділу освіти, дані про викладачів/категорії посад/слухачів/групи/курси/напрямки/чеки

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):  
системний аналіз діяльності відділу освіти ІТ-школи Robocode ФОП «ЮШКОВ Дмитро Сергійович», технічне завдання на розробку системи, задача автоматизації та їх вирішення, заходи з охорони праці та навколишнього середовища

5. Перелік графічного матеріалу:

1. Організаційна структура ФОП «ЮШКОВ Дмитро Сергійович»

2. Контекстна діаграма функціональної моделі (AS-IS) та діаграми декомпозиції

3. Контекстна діаграма концептуальної моделі (TO-BE) та діаграми декомпозиції

4. Логічний та фізичний рівень моделі даних

5. Схема бази даних у MS SQL Server

6. Скріншоти інтерфейсу системи

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Вступ	д.т.н., професор Литвинов В.А.		
I	д.т.н., професор Литвинов В.А..		
II	д.т.н., професор Литвинов В.А.		
III	д.т.н., професор Литвинов В.А.		
IV	д.т.н., професор Литвинов В.А.		
Висновок	д.т.н., професор Литвинов В.А.		

7. Дата видачі завдання: 15 квітня 2023 року

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Написання вступної частини	29.04.2023	виконано
2	Системний аналіз діяльності роботи університету	01.05.2023	виконано
3	Розроблення функціональної моделі	04.05.2023	виконано
4	Аналіз систем-аналогів розробки	07.05.2023	виконано
5	Вибір програмного забезпечення для розробки системи	08.05.2023	виконано
6	Розробка логічної та фізичної моделей бази даних	10.05.2023	виконано
7	Генерація бази даних в MySQL Server	12.05.2020	виконано
8	Розробка інтерфейсу користувача	14.05.2023	виконано
9	Написання інструкції користувача	21.05.2023	виконано
10	Розробка заходів з охорони праці	23.05.2023	виконано
11	Оформлення пояснювальної записки	25.05.2023	виконано
12	Оформлення презентації	26.05.2023	виконано

Здобувач

\_\_\_\_\_

(підпис)

Шкарупа А. В.

\_\_\_\_\_

(прізвище та ініціали)

Керівник роботи

\_\_\_\_\_

(підпис)

Литвинов В. А.

\_\_\_\_\_

(прізвище та ініціали)

## АНОТАЦІЯ

Дипломна робота на тему «Розроблення інформаційної системи для підтримки навчального процесу ІТ-школи Robocode ФОП «Юшков Дмитро Сергійович»» розроблена студентом 4 курсу групи 4-5ск Шкарупою А.В.

Записка складається з 118 сторінок, 13 таблиць, 48 рисунків, 5 додатків та 21 літературних джерел.

На основі проведених досліджень під час проходження виробничої та переддипломної практики в ІТ-школі Robocode – відділу освіти, доведено потребу в розробці інформаційної системи для керуванням навчальним процесом. Робота включає вибір програмного забезпечення, приклади алгоритмів за допомогою яких реалізовано функціонал системи, та розроблену інструкцію користувача. Також проведено аналіз впровадження системи з техніко-економічної точки зору, описані заходи з охорони праці для працівників підприємства.

Ключові слова: МОДЕЛЬ, ІТ-ШКОЛА, СИСТЕМА, ROBOCODE, МОДЕЛЬ IDEF0, ДЕКОМПОЗИЦІЯ, БЕЗПЕКА, КУРСИ, ІНТЕРФЕЙС, ІНФОРМАЦІЙНА СИСТЕМА, MS VISUAL STUDIO 2019, C#, ПЗ, ПРОГРАМНИЙ ДОДАТОК, MS SQL 2019.

## SUMMARY

The thesis on the topic "Development of an information system to support the educational process of the IT-school Robocode FOP «Yushkov Dmytro Serhiyovych»» was developed by a student of the 4th year of the group 4-5sk Shkarupa A.V.

The note consists of 118 pages, 13 tables, 48 figures, 5 appendices and 21 literary sources.

On the basis of research carried out during industrial and pre-diploma practice at the IT-school Robocode - Department of Education, the need for the development of an information system for managing the educational process has been proven. The work includes the selection of software, examples of algorithms with the help of which the functionality of the system is implemented, and the developed user manual. An analysis of the implementation of the system from a technical and economic point of view was also carried out, and measures for labor protection for the company's employees were described.

Keywords: MODEL, IT-SCHOOL, SYSTEM, ROBOCODE, IDEF0 MODEL, DECOMPOSITION, SECURITY, COURSES, INTERFACE, INFORMATION SYSTEM, MS VISUAL STUDIO 2019, C#, SOFTWARE, SOFTWARE APPLICATION, MS SQL 2019.

## ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ІТ-ШКОЛИ ROBOCODE ФОП «ЮШКОВ ДМИТРО СЕРГІЙОВИЧ» .....	9
1.1. Загальна характеристика ІТ-школи Robocode.....	9
1.2. Організаційна структура ІТ-школи Robocode, роль і взаємодія підрозділів .....	10
1.3. Аналіз нинішнього стану комп'ютеризації ІТ-школи Robocode .....	13
1.4. Розроблення функціональної моделі та аналіз існуючих бізнес- процесів .....	13
1.5. Огляд існуючих рішень для розв'язання виявлених проблем.....	17
1.6. Обґрунтування доцільності проектування й розроблення інформаційної системи для підтримки навчального процесу ІТ-школи Robocode .....	23
1.7. Концептуальна модель системи.....	24
1.8. Розрахунок економічного ефекту від впровадження системи .....	24
1.9. Висновок до першого розділу .....	31
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ.....	32
2.1. Загальні положення.....	32
2.2. Призначення і цілі створення системи.....	32
2.3. Характеристика об'єкта автоматизації .....	32
2.4. Вимоги до системи.....	33
2.5. Склад і зміст робіт по створенню системи .....	44
2.6. Порядок контролю і приймання системи .....	44
2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію .....	45
2.8. Вимоги до документації .....	45
2.9. Джерела розробки .....	45
РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ.....	46
3.1. Інформаційне забезпечення системи.....	46

3.2. Алгоритмізація та реалізація комплексу задач автоматизації.....	49
3.3. Інструкція користувача.....	61
3.4. Технічне та системне забезпечення розробки.....	69
РОЗДІЛ 4. ОХОРОНА ПРАЦІ .....	73
4.1. Вимоги до приміщення.....	73
4.2. Вимоги безпеки роботи з технікою .....	75
4.3. Вимоги безпеки життєдіяльності при аварійних ситуаціях.....	76
4.4. Рекомендації до роботи за комп'ютерною технікою .....	77
4.5. Вимоги пожежної безпеки.....	78
ВИСНОВКИ.....	80
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	81
Додаток А.....	83
Додаток Б.....	86
Додаток В .....	80
Додаток Г .....	92
Додаток Д.....	105

## ВСТУП

У сучасному світі управління приватними навчальними закладами вимагає комплексного підходу та використання передових інструментів, серед яких ключове місце займають інформаційні системи [1]. Інформаційні технології та системи управління дозволяють оптимізувати різноманітні бізнес-процеси та підвищувати ефективність управлінських рішень.

Переваги використання інформаційних систем у сфері управління приватними навчальними закладами очевидні. Ці системи допомагають в автоматизації багатьох аспектів діяльності закладу, від обліку фінансів до взаємодії зі студентською аудиторією та партнерами. Завдяки їм, керівництво може ефективно планувати розвиток закладу, забезпечувати якісну освіту та конкурентоспроможність на ринку освітніх послуг.

Враховуючи швидкий темп розвитку технологій та зростання вимог до якості освіти, впровадження інформаційних систем стає необхідністю для приватних навчальних закладів. Вони дозволяють не лише оптимізувати бізнес-процеси, а й відстежувати рівень взаємодії зі студентами, залучати нових клієнтів та забезпечувати ефективне управління ресурсами.

Метою кваліфікаційної роботи є розроблення інформаційної системи для підтримки навчального процесу ІТ-школи Robocode. Ця інформаційна система буде відрізнятися легкістю в управлінні, дозволяючи керівництву та викладачам з легкістю відстежувати та контролювати різноманітні процеси в школі. Зручний інтерфейс системи забезпечить легкість в користуванні системою для всіх її користувачів, що сприятиме підвищенню ефективності.

## **РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ІТ-ШКОЛИ ROBOCODE ФОП «ЮШКОВ ДМИТРО СЕРГІЙОВИЧ»**

### **1.1. Загальна характеристика ІТ-школи Robocode**

ІТ-школа Robocode – це відомий освітній заклад, який заснований в Україні у 2016 році і вже став одним із провідних у галузі навчання робототехніці та програмування. Школа спеціалізується на комплексному навчанні дітей у віці від 8 до 16 років, зосереджуючись на розвитку їх навичок в сфері робототехніки, програмування та проектного мислення [2].

Відділ досліджень і розробки Robocode активно працює над створенням інноваційних роботів для навчальних, сервісних та розважальних цілей. Школа використовує різноманітні методи навчання, включаючи практичні заняття та проектні завдання, щоб залучити учнів до цікавої та ефективної навчальної діяльності.

Загальна мета навчальних програм Robocode полягає в тому, щоб надати дітям можливість вивчати сучасні технології, розвивати творчість та логічне мислення, а також навчити їх співпраці та тимчасового управління проектами. Школа Robocode допомагає молодому поколінню підготуватися до вимогливого цифрового світу та розвивати ключові навички для успішної майбутньої кар'єри.

Аналізуючи вище сказане, можна виділити основні види діяльності школи Robocode:

- створення курсів з програмування;
- виготовлення електроніки для збирання схем;
- розробка роботизованих систем, для програмування;
- виробництво одягу з унікальним брендом компанії;
- організація змагань з програмування;
- проведення таборів для дітей.

На даний момент, компанія займається такими напрямками програмування, а також створенням курсів що в входять до них: робототехніка та програмування (Arduino Kids, Arduino Junior, Arduino KidsPro, Middle Embedded, Pro Embedded,

Senior Embedded, Expert Embedded), розробка ігор (Construct, Unity Beginner, Unity 2D, Unity 3D, Unity Pro), розробка сайтів (Web Beginner, Web Junior, Web Middle), програмування Python (Python Beginner), розробка програмного забезпечення (Java Beginner, Java Junior, Java Middle) [3].

## 1.2. Організаційна структура ІТ-школи Robocode, роль і взаємодія підрозділів

### 1.2.1. Загальна схема організаційної структури

Організаційна схема підприємства, яка узагальнено відображає його структуру та складові елементи представлена на рисунку 1.1.

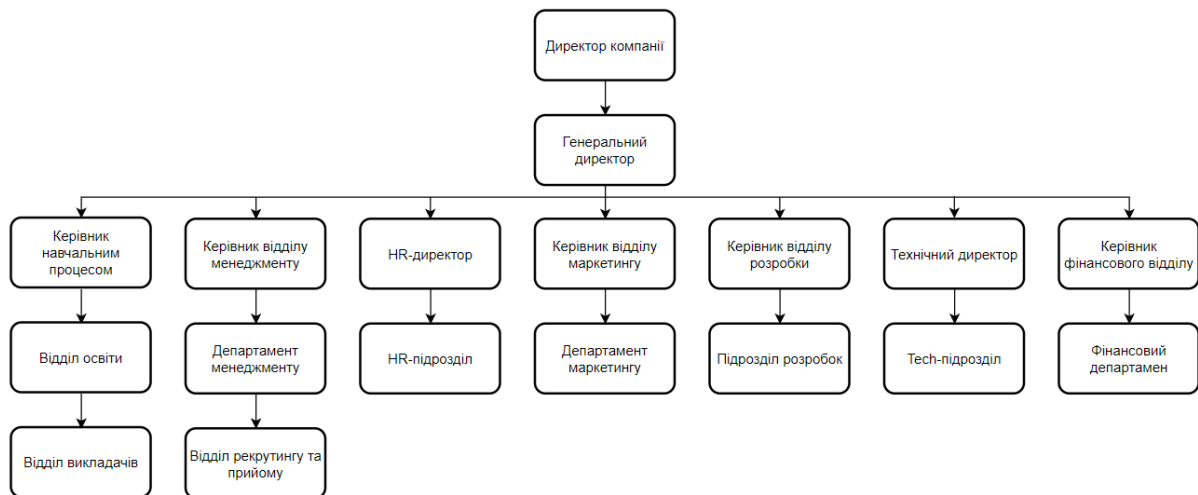


Рисунок 1.1 – Схема організаційної структури ІТ-школи Robocode

Організаційна структура виступає ключовим аспектом діяльності підприємства, оскільки вона визначає розподіл завдань між різними підрозділами, їх ієрархічність та спеціалізацію у конкретних сферах виробництва.

У центрі організаційної структури стоїть директор компанії, який частково розподіляє свої обов'язки генеральному директору, а йому в свою чергу, підпорядковується керівник навчальним процесом, керівник відділу менеджменту, HR-директор, керівник відділу маркетингу, керівник відділу розробки, технічний директор та керівник фінансового відділу [4]. Отримавши

завдання від генерального директора, керівники дають наказ своєму відділу, що до його виконання.

### 1.2.2. Структура відділу освіти

Особливу увагу приділяємо відділу освіти, оскільки він відіграє ключову роль у функціонуванні організації. Схема відділу освіти зображена на рисунку 1.2.



Рисунок 1.2 – Схема відділу освіти ІТ-школи Robocode

В цьому відділі відбувається розробка навчального матеріалу та перевірка його на відповідність. До матеріалу відносяться презентації з певного курсу програмування. Крім цього, цей відділ займається організацією навчального процесу. Більш детальні функції відділу освіти можна побачити в таблиці 1.1.

Таблиця 1.1 Завдання та функції відділу освіти

№	Задачі	Функції
1	Розробка навчальних програм та курсів	- перевірка презентацій на відповідність тематиці курсу з програмування - перевірка матеріалів на новітність інформації
2	Вибір викладачів	- перевірка тестових завдань для визначення рівня знань викладача - пошук викладача для проведення курсів
3	Організація навчального процесу	- встановлення графіку навчальних занять - створення розкладу для набраних груп дітей

Звичайно, для нормально функціонування відділу освіти, йому потрібна допомога інших відділів. Взаємодію відділу освіти з іншими підрозділами та відділами представлено в таблиці 1.2.

Таблиця 1.2 Взаємодія відділу освіти з іншими підрозділами та відділами

№	Підрозділ	Одержання	Надання
1	Департамент маркетингу	- запит на рекламування школи	- реклама в соц-мережах, рекламні матеріали для проведення відкритих уроків
2	Департамент менеджменту	- запит на створення нових груп - список потрібної техніки для навчання	- послуги щодо створення пробних занять - формування заявки на отримання техніки для навчання

*Продовження таблиці 1.2*

3	Tech- підрозділ	- список поламаной техніки (зламаної техніки)	- полагожену техніку для подальшого використання
4	Фінансовий департамент	- відомості про відпрацьовані дні працівників	- квитанцію про оплату послуг

### **1.3. Аналіз нинішнього стану комп'ютеризації ІТ-школи Robocode**

Наразі ІТ-школа Robocode використовує власну CRM систему, в яку інтегрована інформаційна система для управління навчальним процесом. Однак, існують певні недоліки, які вимагають уваги. Зокрема, інтерфейс цієї системи є складним для сприйняття, а також деякі процеси не повністю автоматизовані. Наприклад, відсутнє автоматичне підрахування кількості проведених занять викладачем. Наприклад, відсутня можливість автоматичного підрахунку кількості проведених занять викладачем, що ускладнює процес обліку та аналізу даних.

## **1.4. Розроблення функціональної моделі та аналіз існуючих бізнес-процесів**

### **1.4.1. Функціональна модель ІТ-школи Robocode**

Функціональна модель побудована з використанням методології SADT за допомогою CASE-засобу ERWin Process Modeler. CASE-засіб ERWin Process Modeler використовується для проектування, аналізу, документування та оптимізації бізнес процесів [5].

Система описана за допомогою IDEF0, яка по іншому називається функціональною моделлю. Ця модель може використовувати як текстові, так і графічні елементи для опису існуючих бізнес-процесів. Однією з ключових особливостей методології IDEF0 є її акцент на взаємозв'язках між різними елементами системи, а не їх часова послідовність виконання [6].

Загальна діаграма, що відображає діяльність відділу освіти, а саме проведення та організацію занять, зображена на рисунку 1.3 (номер моделі А- 0). Далі представлений перелік вхідних та вихідних даних, механізмів керування та управління, які взаємодіють у бізнес-процесах.

*Входи:*

- викладачі;
- слухачі;
- перелік допустимих курсів.

*Виходи:*

- розклад проведення занять;
- список призначених викладачів;
- відомості про зареєстрованих слухачів;
- сертифікати про успішне проходження курсів;
- квитанція з оплати курсу;
- список відкритих курсів;
- звіт про дохід.

*Управління:*

- завдання адміністрування;
- відкриті курси;
- правила компанії;
- ставка за урок.

*Механізми керування:*

- розподілені викладачі;
- матеріали;
- комп'ютерна техніка.

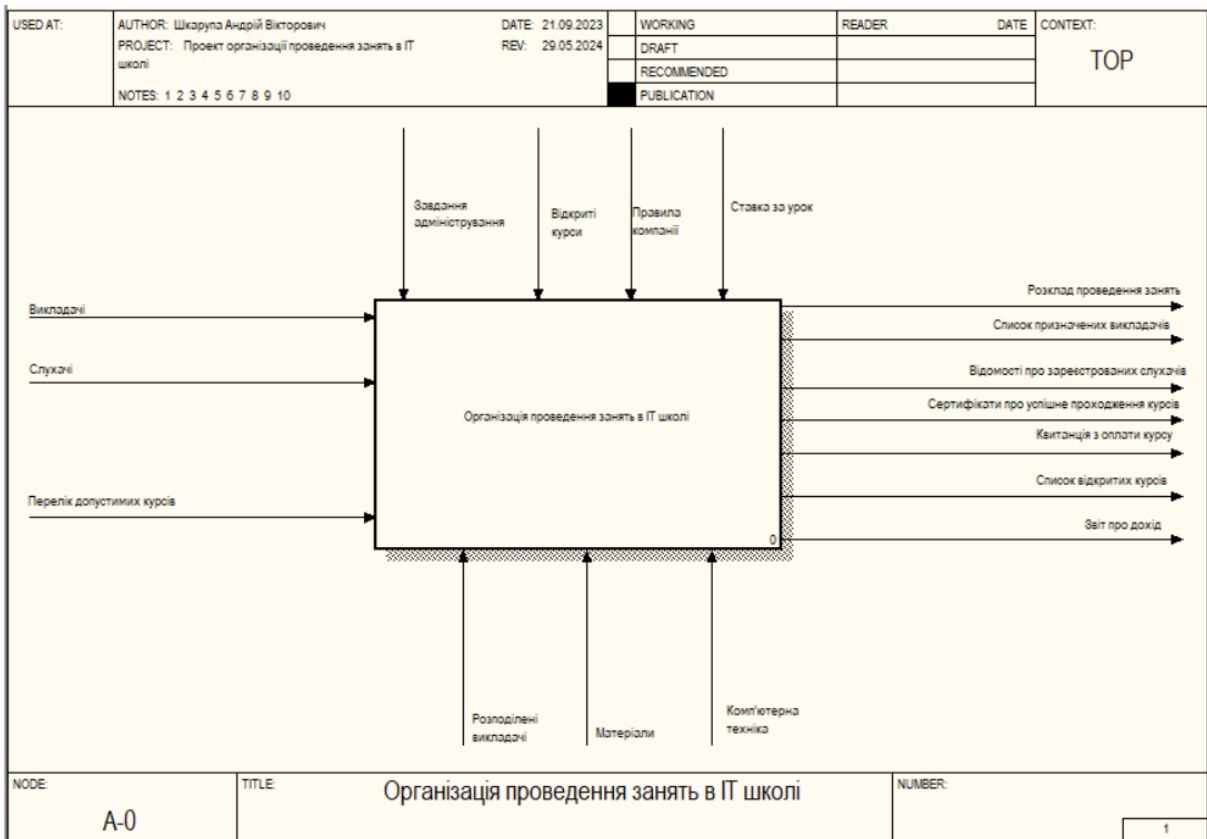


Рисунок 1.3 – Організаційна структура проведення занять, яким займається відділ освіти

Очевидно, що поданий рисунок діаграми верхнього рівня моделі, недостатньо інформативний для повного розуміння роботи об'єкту моделювання, так як він описує систему в цілому.

Таким чином, необхідно розглянути декомпозицію моделі, що наводиться в додатку А, рисунок А.1-5. Де детально відображається процес проведення занять, яким займається відділ освіти, а саме:

- відкриття курсу:
  - відбір викладачів;
  - створення пробного заняття з обраного напрямку;
  - впровадження курсу в дію.
- формування груп та складання розкладу:
  - отримання заявки від слухачів на навчання;
  - складання групи;
  - складання розкладу для групи.

- проведення занять;
- підрахунок виручки за проведені заняття;
- оцінювання слухачів:
  - проходження тестових завдань;
  - формування оцінки за курс;
  - видача сертифікату.
- формування звітів за період навчання.

Як результат, функціональна модель демонструє заходи, які мають виконувати співробітники, для забезпечення ефективної організації навчального процесу. Модель відповідає на такі питання:

- Як плануються та розподіляються курси для учнів?
- Як відбувається реєстрація учнів на курс?
- Як розподіляються викладачі для проведення занять?
- Як визначається успіх учнів?
- Як контролюється присутність учнів?
- Які документи (звіти) потрібні для аналізу ефективності діяльності школи?

#### **1.4.2. Виявлені проблеми**

Під час аналізу діяльності відділу були виявлені наступні недоліки:

- нинішня організаційна структура відділу освіти є застарілою;
- структура наявної бази даних не є правильно спроектованою;
- інформація про підвищення категорії викладача фіксується в окремому файлі, а не в системі;
- для оцінювання учнів використовується стороння система, що не є сильно зручним.

#### **1.4.3. Задачі автоматизації**

Для підвищення ефективності та оптимізації роботи відділу освіти, вирішено автоматизувати ключові процеси, що включають:

- отримання оперативної інформації про хід проведення занять та роботи викладачів;

- миттєве додавання або внесення змін до бази даних про студентів, викладачів та курси;
- розробка та впровадження нової, більш зручної та функціональної бази даних, яка відповідає потребам відділу освіти.

## 1.5. Огляд існуючих рішень для розв'язання виявлених проблем

### 1.5.1. Платформа Moodle

Moodle – це відкрите програмне забезпечення для управління навчанням, що використовується в освітніх установах для організації навчального процесу [7]. Інтерфейс платформи Moodle зображено на рисунку 1.4.

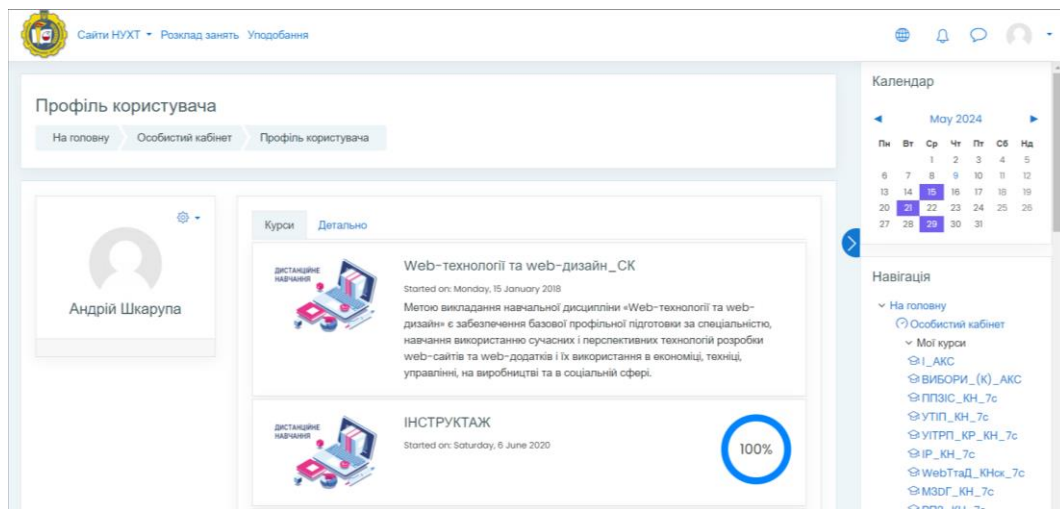


Рисунок 1.4 – Інтерфейс платформи Moodle

Ця платформа має такий функціонал:

- викладачі можуть створювати курси, завантажувати навчальні матеріали, створювати завдання та тести;
- вбудовані засоби комунікації, студенти можуть обговорювати теми, спілкуватися з викладачами та одногрупниками, обмінюватися матеріалами та робити спільні проекти;
- вбудовані інструменти для оцінювання студентів, включаючи тести, завдання та звіти про успішність;
- відстеження прогресу, викладачі можуть відстежувати прогрес студентів, переглядати їхні результати та надавати зворотний зв'язок.

Недоліки платформи Moodle:

- складність установки;
- обмежена інтеграція з іншими системами;
- обмежена швидкодія.

### 1.5.2. Платформа Prosvita

Платформа Prosvita - це інноваційний освітній простір, який надає широкий спектр функціоналу для організації навчального процесу [8]. Інтерфейс платформи Prosvita представлено на рисунку 1.5.

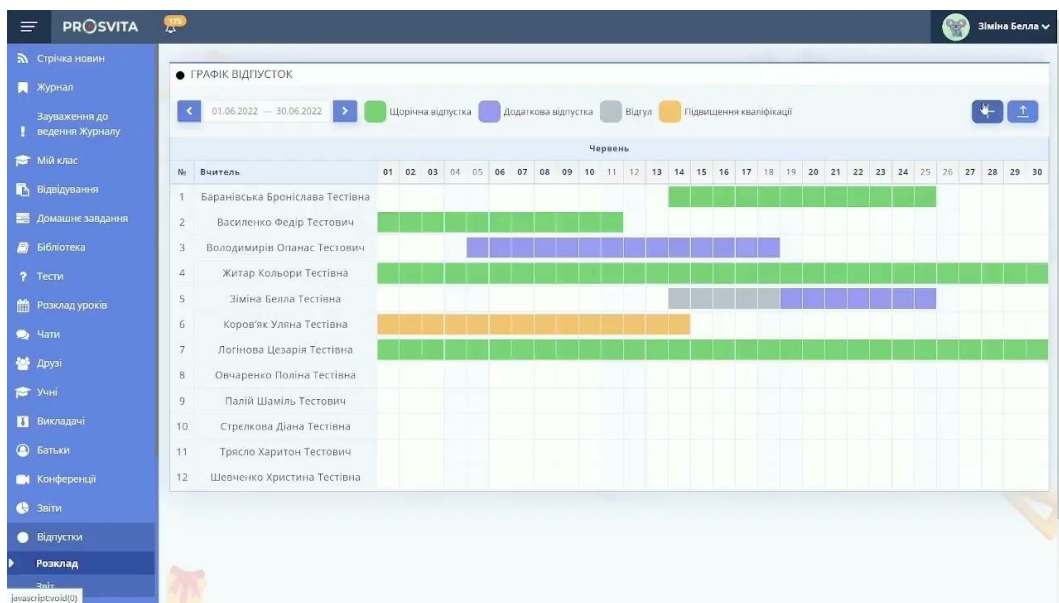


Рисунок 1.5 – Інтерфейс платформи Prosvita

Основні функції та можливості цієї платформи включають:

- налаштування розкладу занять, дзвінків, планування уроків та синхронізація з щоденником учнів;
- формування класів та підгруп, внесення особистої інформації про учнів, автоматичне переведення до наступного класу тощо;
- автоматичне закриття року, звітність по відвідуванню учнів, формування паперової версії журналу відповідно до вимог Міністерства освіти, мережа класів, зауваження до ведення журналів;

Недоліки платформи Prosvita:

- для приватних шкіл тариф складає 40000 тисяч грн в рік;

— обмежена можливість налаштування деяких параметрів для окремих закладів освіти;

### 1.5.3. Платформа Canvas

Платформа Canvas є однією з найвідоміших систем управління навчанням та використовується в освітніх установах по всьому світу для організації навчального процесу [9].

Розроблена компанією Instructure, Canvas став популярним інструментом для навчання в багатьох країнах, включаючи США, Канаду, Великобританію, Австралію та інші.

Завдяки своїй простоті використання, гнучкості та багатофункціональності, вона отримала широке визнання серед освітян і стала популярним вибором для університетів, коледжів, шкіл та інших освітніх установ. Інтерфейс платформи Canvas зображено на рисунку 1.6.

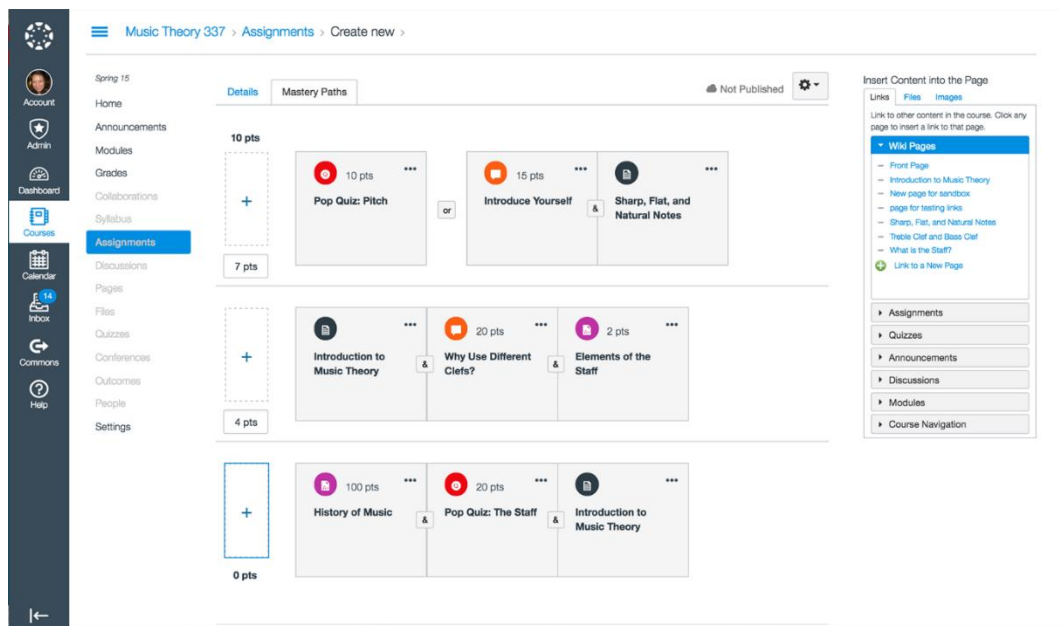


Рисунок 1.6 – Інтерфейс платформи Canvas

Canvas надає такий функціонал:

— завантаження різноманітних навчальних матеріалів, такі як текстові документи, відео, зображення, презентації;

- викладачі можуть створювати й керувати курсами, встановлювати терміни та обов'язкові завдання, створювати розклади та встановлювати специфічні правила для кожного курсу;
- надання завдань студентам, а також оцінювання їхньої роботи. Викладачі можуть встановлювати терміни здачі, оцінювати роботи та вести журнал оцінок;
- платформа дозволяє студентам та викладачам спілкуватися через форуми, чати та електронні листи, а також співпрацювати над завданнями та проектами;
- викладачі можуть створювати групи для співпраці над конкретними завданнями або проектами та керувати доступом до ресурсів та матеріалів.

Недоліки платформи Canvas:

- відсутність української мови в інтерфейсі;
- висока вартість;
- обмежена інтеграція з іншими системами.

#### 1.5.4. CRM система crm.robocode

Для керуванням навчальним процесом, ІТ-школа Robocode використовує власну CRM-систему crm.robocode. Крім цього, в системі є функції що стосуються бізнесу. Інтерфейс CRM-системи представлена на рисунку 1.7.

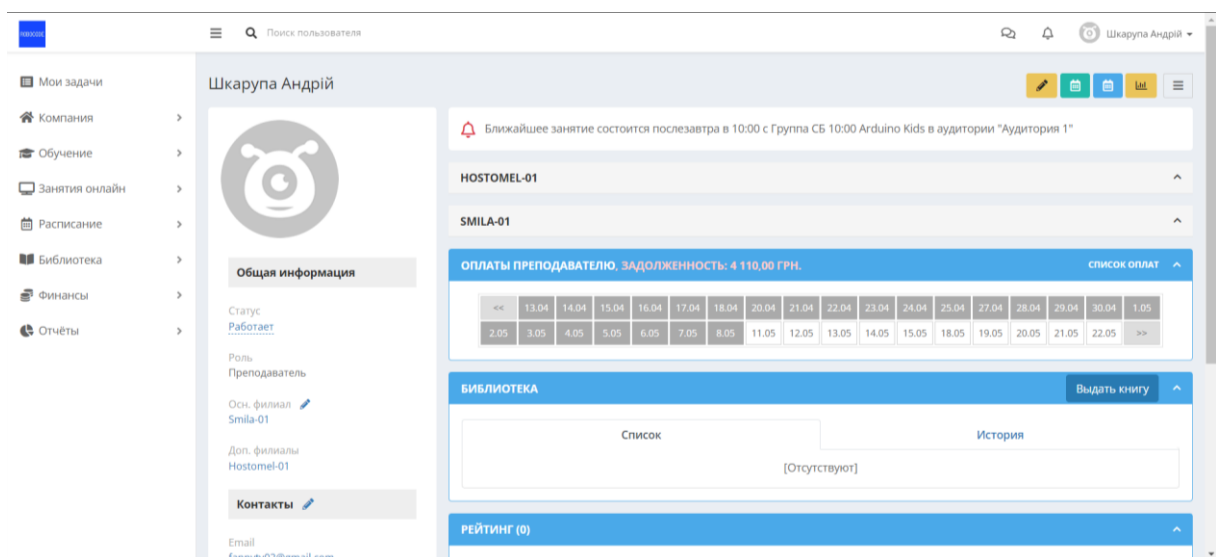


Рисунок 1.7 – Інтерфейс CRM-системи crm.robocode

Система містить такий функціонал:

- зберігання та ведення інформації про групи та курси, що дозволяє керівництву школи з легкістю контролювати всі аспекти навчального процесу;
- облік даних про користувачів, що включає інформацію про студентів, викладачів та інших співробітників;
- ведення інформації про викладання курсів, що дозволяє відстежувати активність викладачів та ефективність їхньої роботи;
- перегляд звітності про успішність учнів, для аналізу навчальних результатів та виявлення потенційних областей для покращення;
- оцінка учнів, що дозволяє відслідковувати академічний прогрес кожного студента;
- обрахунок заробітної плати викладачів, що спрощує процес фінансового управління школою.

Недоліками цієї системи є:

- складність використання;
- вартість;
- відсутність перегляду транзакцій при сплаті курсу;

### **1.5.5. Порівняльна характеристика систем**

Для кожної з наведених систем характерні як переваги, так і недоліки в порівнянні з іншими. Щоб отримати більш об'єктивну оцінку, необхідно скласти порівняльну таблицю. Результати порівняння аналогічних систем наведено в таблиці 1.3.

Таблиця 1.3. Порівняння систем аналогів

<b>Критерії оцінки / функції</b>	<b>Moodle</b>	<b>Prosvita</b>	<b>Canvas</b>	<b>crm.robocode</b>
Управління клієнтською базою	3/5	4/5	2/5	4/5
Зручність інтерфейсу	4/5	5/5	4/5	3/5
Ведення інформації про групи	4/5	4/5	3/5	5/5
Ведення інформації про курси	3/5	4/5	3/5	5/5
Ведення інформації про користувачів	3/5	4/5	4/5	5/5
Ведення інформації про викладання курсів	2/5	4/5	2/5	4/5
Реєстрація дитини в нову групу	3/5	5/5	4/5	4/5
Функції пов'язані з фінансовими операціями	0/5	4/5	0/5	4/5
Онлайн оплата	0/5	4/5	0/5	5/5

Підсумуємо загальну кількість балів, кожної системи. Платформа Moodle набрала 22 бали, Prosvita – 38, Canvas – 22, crm.robocode – 39. Також аналізуючи таблицю, слід зауважити, що деякі з систем не підтримують функції пов'язані з фінансовими операціями, а це є основним для отримання прибутку в приватних ІТ-школах. Крім цього, для швидкого керування навчальним процесом, слід зробити максимально зручний інтерфейс, щоб при роботі з системою, користувач не витрачав час на пошук потрібної йому функції. Отже, деякі з

описаних вище систем не відповідають вимогам для відділу освіти ІТ-школи Robocode, в той час як інші системи потребують вдосконалення в інтерфейсі.

### **1.6. Обґрунтування доцільності проектування й розроблення інформаційної системи для підтримки навчального процесу ІТ-школи Robocode**

По перш, в пункті 1.3 було відзначено, що існуюча CRM система, яка використовується в школі, має деякі суттєві недоліки, зокрема, складний інтерфейс та обмежену автоматизацію певних процесів, таких як підрахунок кількості проведених занять викладачем. Це ускладнює облік та аналіз даних, що є важливим для ефективного управління навчальним процесом.

По-друге, аналіз діяльності відділу освіти показав наявність серйозних недоліків у нинішній організаційній структурі відділу освіти, а саме: для підвищення категорії викладача, та оцінюванні учнів, використовуються сторонні програмні засоби. Додавши цей функціонал в систему, та автоматизувавши ці процеси, а також створивши зручну базу даних можна значно полегшити роботу відділу, тим самим підвищити ефективність його діяльності.

По-третє, порівняльний аналіз існуючих систем показав, що жодна з них не відповідає повністю потребам відділу освіти ІТ-школи Robocode. Деякі з систем не підтримують функції, пов'язані з фінансовими операціями, що є важливим для приватної освітньої установи. Також необхідно покращити інтерфейс користувача для забезпечення зручності та ефективності роботи з системою.

Отже, на основі вищесказаних аргументів, розробка нової інформаційної системи для керуванням навчальним процесом ІТ-школи Robocode є повністю обґрунтованою та необхідною для покращення навчального процесу.

### **1.7. Концептуальна модель системи**

У стандартній методології проектування інформаційних систем, спочатку створюється модель AS-IS, а потім на її основі аналізуються можливості для поліпшення процесів, тобто розробляється модель TO-BE [10].

Лише після створення моделі TO-BE розпочинається розробка моделі даних, прототипу, та наступний етап – створення остаточного варіанту інформаційної системи.

Якщо ж автоматизація підприємства буде ґрунтуватися на моделі AS-IS, це може призвести до простої комп'ютеризації існуючих недоліків, замість повноцінної інформатизації на основі новітніх технологій. У такому випадку, витрати на обладнання, розробку програмного забезпечення та його супровід можуть бути значними.

Зміни, які слід зробити в наявні процеси, наочно відображає функціональна модель TO-BE, зокрема діаграма A0 та інші її рівні декомпозиції. Цей процес був детально описаний за допомогою моделі IDEF3, як показано в додатку Б, рисунок Б.1-5.

### **1.8. Розрахунок економічного ефекту від впровадження системи**

Інформаційна система для підтримки навчального процесу в ІТ-школі Robocode має ступінь новизни розроблюваних задач «В» – використання типових проектних рішень за умови їх змін.

Група складності алгоритму 3 – алгоритми, що реалізують стандартні методи рішень і не передбачають використання складних чисельних і логічних методів.

Узагальнені дані вхідної та вихідної інформації для системи «підтримки навчального процесу ІТ школи Robocode» за видами вхідної та вихідної інформації, наведено в таблиці 1.4.

Таблиця 1.4. Узагальнені дані для вхідної та вихідної інформації системи  
«підтримки навчального процесу ІТ школи Robocode»

Вид інформації	Позначення	К-сть наборів даних
Змінна інформація	ЗІ	m=11
Нормативно – довідкова інформація	НДІ	n=3
Банк(база) даних	БД	p=1
Обробка в режимі реального часу	РЧ	Так
Забезпечення телекомунікаційної обробки даних і управління віддаленими об'єктами	ТОУ	Ні

Тепер слід визначити витрати часу на розробку ескізного проекту. Результати наведені в таблиці 1.5.

Таблиця 1.5. Визначення витрат часу для системи «підтримки навчального процесу ІТ-школи Robocode»

Вид системи	Стадія розробки системи			
	Ескізний проект (ПД), T <sub>1</sub>		Технічне завдання, T <sub>2</sub>	
	В	Г	В	Г
Управління організацією праці і зарплатою, управління кадрами, норми і нормативи, управління охороною праці				

- кількість форм вхідної інформації  $V_1 = 7$ ;
- кількість форм вихідної інформації  $V_2 = 11$ ;

- базове значення витрат часу для стадії «Технічний проект»  $T_{Б3}=115$ ;
- базове значення витрат часу для стадії «Робочий проект»  $T_{Б4}=332$ ;
- базове значення витрат часу для стадії «Впровадження»  $T_{Б5}=115$ .

Перейдемо до розрахунку витрат часу для стадії «Технічний проект».

Перед цим варто знайти коефіцієнти  $k_0$  та  $k_{п}$ .

Щоб знайти  $k_{п}$ , варто визначити коефіцієнти  $k_1, k_2, k_3$ . Коефіцієнти  $k_1, k_2, k_3$  для стадії «Технічний проект» представлені в таблиці 1.6.

Таблиця 1.6. Коефіцієнти  $k_1, k_2, k_3$  для стадії «Технічний проект»

Вид використаної інформації		Ступінь новизни
Назва	Коефіцієнт	В
Змінна інформація (ЗІ)	$k_1$	1.0
Нормативно-довідкова інформація(НДІ)	$k_2$	0.72
Банк (база) даних (БД)	$k_3$	2.08

За формулою 1.1 обрахуємо  $k_{п}$  – коефіцієнт трудомісткості робіт на стадії «Технічний проект».

$$k_{п} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p} = \frac{1 * 11 + 0.72 * 3 + 2.08 * 1}{11 + 3 + 1} = 1.016 \quad (1.1)$$

Наступним потрібно знайти коефіцієнт  $k_0$  – коефіцієнт ступеню новизни проекту. Коефіцієнти ступеня новизни наведені в таблиці 1.7.

Таблиця 1.7. Коефіцієнти ступеня новизни системи

Стадія розробки системи	Вид обробки	Ступінь новизни
		В
Технічний проект	РЧ	1.26
Робочий проект	РЧ	1.32
Впровадження	РЧ	1.21

Для стадії технічний проект  $k_0 = 1.26$ . За формулою 1.2 знайдемо  $T_3$  – витрати часу на розробку «Технічного проекту».

$$T_3 = T_{Б3} * k_{п} * k_0 = 115 * 1.016 * 1.26 = 147.21 \quad (1.2)$$

Розрахуємо витрати часу для стадії «Робочий проект». Аналогічно розрахунку часу для технічного проекту, слід знайти коефіцієнти  $k_0$  та  $k_{\Pi}$ , а також  $k_C$ .

Коефіцієнти для пошуку  $k_{\Pi}$ , а саме  $k_1$ ,  $k_2$ ,  $k_3$  для стадії «Робочий проект» представлені в таблиці 1.8.

Таблиця 1.8. Коефіцієнти  $k_1$ ,  $k_2$ ,  $k_3$  для стадії «Робочий проект»

Вид використаної інформації	Ступінь новизни
	В
$k_1$ (ЗІ)	1.0
$k_2$ (НДІ)	0.48
$k_3$ (БД)	0.40

Обрахуємо  $k_{\Pi}$  за формулою 1.1.

$$k_{\Pi} = \frac{1 * 11 + 0.48 * 3 + 0.40 * 1}{11 + 3 + 1} = 0.856$$

За таблицею 1.7 –  $k_0$  для робочого проекту буде дорівнювати 1.32. Щоб знайти  $k_C$  – коефіцієнт складності контролю вхідної та вихідної інформації, необхідно його ідентифікувати. Отримуємо  $k_C = 1.16$ . Тепер за формулою 1.3 обрахуємо  $T_4$  – витрати часу на розробку «Робочого проекту».

$$T_i = T_{\text{БІ}} * k_{\Pi} * k_0 * k_C = 332 * 0.856 * 1.32 * 1.16 = 435.15 \quad (1.3)$$

Розрахуємо витрати часу для стадії «Впровадження». Маємо такі коефіцієнти:  $k_{\Pi} = 0.856$ ,  $k_0 = 1.21$ ,  $k_C = 1.16$ . Аналогічно, за допомоги формули 1.3, обчислимо  $T_{\text{Б5}}$  – витрати часу на розробку стадії «Впровадження».

$$T_5 = 115 * 0.856 * 1.21 * 1.16 = 138.17$$

Отримавши  $T_3$ ,  $T_4$ ,  $T_5$  – розрахуємо загальні витрати часу на розробку системи по формулою 1.4.

$$T_{\Sigma} = T_1 + T_2 + T_3 + T_4 + T_5 = 817.53 \quad (1.4)$$

На розробку дипломного проекту, що передбачає 530 робочих годин при 8-годинному робочому дні, виділено  $\Phi$  днів.

$$\Phi = 530 / 8 = 66$$

Визначимо кількість місяців із розрахунку 22 робочих днів.

$$M = 66 / 22 = 3$$

Отже, для виконання такого проекту потрібно визначити чисельність виконавців, за формулою 1.5.

$$Ч = \frac{T_{\Sigma}}{\Phi} = 817.53 / 66 = 12 \quad (1.5)$$

Якщо прийняти, що програміст отримує оплату у розмірі ЗП<sub>ПР</sub> = 15000 гривень, то загальна виплата всім виконавцям складе (формула 1.6):

$$V'_1 = Ч * M * ЗП_{ПР} = 12 * 3 * 15000 = 540000 \text{ грн.} \quad (1.6)$$

Тепер обрахуємо витрати пов'язані з розробкою програми на ПК. Дійсний річний фонд часу ПК у годинах дорівнює числу робочих годин у році для оператора, за винятком часу на технічне обслуговування і ремонт ПК (в середньому 5 год/міс + 6 роб. днів/рік) і розраховується за формулою 1.7.

$$T_{ПК} = T_{ОП} - (6 * 8 + 5 - 12) = 000 - (6 * 8 + 5 - 12) = 1892 \text{ год.} \quad (1.7)$$

Оскільки під час виконання дипломного проекту здобувач в середньому витрачає R = 460 годин машинного часу, то величина фонду часу використання ПК для виконання кваліфікаційної роботи розраховується за формулою 1.8.

$$T'_{ПК} = T_{ПК} * \frac{R}{T_{ОП}} = 1892 * \frac{460}{2000} = 435.16 \text{ год.} \quad (1.8)$$

Наступним кроком, буде розрахунок поточних витрат на експлуатацію  $V''_1$ .

Балансована вартість ПК, де Ц<sub>р</sub> – ринкова вартість ПК, орієнтовно складає 14000 грн, К<sub>ун</sub> – коефіцієнт, що враховує витрати на установку ПК, К<sub>ун</sub>=0,12. Тоді за формулою 1.9 отримаємо:

$$Ц_{ПК} = Ц_r * (1 + K_{ун}) = 14000 * (1 + 0.12) = 15680 \text{ грн.} \quad (1.9)$$

Амортизаційні відрахування використання ПК – З<sub>ам</sub>, обчислюються за формулою 1.10. Взявши, що норма амортизаційних відрахувань для ПК Н<sub>а</sub> = 5.

$$З_{ам} = \frac{Ц_{ПК}}{Н_a} = \frac{15680}{5} = 3136 \text{ грн.} \quad (1.10)$$

Витрати на електроенергію З<sub>ел</sub>, споживану ПК, обчислюються за формулою 1.11.

$$З_{ел} = P_{ПК} * T'_{ПК} * Ц_{ел} * A = 0.4 * 435.16 * 1.98 * 0.9 = 310.18 \text{ грн.} \quad (1.11)$$

де потужність ПК,  $P_{ПК} = 0.4$  кВт, фонд корисного часу роботи ПК,  $T'_{ПК} = 435.16$  год, вартість 1 кВт електроенергії для підприємств,  $C_{ЕЛ} = 1.98$  грн/кВт, коефіцієнт інтенсивного використання ПК,  $A = 0.9$ .

Витрати на поточний ремонт і технічне обслуговування ПК  $Z_p$  визначаються як 6% від балансової вартості ПК,  $C_{ПК}$  (формула 1.12).

$$Z_p = C_{ПК} * 0.06 = 15680 * 0.06 = 940.8 \text{ грн.} \quad (1.12)$$

Непрямі витрати, пов'язані з експлуатацією ПК, визначаються як 5% від балансової вартості ПК  $C_{ПК}$  (формула 1.13).

$$Z_{МАТ} = C_{ПК} * 0.05 = 15680 * 0.05 = 784 \text{ грн.} \quad (1.13)$$

Таким чином, маємо: заробітна плата обслуговуючого персоналу (якщо роботи виконуються не на власному ПК)

$$Z_{ОП} = 2000 \text{ грн.} \quad Z_{АМ} = 3136 \text{ грн.} \quad Z_{ЕЛ} = 310.18 \text{ грн.}$$

Поточні витрати на експлуатацію  $V_1''$  розраховуються за формулою 1.14.

$$V_1'' = Z_{ОП} + Z_{АМ} + Z_{ЕЛ} + Z_p + Z_{МАТ} = 7170.98 \text{ грн.} \quad (1.14)$$

Тепер слід розрахувати загальні витрати на розробку програмного забезпечення комп'ютерної системи (формула 1.15).

$$V_1 = V_1' + V_1'' = 540000 + 7170.98 = 547170.98 \text{ грн.} \quad (1.15)$$

Наступним кроком буде розрахунок витрат на придбання і установку ПК  $V_2$  (формула 1.16).

$$V_2 = C_{ПК} \quad (1.16)$$

Отже  $V_2 = 15680$  грн.

Витрати на підготовку приміщення  $V_3$  відсутні, оскільки приміщення вже наявне.

Витрати на навчання персоналу  $V_4$  становлять 2000 грн, приймаючи до уваги, що середня тривалість навчання персоналу складає 1 місяць.

Загальна вартість розробки і впровадження системи вираховується за формулою 1.17.

$$V_{\Sigma} = V_1 + V_2 + V_3 + V_4 = 564850.98 \quad (1.17)$$

Основні джерела прибутку від впровадження комп'ютерної системи і порядок його підрахунку наведені в таблиці 1.9.

*Таблиця 1.9. Джерела прибутку від впровадження*

<b>Джерела прибутку</b>	<b>Порядок підрахунку прибутку</b>	<b>Сума за рік</b>
Збільшення кількості студентів.	Впровадження ефективної системи може зробити процес навчання більш привабливим для студентів, що призведе до збільшення кількості учнів та, відповідно, збільшення доходу від оплати навчання.	65000 грн.
Ефективне управління ресурсами.	Система може допомогти в ефективному плануванні ресурсів, включаючи використання персоналу та обладнання. Це може призвести до зменшення витрат та оптимізації внутрішніх процесів.	37000 грн.
Підвищення репутації.	Успішне впровадження системи може позитивно позначитися на репутації школи серед учнів, батьків та інших зацікавлених сторін. Тим самим залучати до додаткових заходів (змагань, таборів).	20000 грн.
Ефективне використання ресурсів.	Система може допомогти школі в ефективному використанні доступних ресурсів, таких як приміщення, обладнання та викладачі. Це може призвести до зменшення витрат та оптимізації бізнес-процесів.	15000 грн.
<b>Разом</b>		<b>137000</b>

Оскільки норма амортизаційних втрат для комп'ютерних систем  $H_A = 5$ , то для обрахування річного економічного ефекту слід брати до розгляду величину  $V_p$ , яка обраховується за формулою 1.18.

$$V_p = \frac{V_\Sigma}{H_A} = \frac{564850.98}{5} = 112970.196 \text{ грн.} \quad (1.18)$$

Щорічний прибуток від впровадження системи ПР буде досягнуто завдяки автоматизації та підвищенню якості і швидкості, орієнтовно становитиме 90000 гривень на рік.

Коефіцієнт економічної ефективності розробки вираховується за формулою 1.19.

$$K_{\text{ЕФ}} = \frac{P_p}{V_p} = \frac{90000}{112970.196} = 0.80 \quad (1.20)$$

Тепер за допомоги формули 1.20 визначимо термін окупності системи.

$$T_{\text{ок}} = \frac{1}{K_{\text{ЕФ}}} = \frac{1}{0.80} = 1.25 \quad (1.20)$$

Різниця між тим як було та після впровадження системи:

$$(AS-IS - TO-BE) = 330000 - 260000 = 70000$$

$$137000 + 70000 = 207000$$

Таким чином, термін окупності інформаційної системи буде займати 1 рік 2 місяці. Після цього, завдяки покращенню роботи відділу, шляхом автоматизації деяких процесів, прибутки будуть зростати.

### 1.9. Висновок до першого розділу

Отже, розробка нової інформаційної системи для управління навчальним процесом є необхідною та обґрунтованою. Вона дозволить підвищити ефективність роботи відділу, автоматизувавши важливі процеси та спростивши користувацький інтерфейс. Прогнозується, що термін окупності нової системи складе 1 рік і 2 місяці, після чого покращення роботи відділу призведе до зростання прибутків.

## РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ

### 2.1. Загальні положення

2.1.1. Найменування системи: «Інформаційна система для підтримки навчального процесу ІТ-школи Robocode ФОП «ЮШКОВ Дмитро Сергійович»».

2.1.2. Результати робіт зі створення системи оформлюються відповідно до вимог ДСТУ на кожному етапі розробки. Порядок оформлення й передачі результатів у цьому випадку визначається вмістом та календарним планом виконання розробки.

2.1.3. За потребою на подальших етапах робіт зі створення системи, окремі положення можуть бути уточнені й розвинуті.

### 2.2. Призначення і цілі створення системи

2.2.1. Призначення системи.

Система призначена для забезпечення ефективного управління всіма аспектами навчання, включаючи реєстрацію учнів, планування та моніторинг уроків і курсів, ведення звітності та аналізу успішності учнів.

2.2.2. Цілі створення системи.

Основною метою створення системи є автоматизація деяких процесів, а саме: підрахунок заробітної плати для вчителів на основі кількості проведених занять, та його категорії, а також відслідковування кількості учнів в кожній групі для оптимального використання ресурсів школи. Крім цього, зробити максимально зручний інтерфейс, для швидкого управління даним.

### 2.3. Характеристика об'єкта автоматизації

2.3.1. Короткі відомості про об'єкт автоматизації.

Об'єктом автоматизації є діяльність ІТ-школи Robocode ФОП «ЮШКОВ Дмитро Сергійович». Базовий об'єкт впровадження – Robocode.

## 2.4. Вимоги до системи

2.4.1. Вимоги до системи в цілому.

2.4.1.1. Вимоги до структури і функціонування системи.

2.4.1.1.1. Система повинна мати клієнт-серверну архітектуру, що використовує єдину базу даних (надалі — БД).

Згідно з функціональною структурою ІТ-школою Robocode, система повинна бути пов'язана в мережі з автоматизованими робочими місцями:

- менеджера;
- викладача.

2.4.1.1.2. Діагностика функціонування системи в мережі ІТ-школи Robocode має на меті виявлення будь-яких відхилень від нормального процесу вирішення завдань, а також виявлення порушень у роботі комп'ютерно-технічних засобів та програмних помилок.

Це досягається шляхом надання користувачам відповідних діагностичних повідомлень. Взаємодія між підсистемами здійснюється на рівні інформації за допомогою загальної бази даних та технічних засобів – локальних комп'ютерних мереж.

2.4.1.1.3. Розвиток і модернізація системи відбуваються через уточнення, розширення або заміну функцій, оновлення технічних і програмних засобів під час розробки та впровадження нових поколінь комп'ютерів. Структура та технологія програмного забезпечення мають забезпечувати простоту модернізації та розвитку, з можливістю розширення функціоналу та масштабування обсягів інформації, а також можливість реалізації на нових персональних комп'ютерах.

Програмно-технічні засоби мають включати програми з економіко-математичними та статистичними методами, методами моделювання, а також засоби табличного, текстового та графічного відображення даних. Сумісність програм та інформаційна взаємодія забезпечуються за допомогою загальносистемного протоколу обміну, використанням проблемно-орієнтованих

пакетів прикладних програм для між машинних зв'язків та єдиною системою класифікації та кодування.

2.4.1.1.4. Система повинна забезпечувати обробку даних як у діалоговому, так і у мережному (розподіленому) режимах.

2.4.1.2. Вимоги до чисельності і кваліфікації персоналу.

2.4.1.2.1. Персонал, який використовує автоматизовану систему, повинен дотримуватися наступних вимог:

- пройти навчання і отримати навички роботи на ПК;
- дотримуватись технологічних інструкцій під час використання системи у діалоговому режимі;
- виконувати вимоги щодо експлуатації персонального комп'ютера відповідно до інструкцій;
- зберігати інформацію та створювати резервні копії бази даних згідно з встановленими правилами;
- дотримуватись правил техніки безпеки під час роботи на персональному комп'ютері.

2.4.1.2.2. Користувачами системи може виступати менеджер, вчитель, учень. Вхід у систему повинен здійснюватися через пароль, який відображає рівень користувача: із правом коригування БД і без права. Залежно від рівня, користувач отримує повний чи обмежений доступ до системи.

2.4.1.3. Показники призначення.

2.4.1.3.1. Згідно з пунктом 2.2.1, показники призначення мають відображати рівень і якість автоматизації планування, обліку і управління в ІТ-школі Robocode для її оптимального функціонування. Перелік і допустимі значення цих показників, при яких система виконує свої основні завдання, повинні бути визначені на етапі технічного проектування.

2.4.1.3.2. При модернізації та розвитку системи необхідно забезпечити можливість налаштування її параметрів відповідно до характеристик об'єкта управління та периферійного обладнання, а також зміни процесів і методів організаційного управління.

#### 2.4.1.4. Вимоги до надійності.

2.4.1.4.1. Система є багатофункціональною та призначена для використання протягом робочого дня. Всі функції системи виконуються дискретно. Згідно з ДСТУ 2226-93, оцінка надійності проводиться для кожної функції окремо. Ураховуючи особливості функціонування системи, показники її надійності визначаються як показники надійності системи управління базами даних (СУБД), на якій вона базується, та технічних засобів, на яких вона працює. Основними показниками надійності є:

$L_i$  – ймовірність безвідмовного виконання задачі в зазначений термін (імовірність успішного виконання запиту).

$K_r$  – коефіцієнт готовності програмно-технічного комплексу (ПТК).

$T_v$  – середній час відновлення ПТК.

$T_e$  – мінімальний час між двома відмовами за календарний місяць.

#### 2.4.1.4.2. Комплекс технічних засобів повинен передбачати:

- можливість запуску та вирішення функціональних завдань з різних робочих станцій;

- можливість переходу до локального режиму роботи;

Для забезпечення надійності програмного та інформаційного забезпечення необхідно передбачити використання:

- модульного, структурного та об'єктно-орієнтованого програмування;

- програмних засобів контролю вхідної інформації з наданням користувачу повідомлень про виявлені помилки;

- програмних засобів корекції для виявлення та виправлення помилок у базі даних;

- засобів захисту від збоїв, несанкціонованого доступу, помилкових дій персоналу;

- резервних копій бази даних.

#### 2.4.1.5. Вимоги до безпеки.

Для забезпечення безпеки під час експлуатації, налаштування, монтажу, обслуговування та ремонту технічних засобів системи важливо дотримуватись вимог стандартів ДСТУ: ДСТУ 2293-99, ДСТУ ISO 6309:2007, ДСТУ 12.0.230:2008, ДСТУ 7237:2011, ДСТУ 7238:2011, ДСТУ 7239:2011. Також варто враховувати доступні рівні освітленості, вібраційні та шумові навантаження відповідно до вимог стандартів ДСТУ Б А.3.2-15:2011, ДСТУ EN 14253:2018, ДСТУ 2867-94.

#### 2.4.1.6. Вимоги з ергономіки та технічної естетики.

Загальні ергономічні та естетичні вимоги до системи повинні відповідати державним стандартам ДСТУ 8604:2015, ДСТУ 7298:2013. Освітленість робочого місця повинна відповідати вимогам стандартів ДСТУ EN 12464-1:2016 та ДБН В.2.5-28-2006.

Засоби відображення мають бути розташовані так, щоб кут спостереження екрану не перевищував 45 градусів, мінімальна відстань для спостереження екрану становить 0,3 м, рекомендована – 0,5 м.

При розробці програмного забезпечення необхідно створити зручний інтерфейс, щоб уникнути втомлюваність користувача.

#### 2.4.1.7. Вимоги по експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи.

##### 2.4.1.7.1. Види обслуговування системи визначаються згідно з ДСТУ EN 13306:2019. Загальні вимоги щодо експлуатації, технічного обслуговування та ремонту мають відповідати стандарту ДСТУ 3576-97.

##### 2.4.1.7.2. Для розміщення технічних засобів системи необхідно користуватися площами, визначеними відповідно до ДБН В.2.2-9-2009. При цьому слід дотримуватися вимог, викладених у експлуатаційній документації. Напруга живлення технічних засобів системи складає 220/380 В змінного струму з частотою $(50 \pm 1)$ Гц. Допустиме відхилення напруги в межах від +10% до -15%, а тривалість перерв у живленні не повинна перевищувати 0,001 секунди.

2.4.1.7.3. Чисельність, кваліфікація та графік роботи обслуговуючого персоналу мають відповідати рекомендаціям, визначеним у технічних умовах та інструкціях з експлуатації конкретних технічних засобів.

2.4.1.7.4. Склад, розташування та умови зберігання компонентів технічних засобів системи визначаються відповідно до рекомендацій, наведених у експлуатаційній документації для цих елементів.

2.4.1.7.5. План обслуговування має відповідати рівню та умовам роботи, щоб у разі відмови системи забезпечити її функціонування в аварійному режимі.

2.4.1.8. Вимоги до захисту інформації від несанкціонованого доступу.

2.4.1.8.1. Для надійності збереження і доступу до інформації необхідно використовувати засоби захисту:

1) серверних операційних систем Windows Server 2019 або 2020.

2) локальної мережі та програми захисту в мережі Firewall.

3) клієнт-серверної СУБД:

- тригери, представлення;
- процедури та функції;
- встановлення груп користувачів і ролей використання.

У кожному сеансі роботи системи необхідно спочатку ввести індивідуальний пароль. Система захисту паролів повинна мати вбудовані засоби для періодичної зміни паролів або використовувати стандартні інструменти середовища розробки. Для запобігання несанкціонованого доступу кожен співробітник має мати свій власний особистий пароль. Деякі таблиці також потрібно захистити від можливого редагування, додавання або видалення інформації.

2.4.1.9. Вимоги щодо збереження інформації при аваріях.

2.4.1.9.1. Необхідно забезпечити можливість резервного збереження бази даних у архіві після внесення корекцій, а також можливість відновлення бази даних із архіву в разі її пошкодження.

2.4.1.9.2. Резервний архів та база даних повинні бути збережені на різних носіях даних або пристроях.

#### 2.4.1.10. Вимоги по захисту від впливу зовнішніх діянь.

2.4.1.10.1. Електрична складова електромагнітного поля в приміщеннях не повинна перевищувати  $0,3 \text{ В/м}^2$  в діапазоні частот від 0,15 до 300 МГц. Для захисту від впливу електромагнітних полів та індустриальних завад рекомендується використовувати різноманітні екрани та фільтри.

2.4.1.10.2. Засоби, які мінімізують вплив шкідливих факторів на роботу комплексу технічних засобів, повинні бути спроектовані відповідно до вимог ДБН В.2.2-9-2009. Обчислювальні засоби, що мають бути стійкими до зовнішніх впливів, повинні відповідати вимогам, визначеним у ДСТУ 2506-94.

#### 2.4.1.11. Вимоги до патентної чистоти.

2.4.1.11.1. При створенні даної системи патентні дослідження не проводяться.

#### 2.4.1.12. Вимоги по стандартизації і уніфікації.

В системі кодування інформації необхідно дотримуватися світового класифікатора і стандартів.

#### 2.4.2. Вимоги до функцій.

2.4.2.1. Перелік функцій із зазначенням вхідної та вихідної інформації наведено в таблиці 2.1.

Функції повинні організувати робочий процес користувача ефективно, ґрунтуючись на безперервній технології. Це включає заповнення баз даних, роботу з довідниками, формування різноманітних звітів та виконання інших завдань, визначених відповідними документами. Пріоритетом є забезпечення зручного введення та використання інформації користувачем через створення підказок та меню на екрані монітора.

Таблиця 2.1. Перелік функцій, вхідної та вихідної інформації

№ п/п	Найменування функції	Вхідна інформація	Вихідна інформація
1	Формування та виведення детальної інформації про вчителів, з автоматичним розрахунком заробітної плати та кількості проведених занять	Таблиці БД «Вчитель» «Категорія викладача» «Заняття»	Форма зі зведеною інформацією про викладачів
2	Формування та виведення детальної інформації про групи, з автоматичним розрахунком кількості учнів	Таблиці БД «Група» «Слухач» «Курс» «Напрямок»	Форма зі зведеною інформацією про групи
3	Формування та виведення детальної інформації про слухачів, з автоматичним розрахунком середньої оцінки	Таблиці БД «Слухач» «Оцінка» «Група»	Форма зі зведеною інформацією про слухачів
4	Формування та виведення детальної інформації про курси, з автоматичним розрахунком суми їх продажу	Таблиці БД «Курс» «Напрямок» «Чек_з_оплатою»	Форма зі зведеною інформацією про курси
5	Формування та виведення детальної інформації про заняття	Таблиці БД «Заняття» «Тема курсу» «Вчитель»	Форма зі зведеною інформацією про заняття

## Продовження таблиці 2.1

		«Група»	
6	Пошук серед слухачів	Запит користувача, «Слухач» «Оцінка» «Група»	Форма з результатом пошуку
7	Пошук розкладу занять серед груп	Запит користувача, «Розклад занять» «Вчитель» «Група»	Форма з результатом пошуку
9	Пошук серед вчителів	Запит користувача, «Вчитель»	Форма з результатом пошуку

## 2.4.3. Вимоги до видів забезпечення.

2.4.3.1. У вимогах до математичного забезпечення (МЗ) системи, не передбачено потребу у спеціалізованому математичному програмному забезпеченні для виконання її функцій. Можливості обраної СУБД вважаються достатніми.

## 2.4.3.2. Вимоги до інформаційного забезпечення (ІЗ).

2.4.3.2.1. Інформаційне забезпечення системи має включати дані, необхідні для виконання всіх функцій, які на неї покладені. Це забезпечує раціональну організацію зберігання та доступу до інформації.

Заповнення БД інформацією здійснюється замовником з використанням методик та форм, створених розробниками системи.

Склад, структура та організація інформації представлені у логічній моделі БД можуть бути уточнені на етапі технічного проектування.

2.4.3.2.2. Необхідно передбачити захист даних від втрати у випадку аварій та порушень у енергопостачанні системи, за допомогою використання резервних копій БД.

## 2.4.3.3. Вимоги до лінгвістичного забезпечення (ЛЗ).

2.4.3.3.1. Для розробки програмних засобів, які виконують функції та забезпечують обслуговування користувачів, рекомендується використовувати мови високого рівня, що дозволяють створювати структурні програми. Також слід використовувати мову, яка використовується обраною СУБД для доступу та маніпулювання інформацією.

2.4.3.3.2. Взаємодія користувача з системою повинна базуватися на наборах меню та підказок, спрямованих на виконання функцій користувачем. Запити користувача до системи мають бути в основному сформульовані природною мовою.

2.4.3.4. Вимоги до програмного забезпечення (ПЗ).

2.4.3.4.1. Системне ПЗ повинно забезпечувати надійне та якісне виконання функціональних завдань системи. До системного ПЗ відноситься:

- операційна система (далі ОС) — Windows 10, 11;
- система управління БД (далі СУБД) — MySQL .

2.4.3.4.2. Загальні вимоги до системного ПЗ можна сформулювати так:

- мінімальні вимоги до ресурсів технічних засобів (ТЗ);
- максимальна швидкодія;
- повне задоволення потреб функціональних завдань системи.

2.4.3.4.3. Вимоги до ОС:

- мінімальне використання ресурсів комп'ютера для власних потреб, передусім оперативної і дискової пам'яті;
- максимальна швидкодія при управлінні зовнішніми пристроями;
- ОС сервера – Windows Server 2019 або 2020, ОС клієнта – Windows 10, або 11.

2.4.3.4.4. Вимоги до СУБД:

- максимальне задоволення потреб функціональних задач;
- надійність;
- ефективне управління потрібного обсягу і структури;
- швидкість виконання запитів користувачів;
- мінімальні вимоги до ТЗ.

2.4.3.4.5. Програмні засоби введення та виведення даних і ведення діалогу повинні забезпечувати:

- відображення необхідних даних на екрані у вигляді відповідних візуальних елементів;
- супровід введення даних контролем і повідомленням користувачу про наявність помилок з можливістю їх виправлення під час введення;
- інтерактивний діалог при введенні даних з використанням комп'ютерного керування;
- виведення даних у відповідному форматі (наприклад, у вигляді документа) за запитом користувача.

2.4.3.4.6. При розробленні спеціального ПЗ слід виконати наступні вимоги:

- використані програми повинні бути взаємно сумісні із собою, та загальносистемним ПЗ;
- ПЗ має розроблятися з використанням засобів об'єктно-орієнтованого програмування;
- забезпечити відповідність інтерфейсу користувача стандартам Windows;
- необхідно мати модульну структуру програми;
- повинна бути можливість розширення функціоналу відповідно до нових потреб;
- ПЗ не повинно залежати від типу зовнішніх пристроїв (принтерів, дисків, сканерів тощо);
- діалог з користувачем має відбуватися за допомогою клавіатури або миші з поясненням дій та можливістю отримання підказок.

2.4.3.5. Вимоги до технічного забезпечення.

2.4.3.5.1. Технічні засоби системи (таблиця 2.2) повинні забезпечувати виконання функцій, перерахованих в таблиці 2.1.

2.4.3.5.2. Обчислювальна техніка повинна забезпечувати можливість обміну інформацією у визначених обсягах, як це вказано в пункті 2.4.3.2.

*Таблиця 2.2. Вимоги до технічного забезпечення системи*

№ п/п	Основні характеристики комп'ютера
<b>Технічне забезпечення для сервера</b>	
1	HP ML115 Intel Xeon Quad Core 2,5 GHz\8 Gb\1 TB RAID5\ LAN 1 Gbit
<b>Технічне забезпечення для клієнта</b>	
1	Athlon QL-65 Dual Core 2,1 GHz; RAM: 2048 Mb; HDD: 250 Gb
2	Монітор 15"
3	Миша USB
4	Клавіатура USB

2.4.3.6. Вимоги до метрологічного забезпечення.

Система не включає в себе вимірювальні канали, вимірювальне обладнання чи прилади, тому вимоги до цього типу забезпечення не ставляться.

2.4.3.7. Вимоги до організаційного забезпечення.

2.4.3.7.1. Організаційне забезпечення системи розробляється в відповідності з вимогами державного стандарту по АСУП.

2.4.3.7.2. При впровадженні системи не передбачається зростання кількості співробітників на підприємстві. Розташування робочих місць, де буде встановлена система, визначається підприємством.

2.4.3.7.3. До функціонування системи висувуються наступні вимоги:

- список співробітників, які мають доступ до системи, визначається за наказом директора;
- контроль та ухвалення рішень під час аварійних ситуацій при експлуатації системи здійснює відповідальний за неї персонал.

## **2.5. Склад і зміст робіт по створенню системи**

2.5.1. Стадії створення системи і терміни виконання робіт наведені в таблиці 2.3.

Таблиця 2.3. Найменування робіт при створенні системи

№ п/п	Найменування робіт	Строки виконання робіт
1	Передпроектне дослідження об'єкта автоматизації	29.01.2024
2	Технічне завдання	10.02.2024
3	Технічний проект	28.03.2024
4	Оформлення документації	19.04.2024

## 2.6. Порядок контролю і приймання системи

2.6.1. Система вводиться на діючому ФОП Юшков Дмитро Сергійович ІТ-школи Robocode. При введенні в дію система повинна пройти приймальні випробування згідно з ДСТУ 3974-2000.

2.6.2. Розробники разом із замовником проводять тести для визначення працездатності та вирішення питання про придатність системи для дослідної експлуатації. Програму випробувань складає розробник і підтверджує замовник.

2.6.3. Допуск в дослідну експлуатацію надається відповідно до технічного завдання та інструкцій користувача. Після проведення дослідної експлуатації, складається список виявлених недоліків та рекомендації з їх виправлення, із визначенням строків виконання.

2.6.4. Введення в дію системи оформлюється актом здачі-прийому.

## 2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію

Для введення в дію замовник виконує ряд робіт із підготовки об'єкта:

- виконує укомплектування технічних засобів;
- організовує навчання користувачів щодо роботи на ПК та ознайомлення з інструкцією експлуатації;
- здійснює дослідну експлуатацію та вводить систему в роботу.

## **2.8. Вимоги до документації**

2.8.1. На систему розробляється комплекс документації у складі: технічне завдання та технічний проект.

2.8.2. Документація на систему розробляється згідно з вимогами Державних стандартів серії 19 "Єдина система програмної документації" та серії 24 "Єдина система стандартів автоматизованих систем управління".

## **2.9. Джерела розробки**

2.9.1. При розробленні технічного завдання на систему використано наступні документи:

- ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;
- ДСТУ 3973–2000 Система розроблення та поставлення продукції на виробництво;
- ДСТУ Б В.2.5–82:2016 Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом.

## РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ

### 3.1. Інформаційне забезпечення системи

При виконанні етапу проектування бази даних, був використаний програмний засіб CA ERwin Data Modeler. За його допомогою визначено структуру БД, розроблено атрибутивну модель даних, та здійснено перехід до фізичної моделі, завдяки методології IDEF1X.

Логічний рівень моделі представляє дані так, як вони виглядають у реальному світі. Модель цього типу є універсальною і побудована на основі процесної моделі (див. додаток В, рисунок В.1).

Атрибутивна модель містить усі сутності, атрибути та зв'язки.

Сутності моделі представляють об'єкти (реальні чи абстрактні) у вигляді прямокутників. Кожна сутність має власний набір характеристик, тобто атрибутів, і є унікально ідентифікованою за допомогою первинних ключів (РК). Крім первинних ключів, сутності також мають альтернативні ключі (АК), які є унікальними, та інверсні входи (ІЕ).

Деякі сутності пов'язані між собою логічними зв'язками. На схемі показані неідентифіковані зв'язки «один до багатьох», які позначені пунктирною лінією. При встановленні такого зв'язку між сутностями, первинний ключ з батьківської сутності переноситься до неключових атрибутів дочірньої, і позначається (FK) [11, 12].

Після створення логічної моделі даних, відбувається перехід до фізичної, в якій міститься уся інформація про об'єкти в БД (див. додаток В, рисунок В.2). На фізичному рівні, такі поняття як сутність та атрибут, замінюються на таблиця і колонка відповідно. Також цей рівень підтримує створення правил валідації, тобто логічні обмеження на значення атрибутів.

Структура фізичної моделі даних та наявні правила валідації описані в таблиці 3.1.

Таблиця 3.1. Структура фізичної моделі та правила валідації

№	Таблиці	Колонки та їх типи даних	Правила валідації
1	Категорія викладача	<ul style="list-style-type: none"> <li>– ID категорії (PK): integer</li> <li>– Назва_категорії: varchar(20)</li> <li>– Ставка_за_урок: float</li> </ul>	<ul style="list-style-type: none"> <li>– ставка оплати за урок, в залежності від категорії може бути від 135 грн до 675 грн.</li> </ul>
2	Викладач	<ul style="list-style-type: none"> <li>– ID вчителя (PK): integer</li> <li>– Ім'я: varchar(20)</li> <li>– Прізвище: varchar(20)</li> <li>– Вік: integer</li> <li>– Дата народження: datetime</li> <li>– Телефон: varchar(15)</li> <li>– ID категорії (FK): integer</li> </ul>	<ul style="list-style-type: none"> <li>– вік викладача не може бути менше 18 або більше 100 років.</li> </ul>
3	Група	<ul style="list-style-type: none"> <li>– ID_групи (PK): integer</li> <li>– Назва_групи: varchar(20)</li> <li>– ID_вчител (FK): integer</li> <li>– ID_курсу (FK): integer</li> </ul>	
4	Розклад_занять	<ul style="list-style-type: none"> <li>– ID_розкладу (PK): integer</li> <li>– День_тижня: varchar(20)</li> <li>– Дата_заняття: datetime</li> <li>– Час_заняття: datetime</li> <li>– ID_групи (FK): integer</li> </ul>	<ul style="list-style-type: none"> <li>– день тижня може бути позначений від 1 до 7;</li> <li>– якщо дата в розкладі не ставиться, то автоматично встановлюється поточна.</li> </ul>
5	Слухач	<ul style="list-style-type: none"> <li>– ID_слухача (PK): integer</li> <li>– Ім_я: varchar(20)</li> <li>– Прізвище: varchar(20)</li> <li>– Вік: integer</li> <li>– Дата_народження: datetime</li> <li>– Телефон: varchar(20)</li> </ul>	<ul style="list-style-type: none"> <li>– по правилах школи, мінімальний вік слухача для запису на курс це 7 років, максимальний 17;</li> <li>– дата народження слухача не може бути</li> </ul>

## Продовження таблиці 1.3

		– ID_групи(FK): integer	менше 7 років і більше 17 років.
6	Оцінка	– ID_оцінки (PK): integer – Оцінка: integer – ID_слухача (FK): integer – ID_заняття (FK): integer	– оцінка не може бути менше 0 балів та більше 100.
7	Чек_з_оплатою	– ID_чека (PK): integer – Сума: float – Дата_оплати: datetime – ID_слухача (FK): integer – ID_курсу (FK): integer	– сума не може бути оплаченою менше 0 грн; – якщо дату оплати було не вказано, зо за замовчуванням буде встановлено поточну.
8	Заняття	– ID_заняття (PK): integer – Номер_заняття: integer – Дата_заняття: datetime – Тип_заняття: varchar(20) – ID_теми (FK): integer – ID_вчителя (FK): integer – ID_групи (FK): integer	– так як в курсі може бути різна кількість занять, вони не можуть бути менше 0 і більше 46; – якщо не буде встановлену дату заняття, то автоматично з'явиться поточна дата; – вибір типу навчання, офлайн або онлайн.
9	Тема_курсу	– ID_теми (PK): integer – Назва_теми: varchar(40) – Номер_теми: integer – ID_курсу(FK): integer	– так як в курсі може бути різна кількість тем, вони не можуть бути менше 0 і більше 46.
10	Курс	– ID_курсу (PK): integer – Назва_курсу: varchar(20) – Ціна_курсу: float – ID_напрямку (FK): integer	– встановлення ціни курсу, мінімальна вартість якого 1000 грн, максимальна 1500 грн.

## Продовження таблиці 1.3

11	Напрямок	– ID_напрямку (PK): integer – Назва_напрямку: varchar(40)	
----	----------	--	--

На основі трансформаційної моделі було створено базу даних у середовищі СУБД MS SQL Server 2019. Схему цієї бази даних можна знайти в додатку В, на рисунку В.3.

### 3.2. Алгоритмізація та реалізація комплексу задач автоматизації

Перш ніж розпочати вирішення завдань автоматизації та безпосередню приступити до розробки додатку, необхідно створити проект у середовищі Visual Studio 2019 та підключити його до створеної бази даних у MS SQL Server 2019. Спочатку, варто обрати джерело даних – БД (рисунок 3.1).

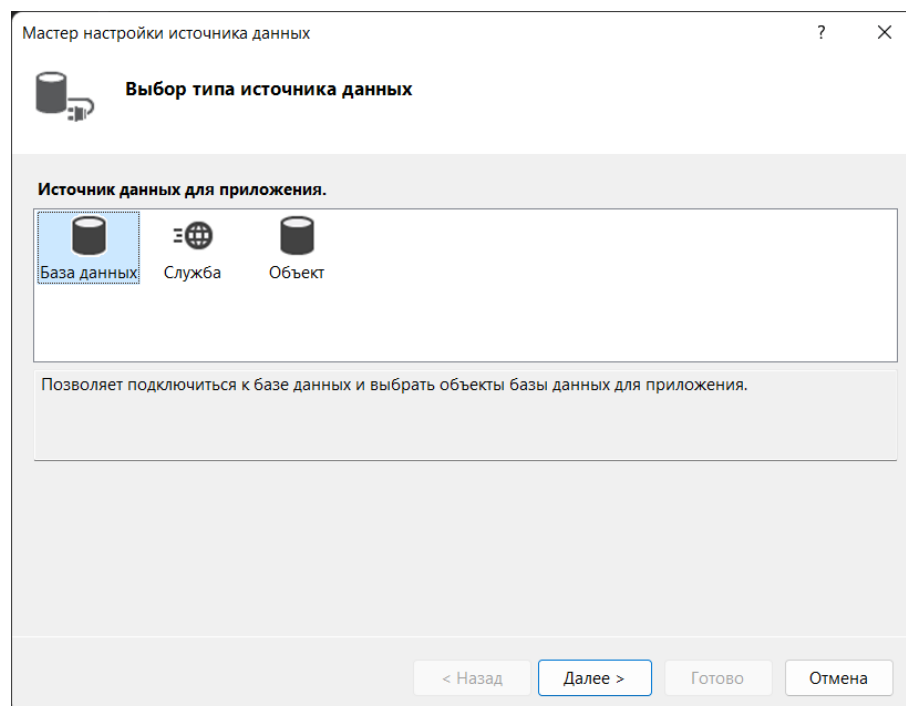


Рисунок 3.1 – Вибір типу джерела даних

На наступному кроці, обираємо модель бази даних (рисунок 3.2).

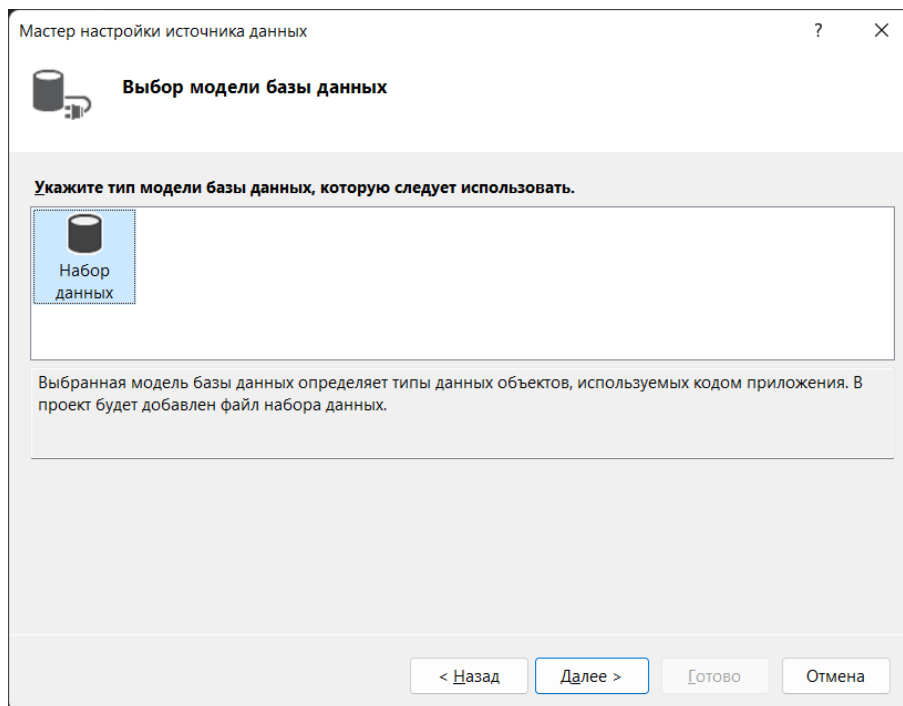


Рисунок 3.2 – Вибір моделі БД

Третім кроком є налаштування підключення. Для цього слід вказати ім'я серверу, яким чином буде відбуватися вхід на сервер, а також ім'я БД (рисунок 3.3).

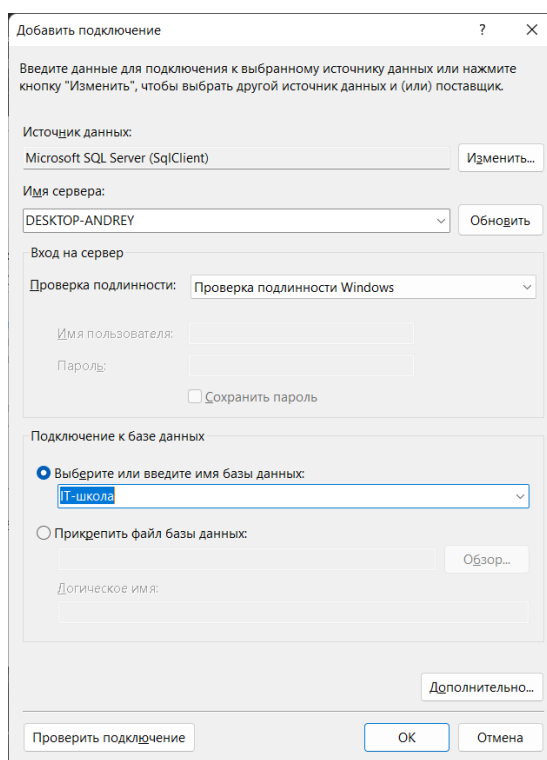


Рисунок 3.3 – Підключення БД

На останньому етапі, обираємо потрібні об'єкти БД, які буду завантажені в середовище розробки (рисунок 3.4).

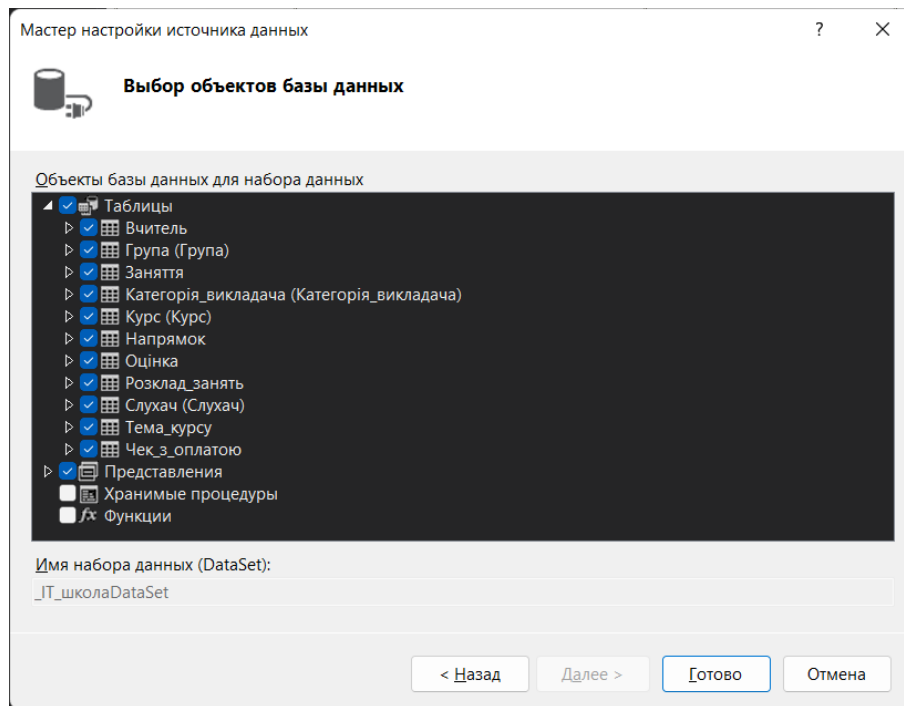


Рисунок 3.4 – Вибір об'єктів БД

Після того, як проект підключено до бази даних, можна приступати до вирішення завдань автоматизації та розробки функцій системи.

Першим завданням є забезпечення безпечного входу до системи. Для цього було створено форму авторизації користувача. На цій формі користувач повинен ввести свій логін та пароль, наданий адміністратором системи (див. додаток Г, рисунок Г.2).

У разі введення правильного паролю, перед користувачем відкривається головна форма. Якщо ж пароль введено неправильно, система видає відповідні повідомлення з вказівкою на помилку (див. додаток Г, рисунок Г.3).

Щоб забезпечити зручну навігацію по системі, було створено форму з меню, яке містить різні вкладки. Вони дозволяють переходити до виконання різних функцій: введення та редагування даних, пошуку даних, формування звітів (див. додаток Г, рисунок Г.4).

Для вирішенні задачі стосовно внесення усіх потрібних даних, а також їх корегування та видалення, були створені відповідні форми для таких таблиць (див. додаток Г , рисунок Г.5-11):

- категорія викладача та пов'язана з ним таблиця вчитель;
- слухач та пов'язана з ним таблиця оцінки;
- курс та пов'язана з ним таблиця тема курсу;
- напрямок;
- група та пов'язана з нею таблиця розклад занять;
- заняття;
- чек з оплатою.

Для кожної форми було застосовано функцію, яка виводить повідомлення про порушення правил валідації під час збереження даних в таблиці. Код опису функції наведено на рисунку 3.5 для таблиці «Викладач».

```
ссылка: 1
private void категория_викладачаBindingNavigator_Click_1(object sender, EventArgs e)
{
    this.Validate();
    this.категория_викладачаBindingSource.EndEdit();
    try
    {
        this.tableAdapterManager.UpdateAll(this._IT_школаDataSet);
    }
    catch (SqlException ex)
    {
        MessageBox.Show("Помилка: " + ex.Message, "Помилка");
    }
}
```

Рисунок 3.5 – Код перехоплення порушень правил валідації

Приклад повідомлення про помилку введення даних, що порушує правила валідації, наведено на рис. 3.7:

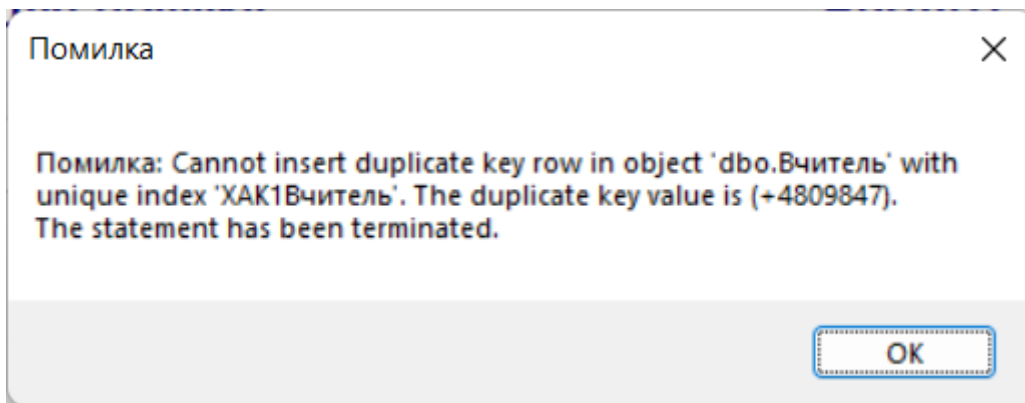


Рисунок 3.6 – Повідомлення про помилку введення даних

Для забезпечення зручності роботи з великою кількістю даних, на деяких формах було реалізовано пошук та фільтрацію (рисунок 3.7-3.8).

	Прізвище	Вік	Дата_народження	Телефон	ID_категорії
▶	Іванова	45	18.12.1977	+380987654324	S-Junior teach
	Григоренко	30	25.07.1992	+380987654325	S-Junior teach
*					

Рисунок 3.7 – Приклад процедури фільтрації даних на формі

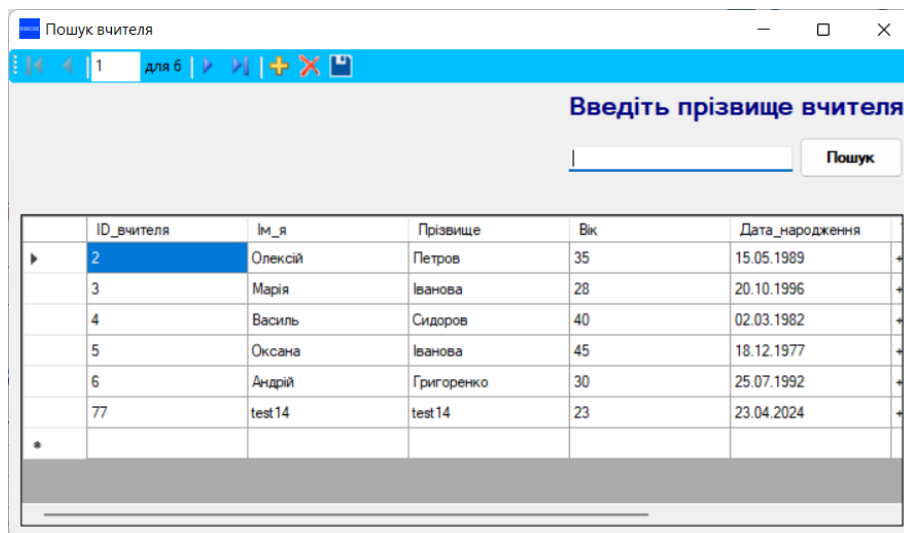


Рисунок 3.8 – Приклад процедури пошуку даних на формі  
Код для здійснення сортування даних, представлено на рисунку 3.9.

```

ссылка: 1
private void radioButton_filtOn_CheckedChanged(object sender, EventArgs e)
{
    this.категорія_викладачаBindingSource.Filter = $"Назва_категорії = '{textBox_filt_nameCategory.Text}'";
}

ссылка: 1
private void radioButton_filtOff_CheckedChanged(object sender, EventArgs e)
{
    this.категорія_викладачаBindingSource.RemoveFilter();
}

```

Рисунок 3.9 – Код виконання та припинення фільтрації

Код для здійснення пошуку даних, з переходом по записах, у разі знайденого дублікату представлено на рисунку 3.10.

```

private int lastFoundIndex = -1;
ссылка: 1
private void button_searchTeacher_Click(object sender, EventArgs e)
{
    string searchText = textBox_searchTeacher.Text.Trim();

    if (string.IsNullOrEmpty(searchText))
    {
        MessageBox.Show("Будь ласка, введіть прізвище для пошуку.");
        return;
    }

    int startIndex = lastFoundIndex + 1;

    if (startIndex >= вчительBindingSource.Count)
    {
        startIndex = 0;
        lastFoundIndex = -1;
    }

    int nextFoundIndex = -1;

    for (int i = startIndex; i < вчительBindingSource.Count; i++)
    {
        DataRowView rowView = (DataRowView)вчительBindingSource[i];
        string surname = rowView["Прізвище"].ToString().Trim();

        if (surname.IndexOf(searchText, StringComparison.OrdinalIgnoreCase) >= 0)
        {
            nextFoundIndex = i;
            break;
        }
    }

    if (nextFoundIndex != -1)
    {
        вчительBindingSource.Position = nextFoundIndex;
        lastFoundIndex = nextFoundIndex;
    }
    else
    {
        MessageBox.Show("Вчителя з введеним прізвищем не знайдено.");
        lastFoundIndex = -1;
    }
}

```

Рисунок 3.10 – Код виконання пошуку даних по записах

На деяких формах, присутні розрахункові поля. Наприклад, для таблиці «Вчитель» за допомогою SQL запиту розраховується кількість проведених занять, та його заробітна плата (рисунок 3.11).

Щоб реалізувати розрахункові поля, слід в схемі даних БД (рисунок 3.12) в TableAdapter додати інструкцію SELECT (рисунок 3.13).

**Данные для загрузки в таблицу**

```

SELECT Вчитель.Прізвище, Вчитель.Ім_я, Вчитель.Вік, Вчитель.Телефон, Категорія_викладача.Назва_категорії,
COUNT(Заняття.ID_заняття) AS Кількість_проведених_занять,
Категорія_викладача.Ставка_за_урок * COUNT(Заняття.ID_заняття) AS Зарплата
FROM Вчитель INNER JOIN
Категорія_викладача ON Вчитель.ID_категорії = Категорія_викладача.ID_категорії INNER JOIN
Заняття ON Вчитель.ID_вчителя = Заняття.ID_вчителя
GROUP BY Вчитель.ID_вчителя, Вчитель.Ім_я, Вчитель.Прізвище, Вчитель.Вік, Вчитель.Дата_народження,
Вчитель.Телефон, Вчитель.ID_категорії, Категорія_викладача.Ставка_за_урок,
Категорія_викладача.Назва_категорії

```

Конструктор запросов...

Рисунок 3.11 – Код SQL запиту для підрахунку кількості проведених занять, та заробітної плати викладача

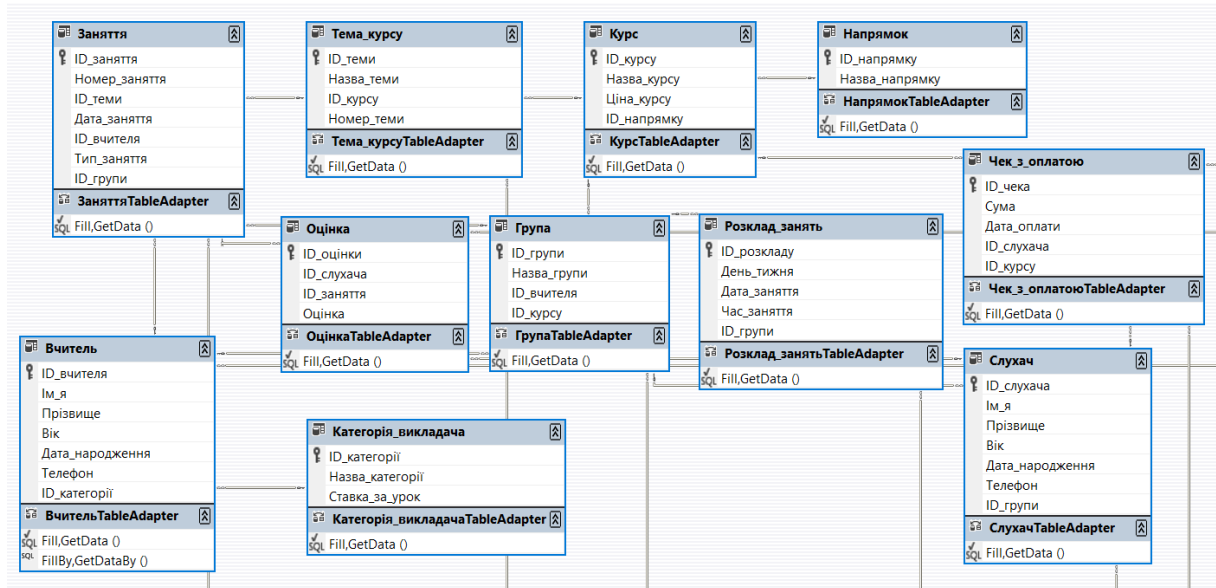


Рисунок 3.12 – Схема БД DataSet у вигляді XSD

Аналогічним чином, були розроблені розрахункові поля й для інших таблиць, додавши SQL код в інструкцію SELECT об'єкта TableAdapter.

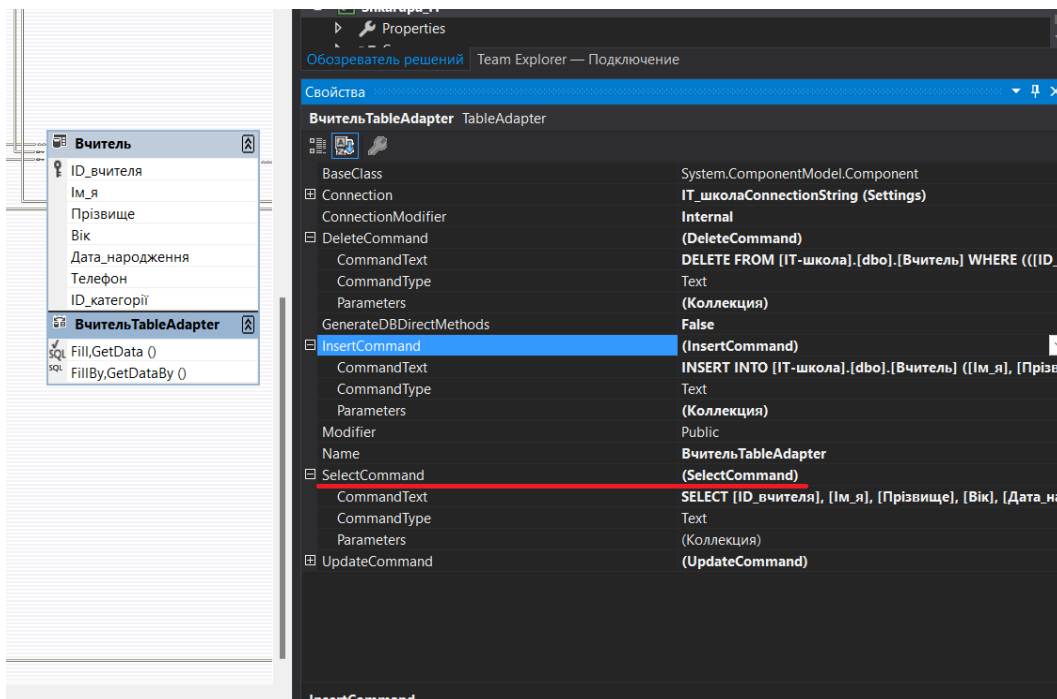


Рисунок 3.13 – Властивості ВчительTableAdapter

В таблиці «Група» реалізовано автоматичний розрахунок кількості дітей в групі (рисунок 3.14).

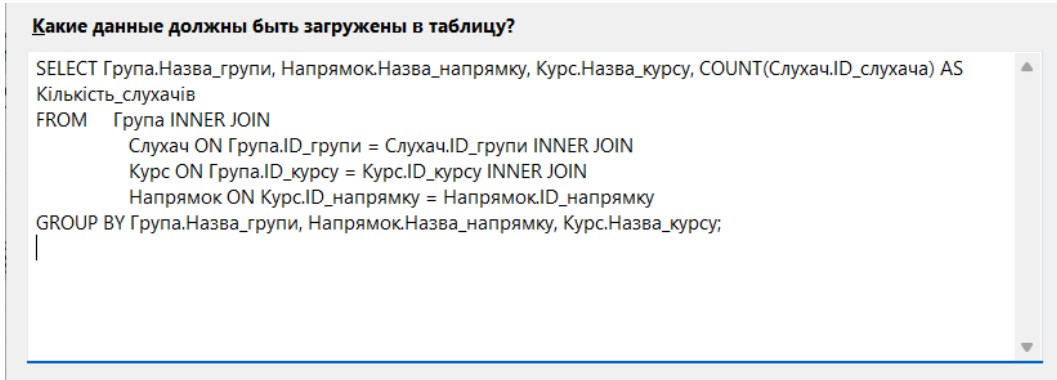


Рисунок 3.14 – Код SQL запиту для підрахунку кількості учнів в групі

В таблиці «Слухач» реалізовано автоматичний розрахунок середньої оцінки учня (рисунок 3.15).

```

Какие данные должны быть загружены в таблицу?
SELECT Слухач.Ім_я, Слухач.Прізвище, Слухач.Вік, Слухач.Дата_народження, Слухач.Телефон, AVG
(Оцінка.Оцінка) AS Середня_оцінка, Група.Назва_групи
FROM   Слухач INNER JOIN
        Оцінка ON Слухач.ID_слухача = Оцінка.ID_слухача INNER JOIN
        Група ON Слухач.ID_групи = Група.ID_групи
WHERE  (Слухач.Прізвище = @прізвище)
GROUP BY Слухач.Ім_я, Слухач.Прізвище, Слухач.Вік, Слухач.Дата_народження, Слухач.Телефон,
Група.Назва_групи

```

Рисунок 3.15 – Код SQL запиту для підрахунку середньої оцінки учня

В таблиці «Курс» реалізовано автоматичний розрахунок суми проданих курсів, за період навчання (рисунок 3.16).

```

Какие данные должны быть загружены в таблицу?
SELECT Курс.Назва_курсу, Курс.Ціна_курсу, Напрямок.Назва_напрямку, COALESCE (SUM
(Чек_з_оплатою.Сума), 0) AS Сума_продажів
FROM   Курс INNER JOIN
        Напрямок ON Курс.ID_напрямку = Напрямок.ID_напрямку LEFT OUTER JOIN
        Чек_з_оплатою ON Курс.ID_курсу = Чек_з_оплатою.ID_курсу
GROUP BY Курс.Назва_курсу, Курс.Ціна_курсу, Напрямок.Назва_напрямку;

```

Рисунок 3.16 – Код SQL запиту для підрахунку загальної суми продаж курсів

В системі також присутній розширений пошук, завдяки параметричного запиту. Реалізація параметричного запиту представлена на рисунках 3.17-3.21.

Для початку слід створити TableAdapter, в якому буде відбуватися налаштування параметричного запиту (рисунок 3.17).

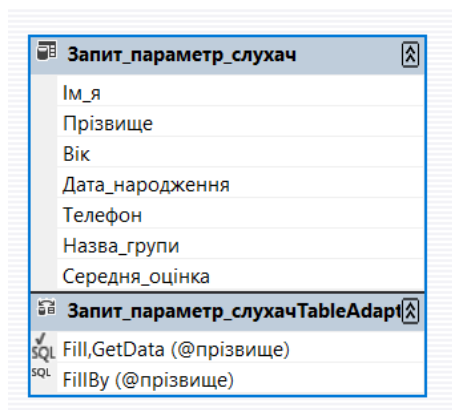


Рисунок 3.17 – Створений TableAdapter

З'явиться вікно «Майстер налаштування адаптеру таблиці». Обираємо відповідне підключення до бази даних та переходимо в «Конструктор запитів».

В конструкторі обираємо необхідні таблиці, визначаємо поля та задаємо параметр пошуку. На рисунку 3.18 зображено налаштування параметричного запиту.

Наступним кроком йде налаштування форми, де користувач зможе здійснювати пошук по прізвищу учня. Для цього на формі розміщується табличний вигляд створеного параметричного запиту та поля для вводу даних (рисунок 3.19).

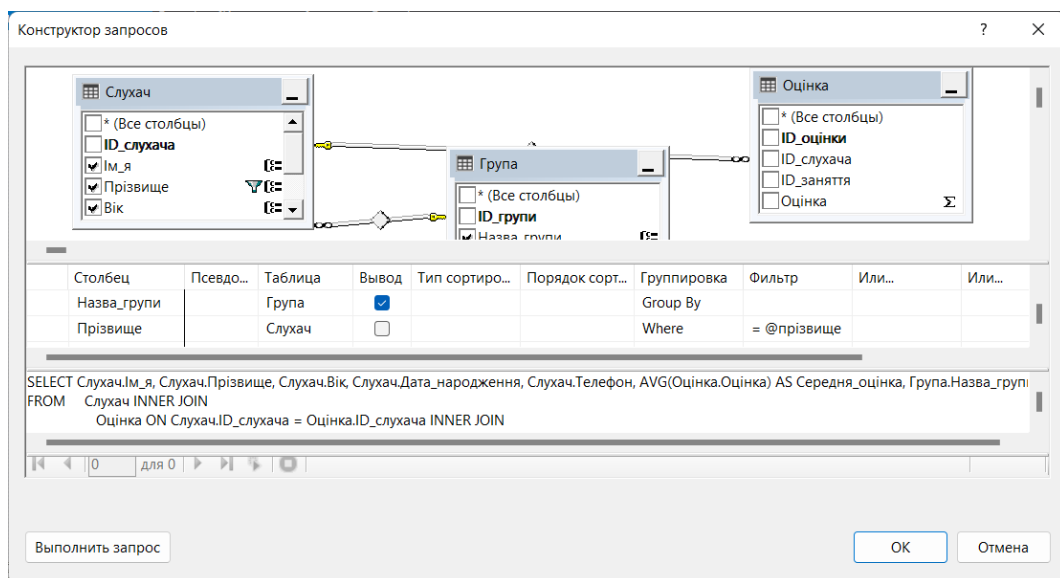


Рисунок 3.18 – Створення SQL-запиту з параметром для пошуку даних

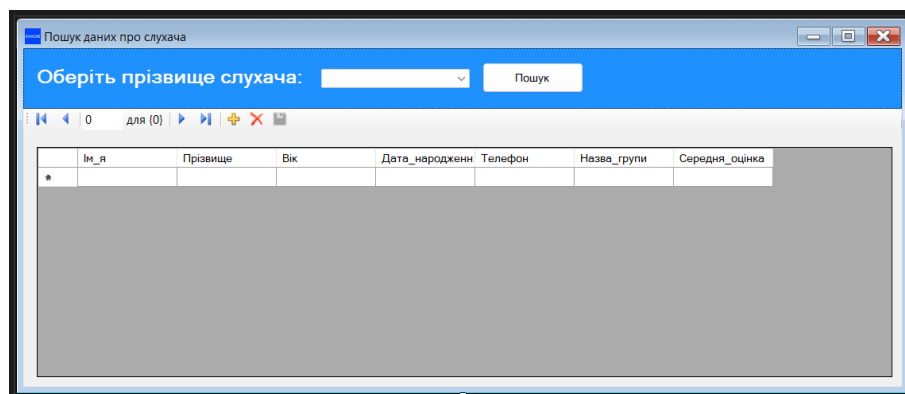


Рисунок 3.19 – Структура форми параметричного запиту

Тепер варто налаштувати детально combobox, для відображення списку прізвищ учнів із бази даних (рисунок 3.20).

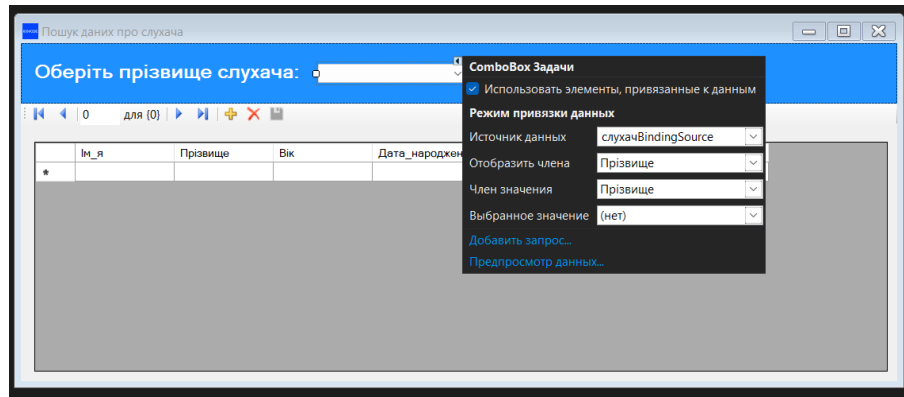


Рисунок 3.20 – Налаштування comboBox

Щоб пошук спрацював, на кнопку «Пошук» слід додати такий код (рисунок 3.21).

```

ссылка: 1
private void reqStudentButton_Click(object sender, EventArgs e)
{
    try
    {
        this.запит_параметр_слухачTableAdapter.Fill(this.IT_школаDataSet.запит_параметр_слухач, comboBox_reqStudent.SelectedValue.ToString());
    }
    catch (System.Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
}

```

Рисунок 3.21 – Код для кнопки пошук в параметричному запиту

Тож, в інформаційній системі присутні наступні пошуки:

- пошук даних по учню, за його прізвищем;
- пошук розкладу для групи по її назві;
- пошук даних про вчителя за прізвищем;
- пошук напрямку програмування, по назві напрямку;
- пошук інформації про групу, по назві групи;
- пошук інформації про заняття, по даті проведення;
- пошук інформації про оплату курсу, по даті сплати.

Користувач системи може переглянути деяку звітність. Функція формування звітів реалізована в пункті «Звіти». Йому достатньо обрати назву звіту, і він автоматично виведеться на екран.

Аби забезпечити можливість створення звітів, потрібно додати до проекту новий елемент — «Майстер звіту» (рисунок 3.22).

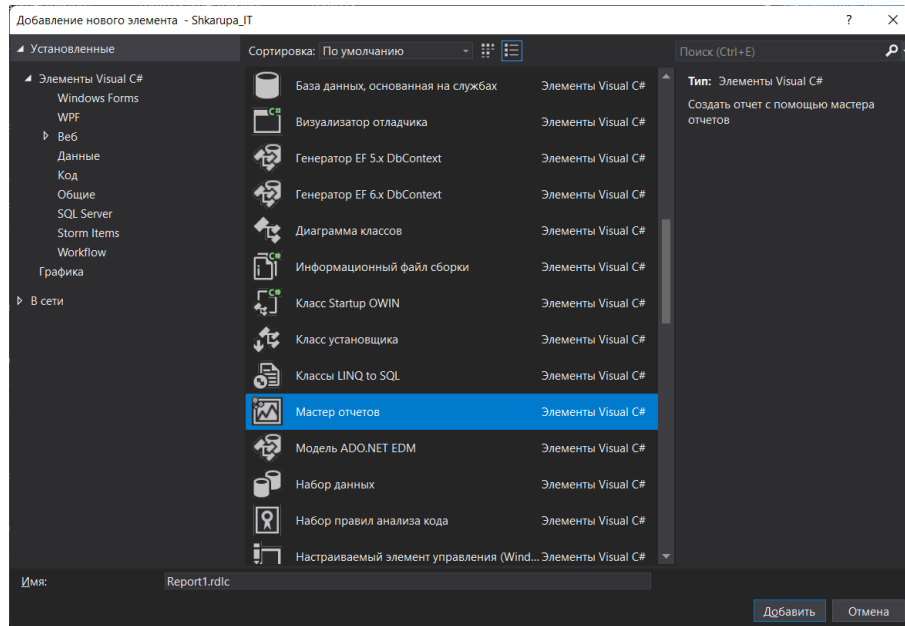


Рисунок 3.22 – Додавання нового елементу «Майстер звіту» до проекту

Наступним кроком є підв'язка «Майстра звіту» з відповідною таблицею або запитом (рисунок 3.23), а також налаштувати оформлення звіту (рисунок 3.24).

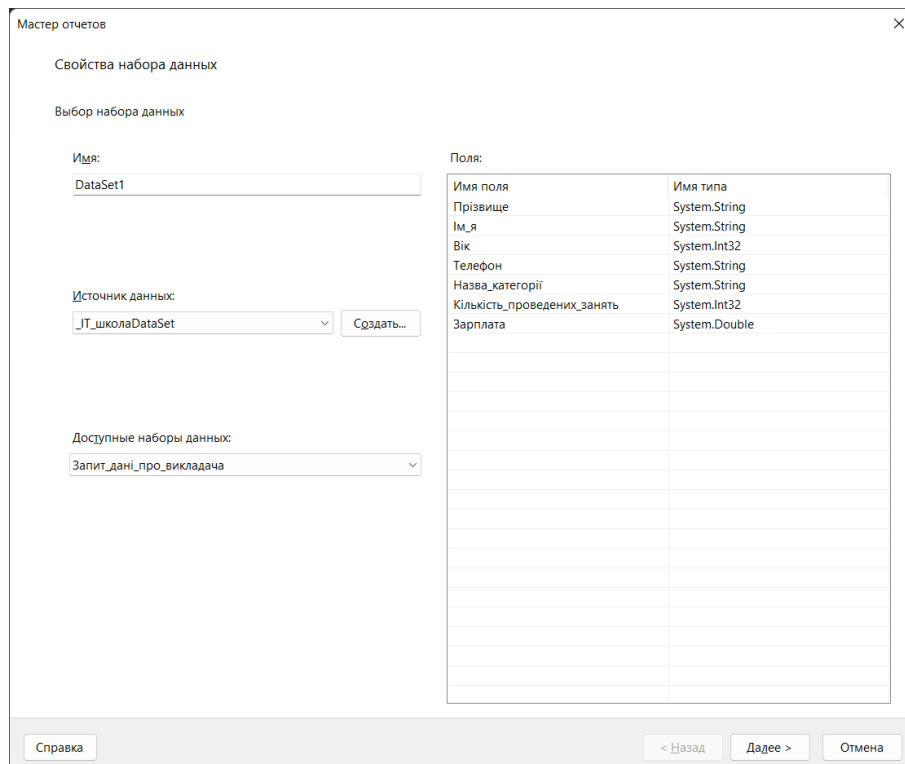


Рисунок 3.23 – Вибір набору даних та розміщення полів

Прізвище	Ім'я	Вік	Телефон	Назва категорії	Кількість проведених занять	Зарплата
[Прізвище]	[Ім_я]	[Вік]	[Телефон]	[Назва_категорії]	[Кількість_проведених_занять]	Sum(Зарплата)

Рисунок 3.24 – Налаштування вигляду звіту

Після налаштування, створюємо форму та додаємо на неї ReportViewer, зв'язуємо його з необхідним шаблоном звіту (рисунок 3.25).

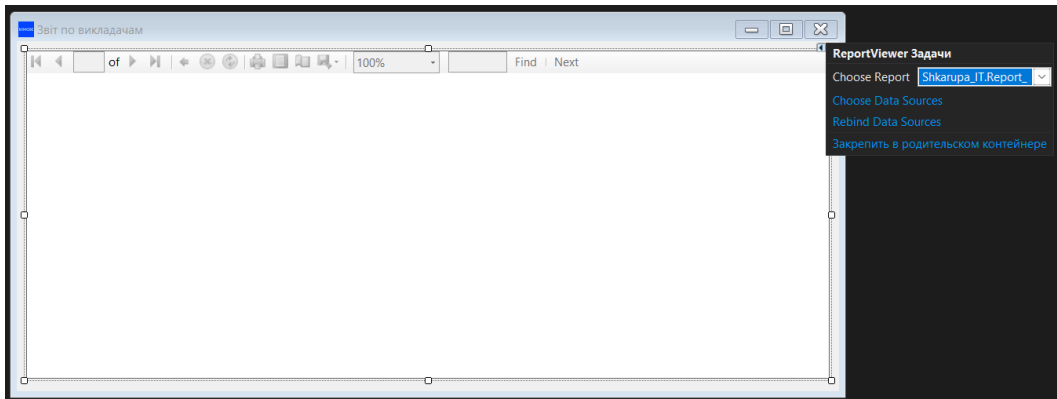


Рисунок 3.25 – Створення форми для відображення звіту

Система надає користувачу, створити такі звіти:

- звіт, щодо роботи викладачів;
- звіт, щодо успішності учнів;
- звіт, щодо інформації з продажу курсів;
- звіт, щодо інформації по групам;
- звіт по заняттях.

### 3.3. Інструкція користувача

Запустивши програму, з'являється загрузочний екран. На ньому міститься назва ІТ-школи, та інформація про розробника (рисунок 3.26).

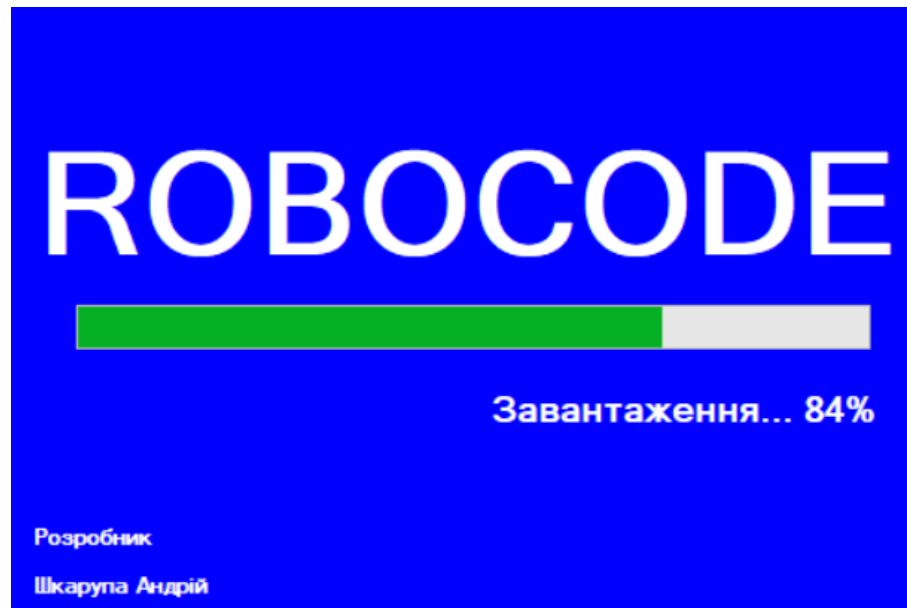


Рисунок 3.26 – Загрузочний екран

Після завершення завантаження користувач автоматично перенаправляється до вікна авторизації. У цьому вікні користувачеві пропонується ввести свої облікові дані – логін та пароль (рисунок 3.27).



Рисунок 3.27 – Вікно авторизації користувача

Після успішної авторизації користувачеві відкривається головне меню (рисунок 3.28), яке забезпечує зручну навігацію. Для доступу до повного списку вкладок і можливостей необхідно обрати відповідний пункт меню. Це дозволить виконати введення, перегляд або друк даних залежно від потреб користувача.

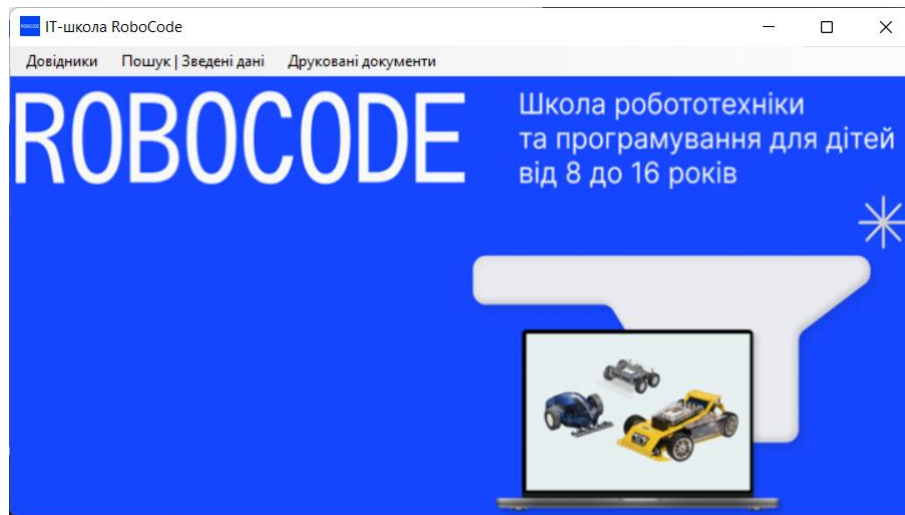


Рисунок 3.28 – Головне меню системи

При виборі пункту «Довідники» відкривається список можливих таблиць для введення даних в БД (рисунок 3.29). Натиснувши на будь-який пункт списку, буде відкрита відповідна форма.

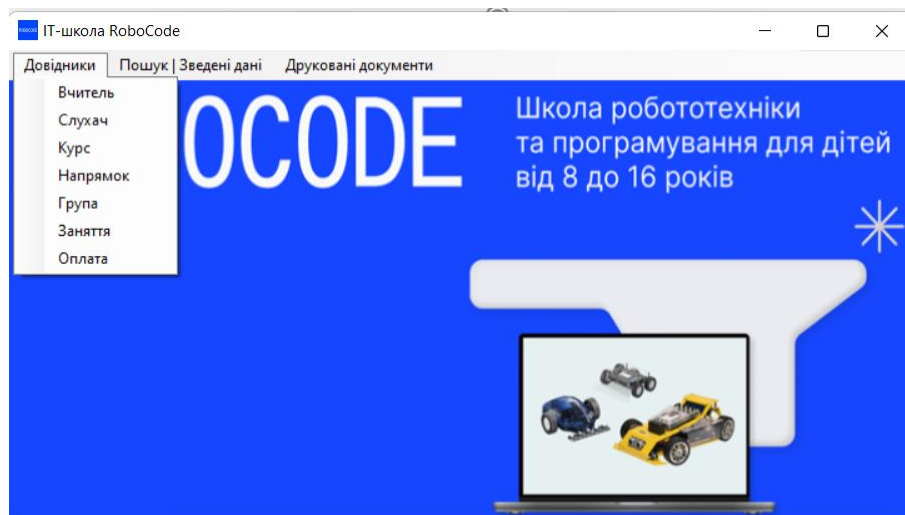


Рисунок 3.29 – Підпункт меню «Довідники»

Обравши пункт меню «Пошук | Зведені дані», відкриється список можливих запитів до БД (рисунок 3.30). За допомогою них, можна дізнатися детальну інформацію про вчителів, слухачів, заняттях, групах та курсах, а також здійснити пошук.

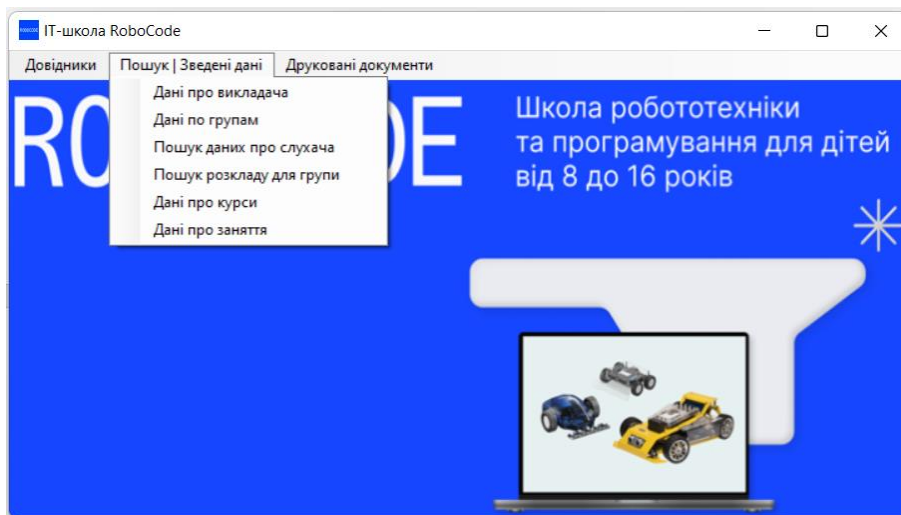


Рисунок 3.30 – Підпункт меню «Пошук | Зведені дані»

Останнім пунктом головного меню є вкладка «Друковані документи», натиснувши на неї, з'явиться список можливих звітів (рисунок 3.31). В цьому пункті можна переглянути роботу викладачів, успішність учнів, продаж курсів.

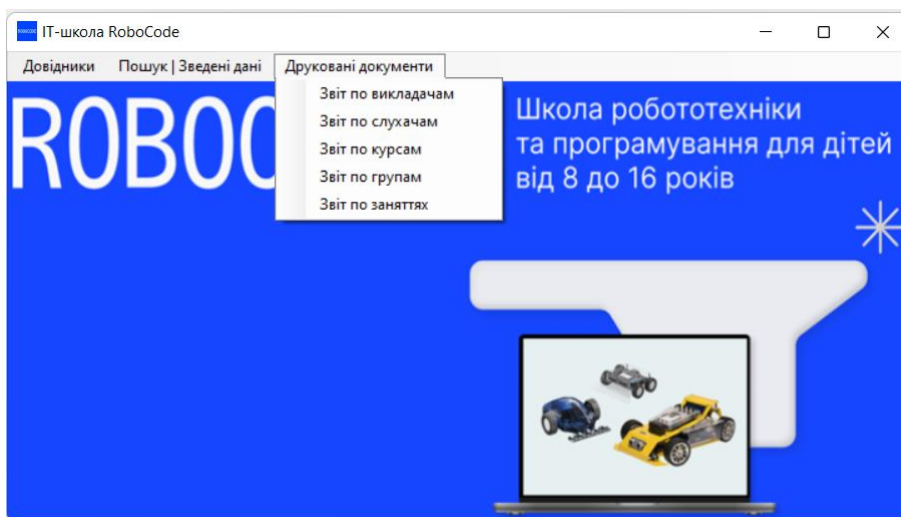


Рисунок 3.31 – Підпункт меню «Друковані документи»

В пункті «Довідники» з розкриваючого списку ватро обрати необхідний підпункт для введення даних в БД. Наприклад підпункт «Вчитель» (рисунок 3.32).

ID_вчителя	Ім'я	Прізвище	Вік	Дата_народження	Телефон
2	Олексій	Петров	35	15.05.1989	+380987
3	Марія	Іванова	28	20.10.1996	+380987

Рисунок 3.32 – Зв'язана форма для введення даних про вчителя та категорію викладача

Для навігації по даним, використовується компонент BindingNavigator (рисунок 3.33).

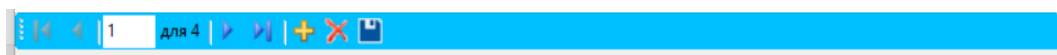


Рисунок 33 – Елемент BindingNavigator, для навігації по даним

Для перегляду даних можна скористатися кнопками у вигляді синіх стрілок, які дозволяють переміщатися між записами вперед і назад.

Щоб додати новий запис, потрібно натиснути на жовтий хрестик у верхньому меню вікна. В результаті з'явиться порожній рядок, в який користувач може ввести потрібні дані у відповідні поля поруч з таблицею.

Для збереження доданого запису, або змін при його видаленні, необхідно натиснути кнопку збереження у верхньому меню.

Якщо потрібно видалити певний запис, достатньо натиснути на червоний хрестик .

На формі «Вчитель» (рисунок 3.32) є кнопка «Відкрити». Натиснувши на неї, відкривається форма «Пошук вчителя». На цій формі слід ввести прізвище викладача, та натиснути кнопку «Пошук» (рисунок 3.34).

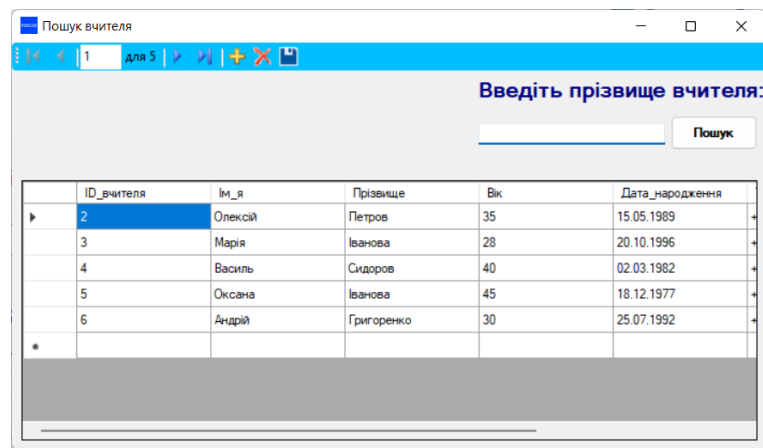


Рисунок 3.34 – Форма «Пошук вчителів» з результатом пошуку

Крім того, на формі «Вчитель» (рисунок 3.32) присутня фільтрація даних по категорії викладача. Щоб відфільтрувати дані, слід в поле ввести назву категорії, та обрати пункт «Відфільтрувати», в результаті чого, виведуться всі викладачі, що займають посаду, за якої була здійснена фільтрація (рисунок 3.35).

Для зняття фільтру є пункт «Зняти фільтр», який поверне таблицю в початковий стан.

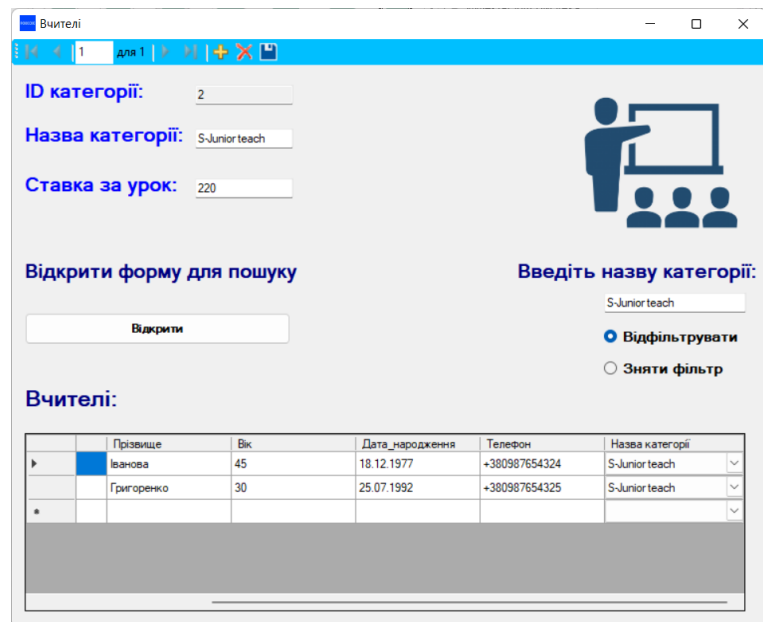


Рисунок 3.35 – Результат фільтрації даних по назві категорії викладача

Пошук даних реалізовано на таких формах:

- форма «Слухачі» – пошук по прізвищу;
- форма «Напрямки» – пошук по назві напрямку програмування;
- форма «Групи» – пошук по назві групи;

- форма «Заняття» – пошук по даті заняття;
- форма «Чеки з оплатою» – пошук по даті оплати.

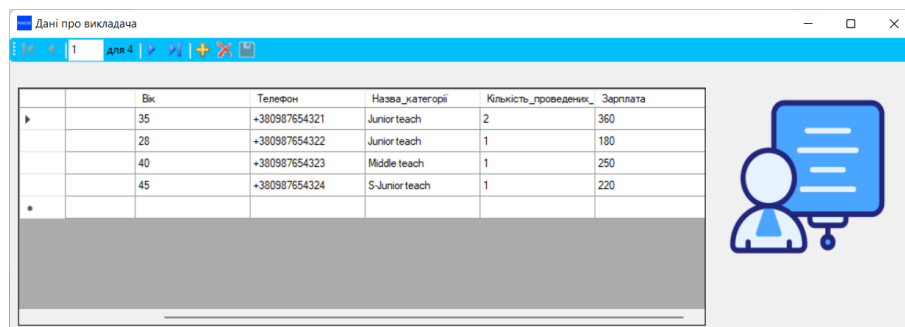
Фільтр даних, реалізовано на таких формах:

- форма «Курс» – фільтр по назві курсу.

Результати всіх пошуків та фільтрів представлено в додатку Г, рисунок Г.12-19.

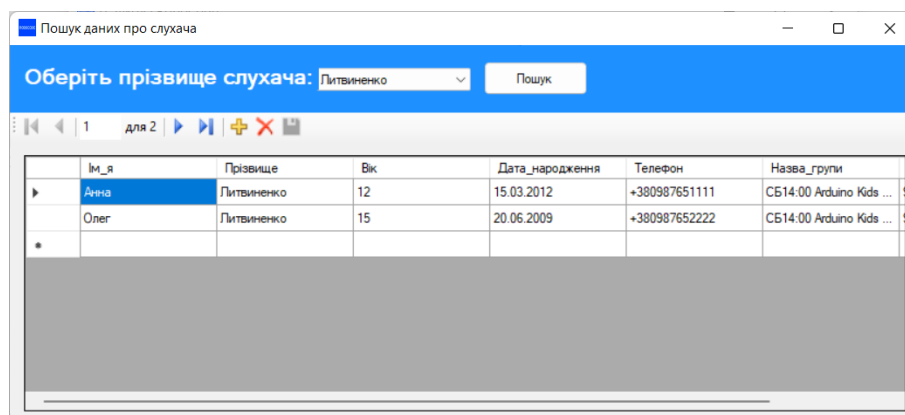
Перейшовши у вкладку головного меню «Пошук | Зведені дані», користувач може отримати повну інформацію про роботу викладачів, кількість дітей в групі, успішність учнів, загальну суму продажів курсів, та деяку інформацію по заняттям. Приклад запити «Дані про викладача» зображено на рисунку 3.36.

Також за допомогою запитів з параметром, можна здійснити пошук інформації по слухачам та групам. Для здійснення пошуку, слід з випадального списку обрати прізвище учня та натиснути кнопку «Пошук» (рисунок 3.37).



Вік	Телефон	Назва_категорії	Кількість_проведення	Зарплата
35	+380987654321	Junior teach	2	360
28	+380987654322	Junior teach	1	180
40	+380987654323	Middle teach	1	250
45	+380987654324	S-Junior teach	1	220

Рисунок 3.36 – Запит, який виводить дані про викладача



Ім'я	Прізвище	Вік	Дата_народження	Телефон	Назва_групи
Анна	Литвиненко	12	15.03.2012	+380987651111	СБ14:00 Arduino Kids ...
Олег	Литвиненко	15	20.06.2009	+380987652222	СБ14:00 Arduino Kids ...

Рисунок 3.37 – Результат пошуку через параметричний запит, даних про слухача

Форми всіх запитів, та результати пошуків представлено в додатку Г, рисунок Г.20-25.

В останній вкладці головного меню «Друковані документи», можна переглянути звітну документацію, для подальшого аналізу. Приклад звіту який виводить дані по викладачам, представлено на рисунку 3.38.

Телефон	Назва категорії	Кількість проведених занять	Зарплата
+380987654322	Junior teach	1	180
+380987654324	S-Junior teach	1	220
+380987654321	Junior teach	2	360
+380987654323	Middle teach	1	250

Рисунок 3.38 – Звіт по викладачам

Документ можна одразу роздрукувати, натиснувши на позначку принтера у верхній навігаційній панелі, або експортувати до Word, Excel та PDF (рисунок 3.39).

	Прізвище	Ім'я	Вік	Телефон	Назва категорії	Кількість проведених занять	Зарплата
1	Іванова	Марія	28	+380987654322	Junior teach	1	180
2		Оксана	45	+380987654324	S-Junior teach	1	220
3	Петров	Олексій	35	+380987654321	Junior teach	2	360
4	Сидоров	Василь	40	+380987654323	Middle teach	1	250

Рисунок 3.39 – Експортований звіт по викладачам в Excel

Приклади усіх інших звітів, наведено в додатку Г, рисунок Г.26-30.

### 3.4. Технічне та системне забезпечення розробки

#### 3.4.1. Обґрунтування вибору технічних засобів

Для розробки системи, з технічних засобів використовувався виключно персональний комп'ютер, на якому були встановлені такі програми для

підтримки розробки, як: CASE-засіб AllFusion Process Modeler та ERwin Data Modeler, MS SQL Server 2019 та Microsoft Visual Studio 2019.

Для забезпечення подальшого коректного функціонування системи, відділ освіти має бути оснащений певною кількістю ПК (в нашому випадку мінімум чотири для персоналу відділу та один для загального сервера). Крім того, необхідно встановити комутатор, який дозволить об'єднати ці комп'ютери в єдину мережу.

### 3.4.2. Розрахунок та визначення топології комп'ютерної мережі

Комп'ютерна мережа — це група комп'ютерів, з'єднаних каналами зв'язку в єдину систему для забезпечення обміну даними між ними [13]. Способи з'єднання комп'ютерів можуть відрізнятися. Логічні та фізичні методи з'єднання ПК, називають топологією.

У нашому випадку мережева топологія має форму зірки, а її структуру можна побачити на рисунку 3.40.

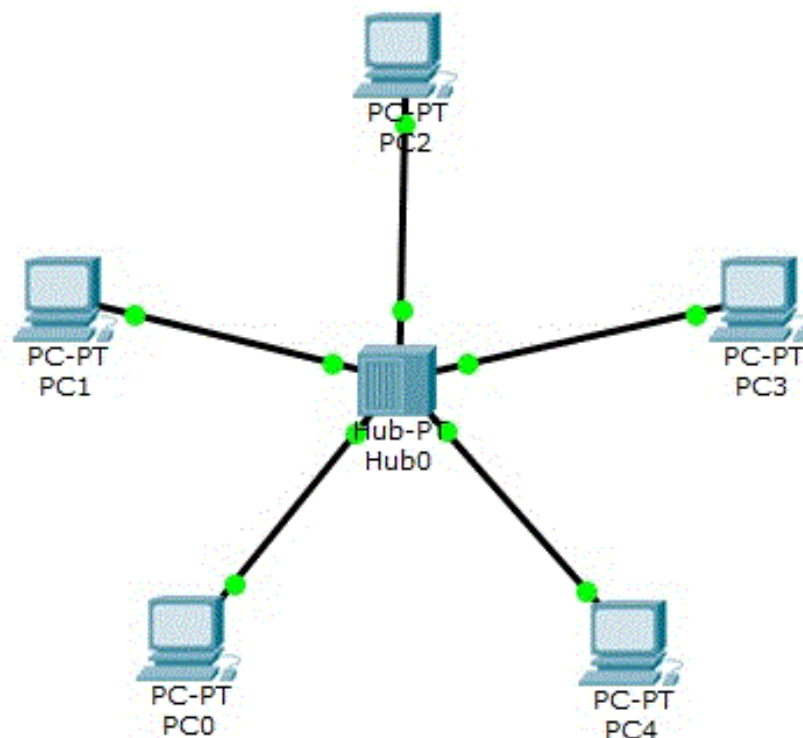


Рисунок 3.40 – Зіркоподібна топологія комп'ютерної мережі

Всі пристрої в нашій мережі підключені до центрального пристрою за топологією «точка – точка». Центром такої мережі може бути хаб (hub) або комутатор (switch). Зіркову топологію часто використовують у сучасних мережах Ethernet, так як, в ній простіше виявити помилки. Однак ця топологія вимагає багато кабелю, тому зазвичай використовується дешева витута пара UTP. Однією особливістю такої мережі є те, що в разі несправності центрального пристрою вся мережа комп'ютерів припиняє своє функціонування.

### **3.4.3. Обґрунтування вибору ОС та протоколу обміну даними**

Система була розроблена за допомогою операційної системи Windows 10. Ця ОС є найновішою версією в лінійці Windows і має кілька переваг, таких як:

- Windows 11 має сучасний і естетичний дизайн з закругленими кутами вікон;
- система оптимізована для більш швидкої та ефективної роботи;
- нові функції, такі як Snap Layouts, Snap Groups і Virtual Desktops, дозволяють зручніше керувати відкритими вікнами та робочими просторами;
- покращені функції безпеки, включаючи апаратні вимоги для безпечного завантаження та TPM 2.0, забезпечують додатковий рівень захисту;
- ця версія підходить для використання як на настільних комп'ютерах, так і на портативних пристроях;
- оновлення виконуються автоматично у фоновому режимі, що не потребує перезавантаження ПК та не перешкоджає роботі на комп'ютері.

Головною причиною вибору операційної системи Windows 10 було те, що всі використані програмні продукти для розробки інформаційних систем працюють інтегровано з нею та можуть безперебійно функціонувати.

### **3.4.4. Розробка і обґрунтування стратегії адміністрування системи**

Системне адміністрування охоплює ряд дій, які включають конфігурацію, керування та забезпечення працездатності комп'ютерного обладнання, а також забезпечення безпеки інформаційних систем та організацію даних [14].

Основна мета адміністрування системи полягає в наданні користувачам відповідних прав доступу до можливостей роботи з інформаційною системою. В нашому випадку користувачами системи є менеджер та викладач. Кожен має в доступі певні можливості.

Функції, які доступні менеджеру:

- створення, перегляд, коригування та видалення записів;
- внесення даних про викладачів, учнів, груп, розклад занять, курсів, напрямків та занять;
- розподіл викладачів по групам.

Функції, які доступні викладачу:

- внесення оцінок в журнал;
- перегляд створених записів та їх коригування;
- формування необхідних звітів.

#### **3.4.5. Заходи захисту від несанкціонованого доступу до системи**

Несанкціонований доступ до інформації в системі становить значну загрозу, тому для захисту даних було впроваджено механізм авторизації, який надає різні рівні доступу залежно від типу користувача. Завдяки цьому підходу, кожен користувач має доступ лише до тих функцій і даних, які відповідають його ролі та правам у системі, що значно підвищує безпеку інформаційних ресурсів.

Паролі для входу зберігаються в коді системи: для менеджера він встановлений як «G8zY7b!pQ3m@», а для викладача – «8hT\$nfG2@pX!zYq». Ці паролі повинні бути відомі лише відповідним особам. У випадку втрати пароля адміністратор системи може легко змінити його і передати нові дані необхідному співробітнику.

## РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Охорона праці — це комплекс правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів і засобів, які спрямовані на забезпечення безпеки, збереження здоров'я та працездатності людини під час роботи [15].

На підприємстві під час робочого процесу проводяться різні інструктажі з питань охорони праці. Обов'язковим є вступний інструктаж, який проводиться перед початком роботи нового співробітника. Він містить інформацію про майбутні умови праці та деталі щодо всіх небезпечних або шкідливих виробничих факторів, а також їх можливий вплив на життя та здоров'я працівника.

За проведення вступного та інших інструктажів у компанії відповідають керівники підрозділів. Як правило, повторні інструктажі проводяться щоквартально або при зміні правил охорони праці, зміні обладнання, чи у випадку порушення працівником правил безпеки. Всі дані про проведення інструктажів заносяться до спеціального журналу і підтверджуються підписами осіб, які проводили інструктаж, та тих, хто його прослухав.

### 4.1. Вимоги до приміщення

Щоб забезпечити комфортні умови для працівників підприємства, потрібно дотримуватися санітарно-гігієнічних вимог щодо робочих умов в приміщеннях [16]:

- площа приміщення повинна складати щонайменше 6,0 м<sup>2</sup> на кожне робоче місце. Робочі місця повинні розташовуватися на відстані не менше 1 метра від стіни з вікном і 1,4 метра від звичайної стіни. Відстань між боковими сторонами комп'ютерів має бути щонайменше 1,2 метра, а відстань між задньою частиною одного комп'ютера і екраном іншого повинна бути не менше 2,5 метра;
- у приміщеннях, де працюють з комп'ютерами, щоденно слід проводити вологе прибирання, щоб уникнути запиленості підлоги та меблів. Крім цього, необхідно обладнати кімнату для психологічного розвантаження.

- конструкція робочого столу і крісла для користувача персонального комп'ютера повинна забезпечувати підтримання оптимальної робочої пози та сприяти зручному розміщенню використовуваного обладнання (дисплей, клавіатура, принтер) і документів на робочій поверхні;
- у робочих зонах слід підтримувати оптимальні мікрокліматичні умови: температура повітря повинна бути в межах 22-25°C, відносна вологість повітря – 40-60%, а швидкість руху повітря – не більше 0,1 м/с. Якщо ці параметри мікроклімату не дотримуються, робочий день працівників має бути скорочений щонайменше на 10%;
- природне освітлення повинно забезпечувати коефіцієнт природної освітленості (КПО) не менше 1,5%. Для регулювання рівня природного світла бажано використовувати жалюзі. Робоче місце з комп'ютером має бути розташоване так, щоб уникнути прямого сонячного світла в очі;
- штучне освітлення в приміщенні повинно бути рівномірним. Використання світильників без розсіювачів і захисних сіток забороняється. Рівень освітленості на робочому столі в зоні розміщення документів повинен бути в межах 300-500 люкс;
- у офісних приміщеннях регламентуються також допустимі рівні звуку: для програмістів – 50 дБА, а для операторів у залах обробки інформації на ПК і операторів комп'ютерного набору – 65 дБА;
- вимоги щодо рівнів неіонізуючих електромагнітних випромінювань, електростатичних і магнітних полів, а також інтенсивності інфрачервоного та ультрафіолетового випромінювань визначаються згідно з ДСанПіН 3.3.2.007-98 і ДСанПіН 3.3.6.096-2002.

#### **4.2. Вимоги безпеки роботи з технікою**

Оскільки робота на підприємстві неможлива без використання обчислювальної техніки, існують певні правила, яких слід дотримуватися, щоб уникнути неприємних ситуацій.

Перед початком роботи потрібно виконати такі вимоги:

- оглянути обладнання та електропроводку, переконатися у їхній справності та готовності до роботи;
- перевірити рівень освітленості робочого місця;
- упевнитися у справному стані електричного шнура та вилки;
- у разі виявлення будь-яких несправностей не приступати до роботи та негайно повідомити керівника;
- увімкнути комп'ютер, перевірити чіткість зображення на моніторі та переконатися у відсутності запаху диму від ПК.

Після проведення перевірки обладнання можна розпочинати роботу, дотримуючись правил безпеки, зокрема:

- вмикати, вимикати комп'ютер та іншу техніку слід лише вимикачами, забороняється витягувати вилку з розетки;
- не знімати захисні пристрої з обладнання;
- не переміщати обладнання, що знаходиться під напругою;
- не залишати обладнання увімкненим без нагляду;
- не вживати напоїв та їжі під час роботи з технікою;
- не класти будь-які предмети на обладнання;
- у разі виникнення несправностей, не намагатися самостійно ремонтувати апаратуру;
- повідомляти про несправності керівника чи інженера з обслуговування техніки;
- суворо дотримуватися правил безпеки при роботі з електроприладами та вимог пожежної безпеки;
- не допускати до техніки осіб, які не пройшли інструктаж з техніки безпеки.

Після закінчення роботи з комп'ютерною технікою слід виконати такі дії:

- вимкнути всю техніку від електромережі: спочатку тумблери, а потім обережно витягти штепсельні вилки з розетки;
- чистою вологою ганчіркою протерти зовнішні поверхні апаратури від пилу та бруду;

- прибрати своє робоче місце;
- провітрити приміщення.

Керівництво підприємства або підрозділу повинно регулярно проводити збори для обговорення правил техніки безпеки, і тільки після підпису в документі про прослухані правила допускати персонал до роботи з обладнанням.

### **4.3. Вимоги безпеки життєдіяльності при аварійних ситуаціях**

Іноді аварійні ситуації можуть трапитися несподівано під час виконання робочих обов'язків. У таких випадках важливо не панікувати і дотримуватися чітких інструкцій, щоб уникнути подальших ускладнень.

Якщо раптово припинилася подача електроенергії, апаратуру необхідно вимкнути з розетки. Якщо з'явився незрозумілий звук або запах горілого, потрібно негайно вимкнути обладнання і повідомити про це керівництво.

У разі виникнення пожежі слід швидко вимкнути техніку, знеструмити електромережу, сповістити всіх співробітників і евакуювати їх. Необхідно також викликати пожежну службу і, якщо можливо, самотійно приступити до гасіння вогню.

Якщо стався нещасний випадок, потрібно допомогти постраждалому вибратися і доставити його до медпункту, залишивши місце травмування в тому стані, в якому воно було на момент події. У разі нещасного випадку, пов'язаного з електрострумом, слід бути особливо обережним, щоб не потрапити під напругу самому, надаючи допомогу постраждалому [17].

### **4.4. Рекомендації до роботи за комп'ютерною технікою**

Робота у відділі освіти включає постійне використання комп'ютерної техніки, що може бути небезпечним для здоров'я. Тривала робота за комп'ютером значно навантажує зорову систему, а також впливає на нервову, розумову та опорно-рухову системи людини.

Щоб уберегти працівників від негативних наслідків, їм необхідно дотримуватися таких правил під час роботи з комп'ютером у сидячому положенні [18]:

- сидіти слід прямо, тримаючи спину рівною;
- комп'ютер повинен бути розташований на відстані 70 см;
- забороняється класти ногу на ногу, стопи повинні міцно стояти на підлозі;
- між користувачем та краєм столу має бути вільний простір щонайменше 5 см;
- коліна під столом повинні бути зігнуті під кутом 90 градусів, а відстань до крісла має дорівнювати розміру кулака;
- стіл повинен бути на 2-3 см вище рівня ліктів, плечі мають бути розслаблені, а руки повинні бути зігнуті під кутом 90 градусів;
- не варто часто використовувати підлокітники, якщо вони розташовані під неправильним кутом;
- потрібно регулярно робити перерви, щоб трохи розім'ятися.

Правильне положення сидячої людини за комп'ютером зображено на рисунку 4.1.



Рисунок 41 – Правильне розміщення тіла сидячої людини за ПК

#### 4.5. Вимоги пожежної безпеки

Усі вимоги щодо пожежної безпеки викладені в Законі України «Про пожежну безпеку». Цей документ містить основні правові та соціальні засади забезпечення пожежної безпеки на території України.

Пожежі часто виникають через неправильне поводження з вогнем та несправний стан електроприладів. Щодня стається багато пожеж, внаслідок яких страждає майно і гинуть люди. Під час пожежі люди піддаються небезпеці через вогонь, дим, нестачу кисню, токсичні продукти згоряння та вибухи. Це може призвести до тяжких ушкоджень організму та тіла, які можуть бути несумісні з життям. Щоб уникнути таких наслідків, необхідно обов'язково дотримуватися наступних вимог [19]:

- розробити комплекс заходів для забезпечення пожежної безпеки;
- створити детальні інструкції та правила з пожежної безпеки;
- організувати регулярні збори колективу для навчання правилам пожежної безпеки;
- підтримувати електромережі у належному стані, періодично їх перевіряючи;
- скласти план евакуації співробітників на випадок пожежі.

Всі приміщення підприємства повинні бути оснащені необхідним обладнанням для гасіння пожеж. Це включає вуглекислотні вогнегасники, зовнішнє та внутрішнє водопостачання. Вогнегасники повинні бути розміщені таким чином, щоб на кожні 20 м<sup>2</sup> площі приміщення припадало по 2 вогнегасники. Відстань від можливого осередку пожежі до місця розташування вогнегасника не повинна перевищувати 20 метрів [20].

Системи зовнішнього протипожежного водопостачання, такі як гідранти, повинні бути розміщені на території таким чином, щоб відстань по периметру будівель вздовж доріг не перевищувала 100 метрів, а від стін – не менше 5 метрів [21]. Внутрішні системи водопостачання, у вигляді пожежних кранів, встановлюються всередині приміщень біля виходів і на сходових клітинах на

висоті 1,35 метра від підлоги. Пожежний кран включає прогумований рукав довжиною близько 15 метрів і пожежний ствол.

Оскільки дуже важливо виявити пожежу на початковій стадії, приміщення також повинні бути оснащені автоматичною пожежною сигналізацією та автоматичною системою пожежогасіння з димовим індикатором.

## ВИСНОВКИ

Головним завданням дипломної роботи було створення інформаційної системи для підтримки навчального процесу ІТ-школи Robocode ФОП «ЮШКОВ Дмитро Сергійович».

Для ефективної підтримки діяльності відділу освіти на різних етапах розробки системи використовувалися такі програмні засоби: CASE-засіб AllFusion Process Modeler для аналізу діяльності відділу, CASE-засіб AllFusion ERwin Data Modeler для проектування бази даних на двох рівнях та її генерування, MS SQL Server 2019 для адміністрування бази даних, а також Microsoft Visual Studio 2019 для розробки додатку мовами C# та SQL.

Розроблена система значно спрощує роботу співробітників відділу, оскільки забезпечує зручний доступ до даних, можливість додавання та видалення інформації, а також здійснення пошуку та фільтрації необхідних записів за допомогою запитів.

Крім цього, розроблена ІС дозволяє автоматично підраховувати заробітну плату викладачам, та миттєво отримати звітність, щодо їх праці. Це значно спрощує керування навчальним процесом, так як не потрібно витрачати час, на підрахунок заробітної плати кожного викладача окремо.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. А. О. Литвинчук РОЗВИТОК ІНФОРМАЦІЙНИХ СИСТЕМ УПРАВЛІННЯ ОСВІТОЮ ЯК ІНСТРУМЕНТ РЕАЛІЗАЦІЇ ДЕРЖАВНОЇ ОСВІТНЬОЇ ПОЛІТИКИ / Г. М. Терещенко, А. В. Кир'янов. – Київ: Державна наукова установа «Інститут освітньої аналітики», 2020. – 259 с. – (НАУКОВЕ ВИДАННЯ).
2. Розвиток Robocode в Україні [Електронний ресурс] – Режим доступу до ресурсу: <https://kyiv.dityvmisti.ua/shkola-robototehniki-i-programmirovaniya-robocode/>.
3. Основні напрямки програмування [Електронний ресурс] – Режим доступу до ресурсу: <https://robocode.ua/>.
4. Команда Robocode [Електронний ресурс] – Режим доступу до ресурсу: <https://robocode.ua/vacancies-ua>.
5. Опис СА ERwin Process Modeler [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/6074887/page:2/>.
6. IDEF0. Знайомство з нотацією та приклад використання [Електронний ресурс] – Режим доступу до ресурсу: <https://trinion.org/blog/idef0-znakomstvo-s-notaciey-i-primer-ispolzovaniya>.
7. Опис платформи Moodle [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.moodle.org/404/en/Features>.
8. Опис платформи Prosvita [Електронний ресурс] – Режим доступу до ресурсу: <https://prosvita.net/>.
9. Опис платформи Canvas [Електронний ресурс] – Режим доступу до ресурсу: <https://www.instructure.com/>.
10. Використання моделей AS-IS та TO-BE [Електронний ресурс] – Режим доступу до ресурсу: <https://studbooks.net/1173398/informatika/model>.
11. М'якшило О.М. Моделювання баз даних засобами CASE – технології ERWin: конспект лекцій/ О.М. М'якшило – К.:НУХТ, 2007 – 60 с.

**12.** Проектування інформаційних систем: лабораторний практикум для студ. освіт. ступ. "Бакалавр" спец. 122 "Комп'ютерні науки " ден. і заоч. Форм навч. Частина 2 "Проектування клієнтського додатку" / уклад. : О. М. М'якшило, О. В. Харкянєн; Нац. ун-т харч. технол. - Київ : НУХТ, 2017. - 33 с.

**13.** Комп'ютерні мережі [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/txhxp>.

**14.** Системне адміністрування [Електронний ресурс] – Режим доступу до ресурсу: <https://vistplus.com/it-poslugi/sistemne-administruvannya/>.

**15.** Охорона праці [Електронний ресурс] – Режим доступу до ресурсу: [https://vk24.ua/practical\\_work/okhorona-praci](https://vk24.ua/practical_work/okhorona-praci).

**16.** Робота в офісі: основні санітарно-гігієнічні вимоги [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/txhze>.

**17.** ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ : дис. докт. техн. наук : 133 / Левченко Олег Григорович – Київ, 2019. – 30 с.

**18.** Правила сидіння за робочим комп'ютером [Електронний ресурс]. – Режим доступу до ресурсу: <https://jysk.ua/blog/yak-pravilno-siditi-za-robochim-stolom>.

**19.** ПОЖЕЖНА БЕЗПЕКА. НЕОБХІДНІ ДІЇ ТА ЗАХОДИ КЕРІВНИКІВ ТА УЧАСНИКІВ ОСВІТНЬОГО ПРОЦЕСУ [Електронний ресурс] – Режим доступу до ресурсу: <https://info.eo.gov.ua/pozhezhna-bezpeka-neobhidni-diyi-ta-zahody-kerivnykiv-ta-uchasnykiv-osvitnogo-proczesu/>.

**20.** РЕКОМЕНДАЦІЇ ЩОДО ОСНАЩЕННЯ ОБ'ЄКТІВ ПЕРВИННИМИ ЗАСОБАМИ ПОЖЕЖОГАСІННЯ [Електронний ресурс] – Режим доступу до ресурсу: <http://edu-mns.org.ua/avtomat/lessons/25/5.html>.

**21.** Принципи дії протипожежного водопостачання [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/9990277/page:32/>.

# ДОДАТКИ

## Додаток А. Модель «AS-IS»

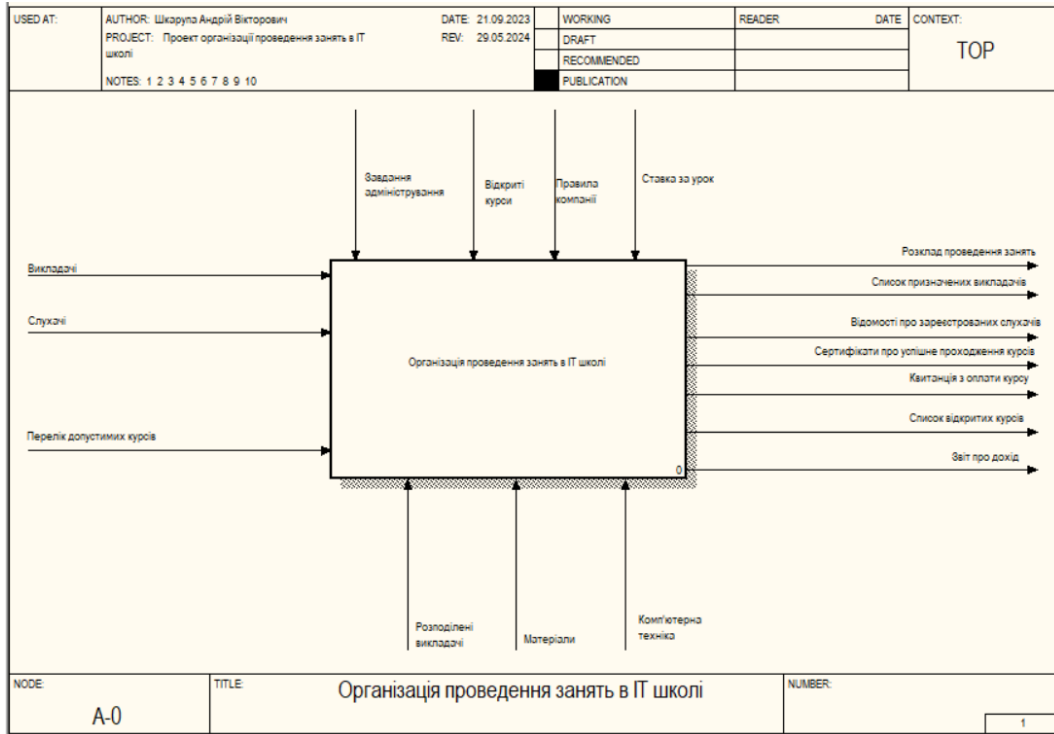


Рисунок А.1 – Модель «AS-IS». Контекстна діаграма

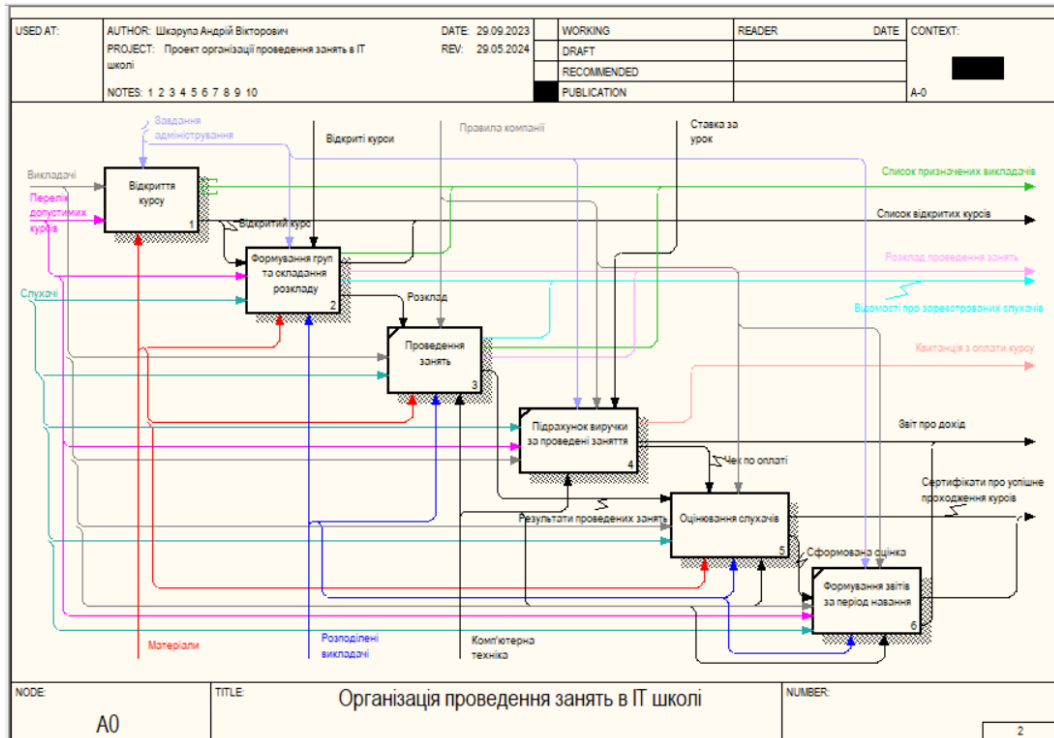


Рисунок А.2 – Модель «AS-IS». Діаграма декомпозиції на першому рівні

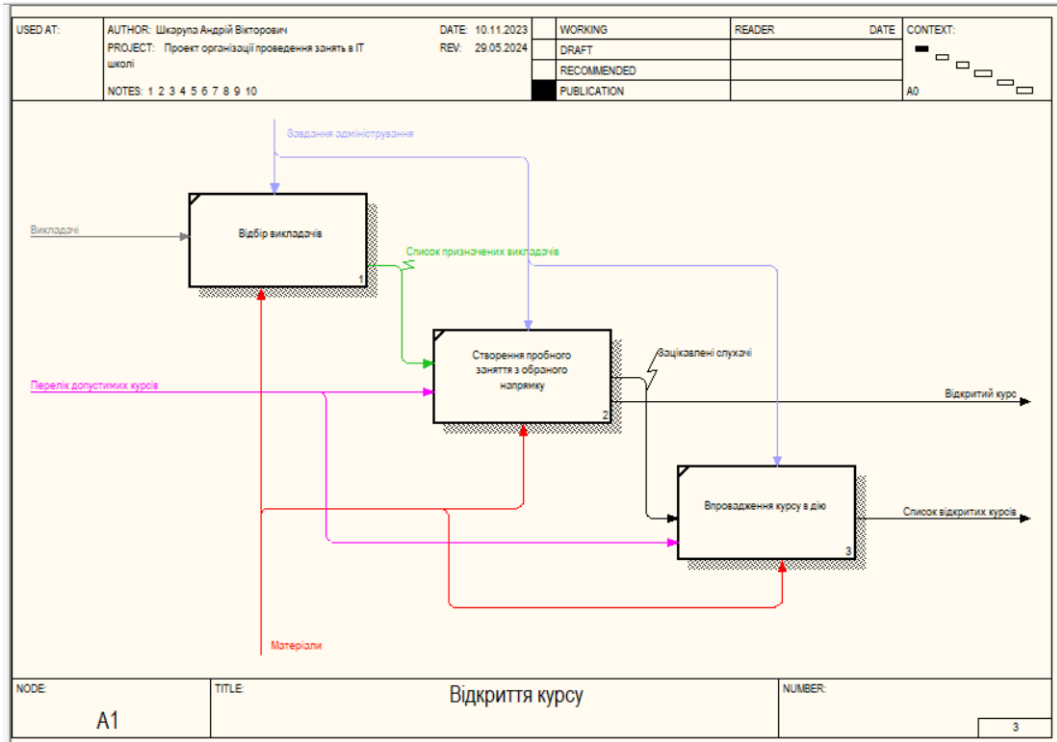


Рисунок А.3 – Модель «AS-IS». Діаграма декомпозиції блоку «Відкриття курсу»

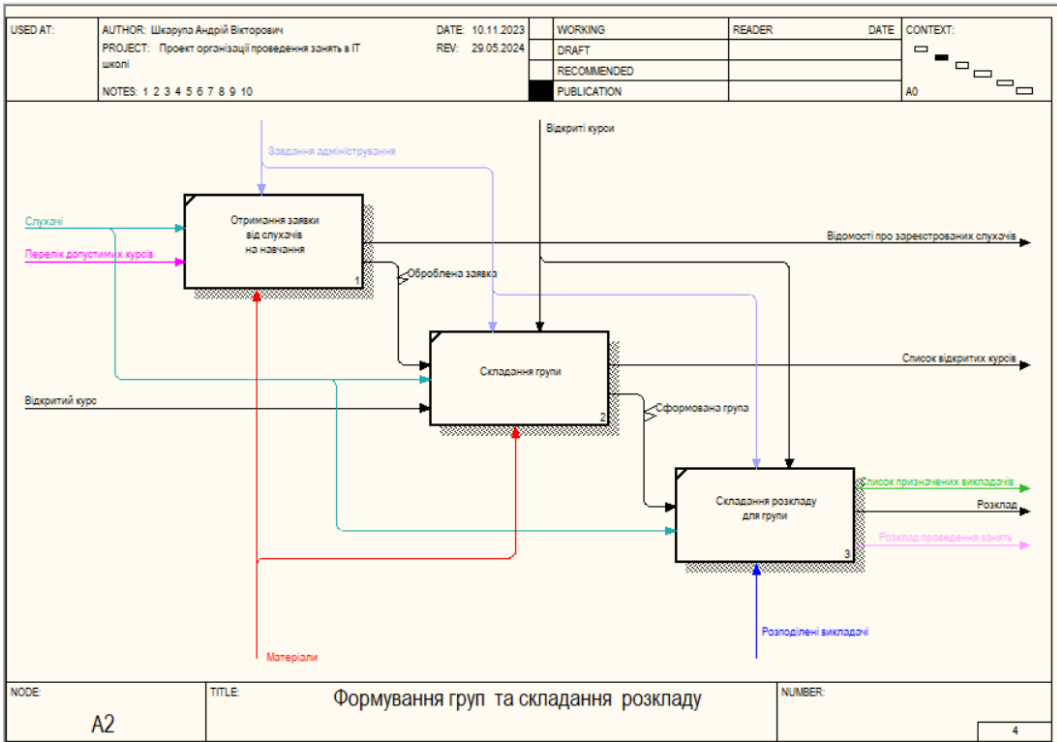


Рисунок А.4 – Модель «AS-IS». Діаграма декомпозиції блоку «Формування груп та складання розкладу»

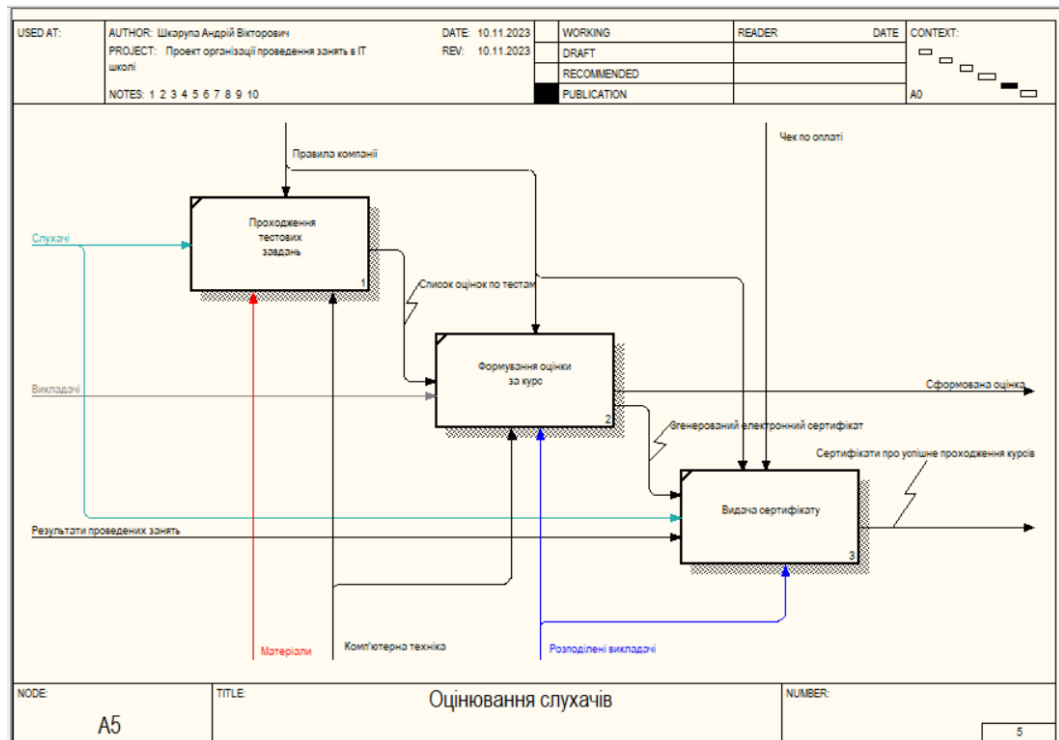


Рисунок А.5– Модель «AS-IS». Діаграма декомпозиції блоку «Оцінювання слухачів»

Додаток Б. Модель «ТО-ВЕ»

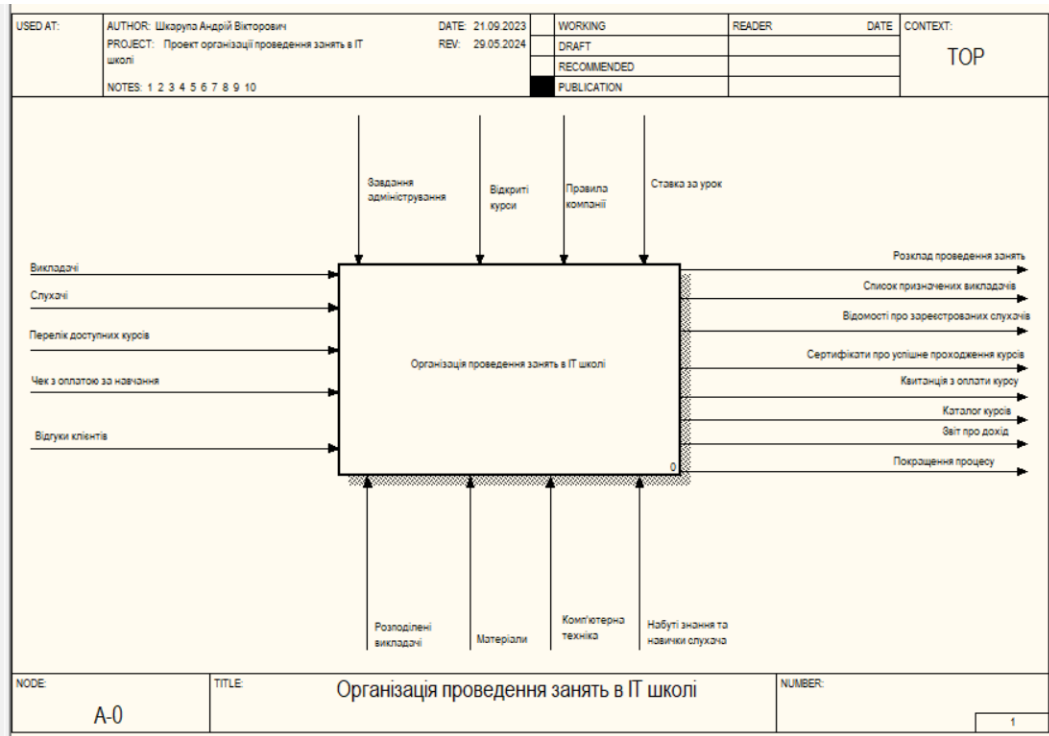


Рисунок Б.1. – Модель «ТО-ВЕ». Контекстна діаграма

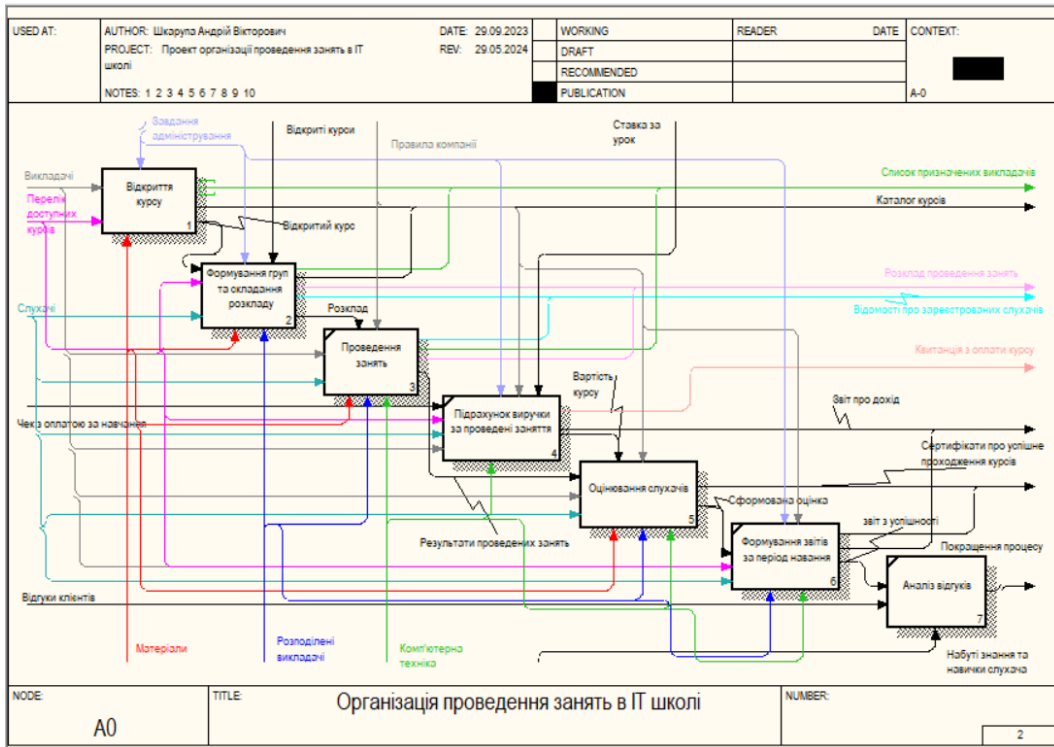


Рисунок Б.2 – Модель «ТО-ВЕ». Діаграма декомпозиції на першому рівні

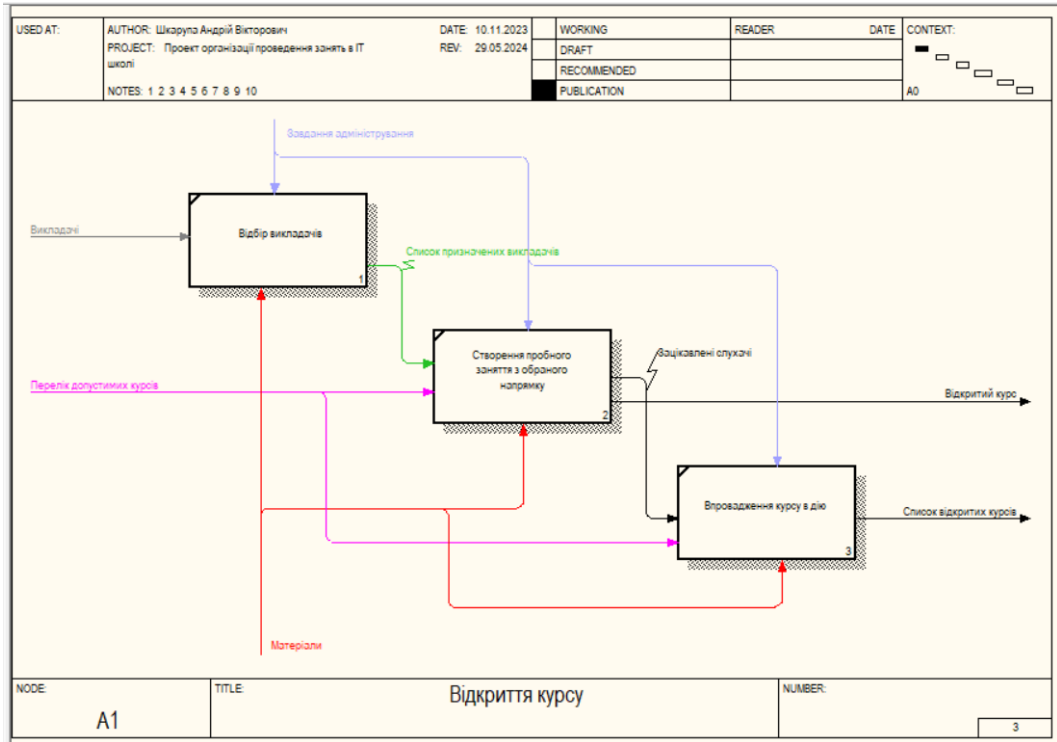


Рисунок Б.3 – Модель «ТО-ВЕ». Діаграма декомпозиції блоку «Відкриття курсу»

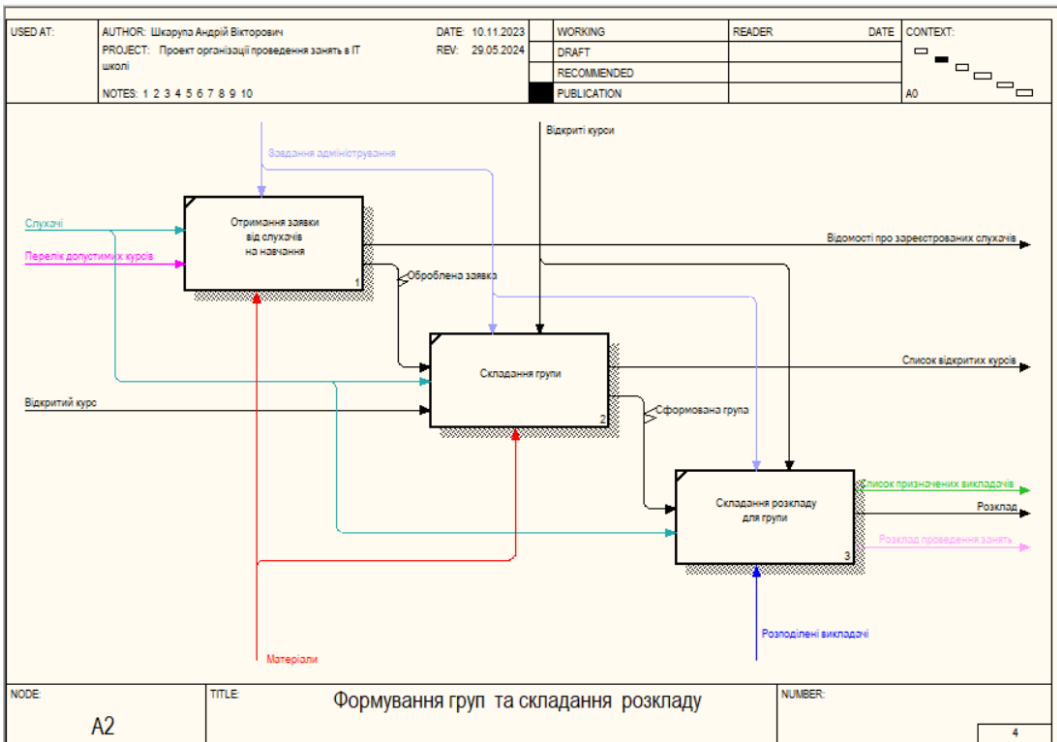


Рисунок Б.4 – Модель «ТО-ВЕ». Діаграма декомпозиції блоку «Формування груп та складання розкладу»

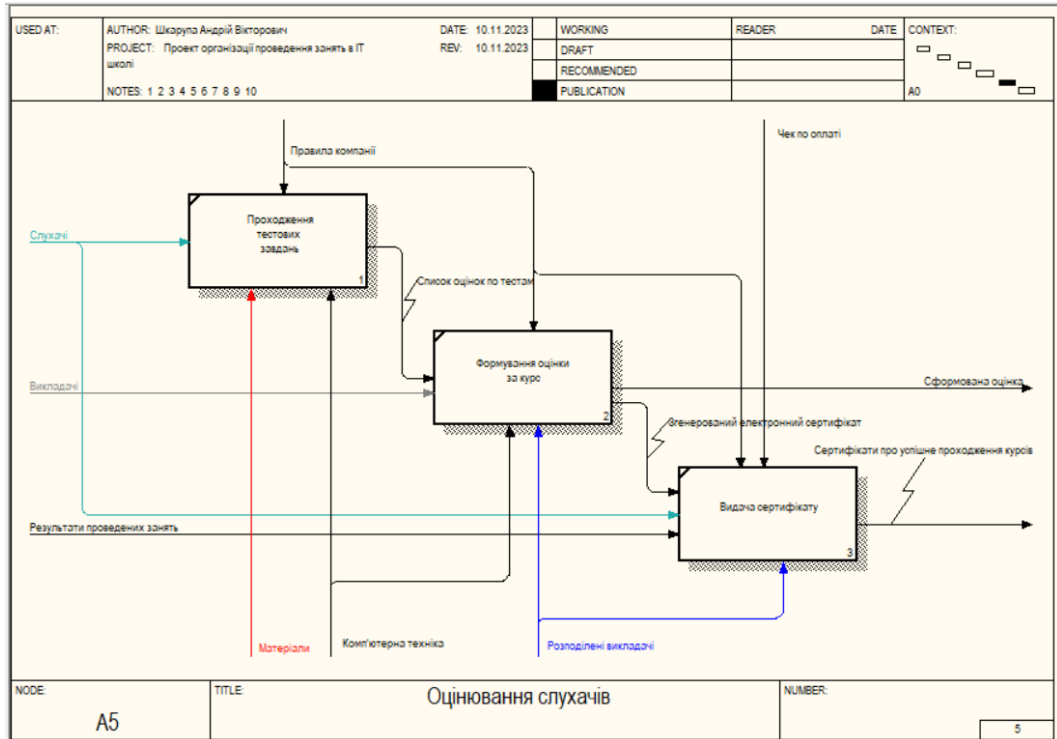


Рисунок Б.5– Модель «ТО-ВЕ». Діаграма декомпозиції блоку «Оцінювання слухачів»

## Додаток В. Схеми моделей даних

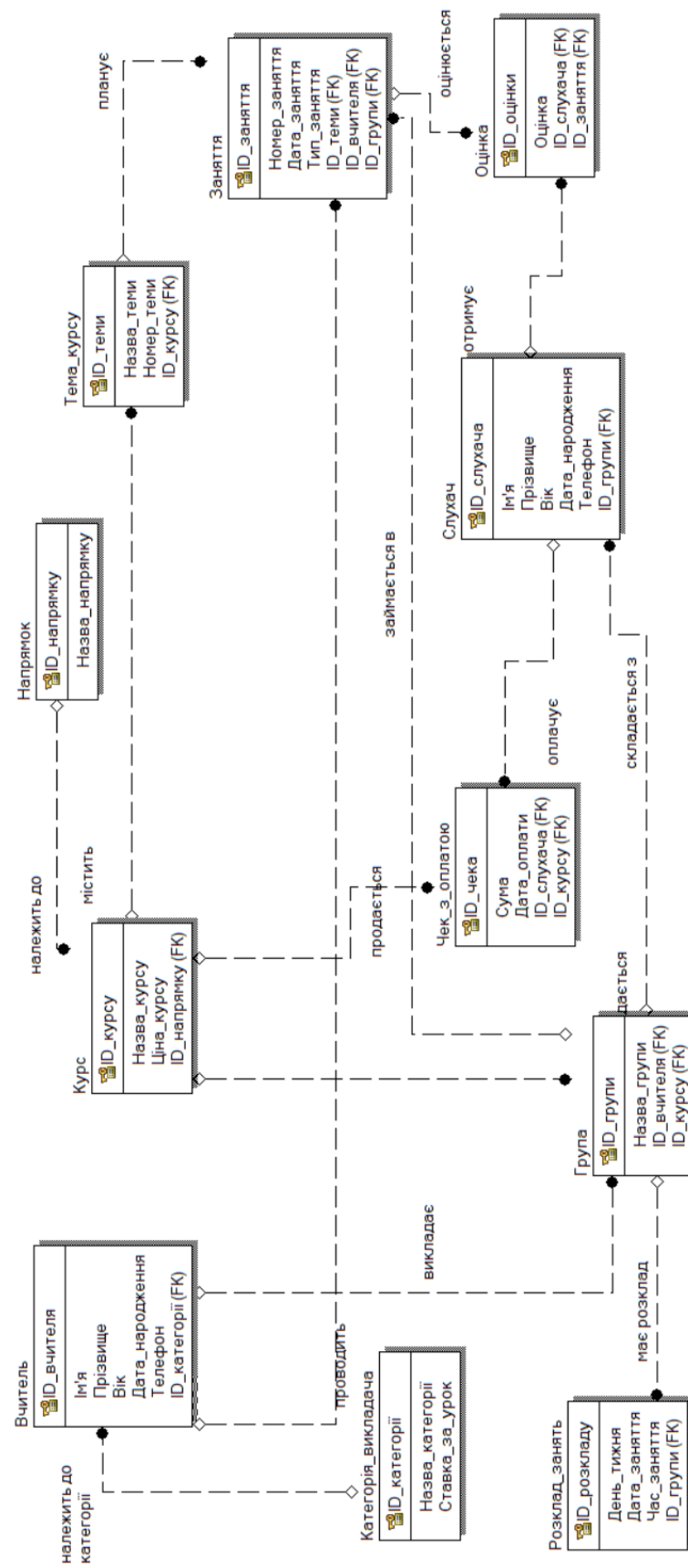


Рисунок В.1 – Логічна модель БД

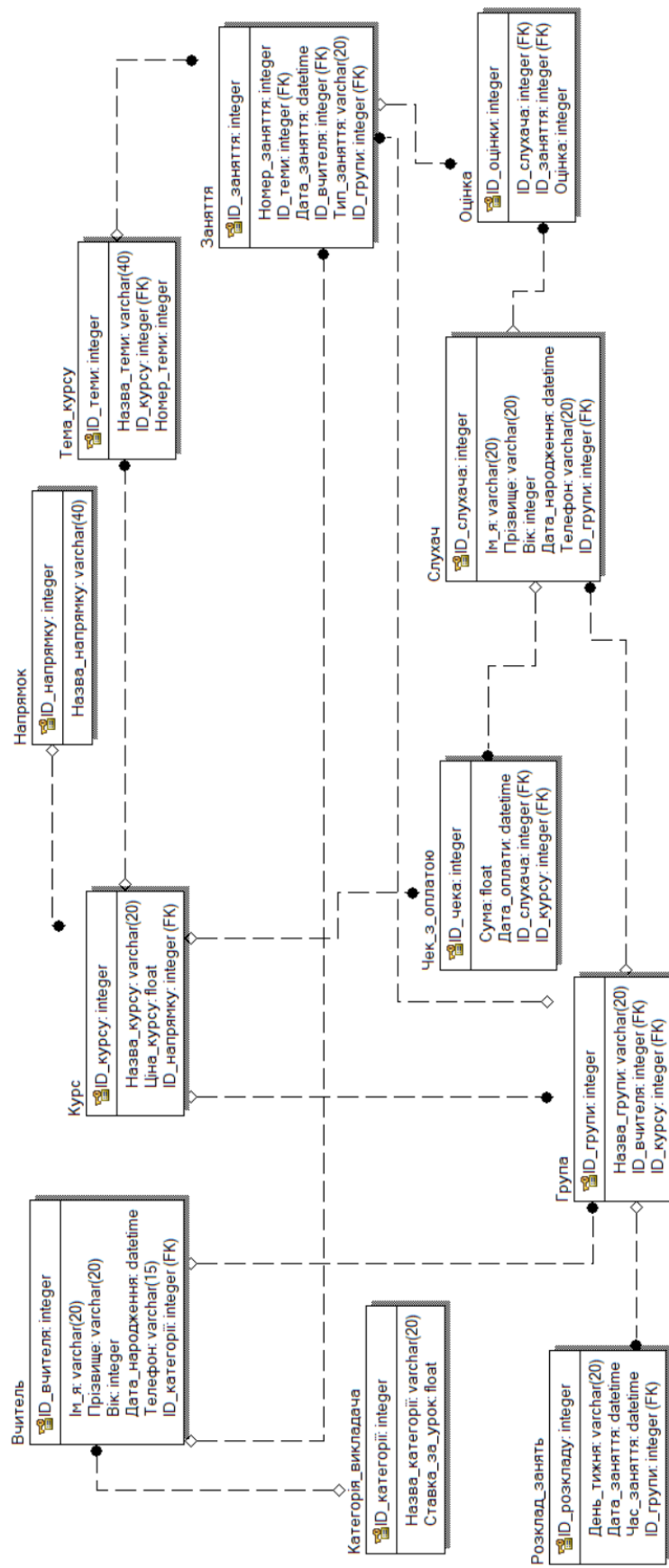


Рисунок В.2 – Фізична модель БД

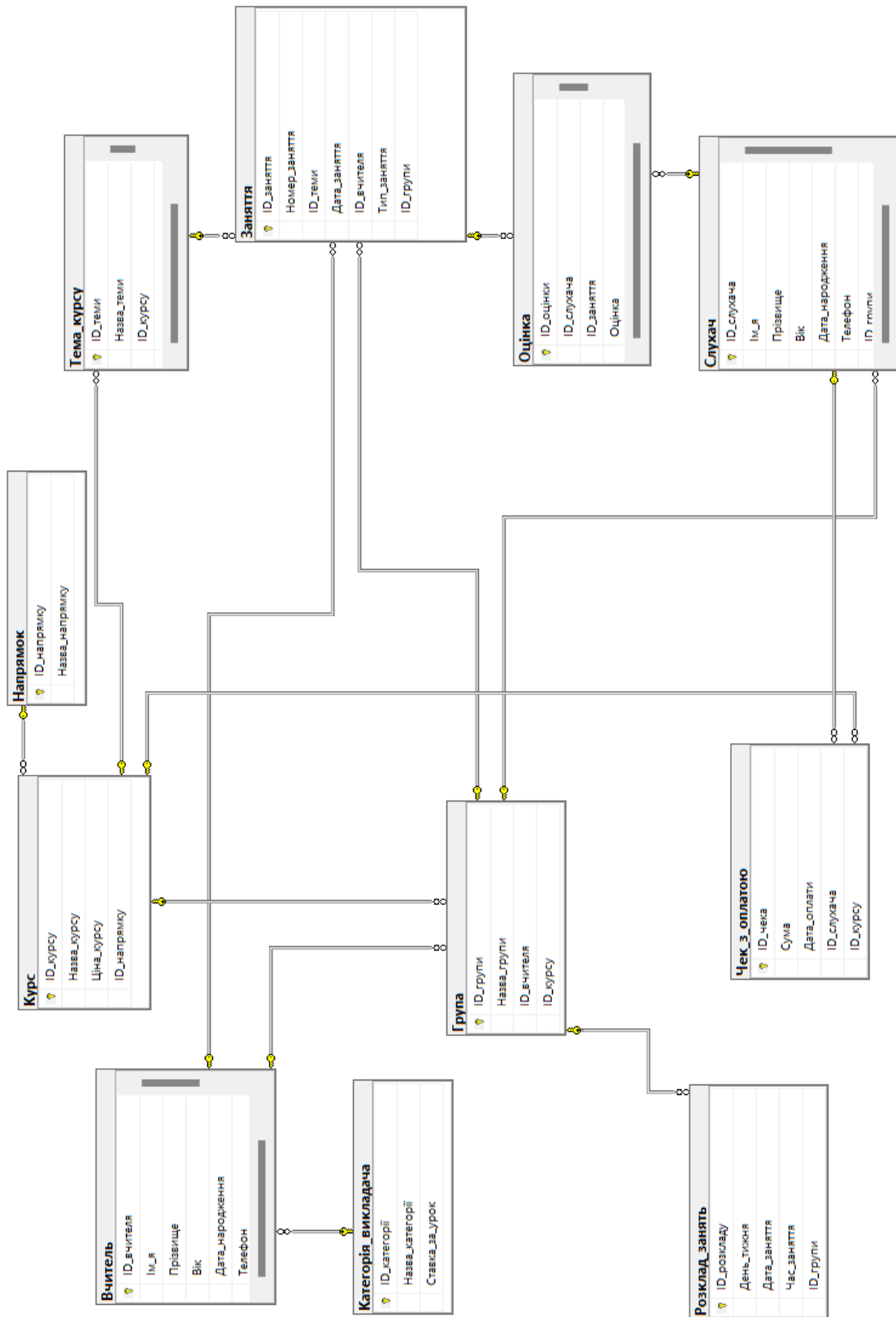


Рисунок В.3 – Схема бази даних в MS SQL Server

## Додаток Г. Скріншоти інтерфейсу



Рисунок Г.1 – Загрузочний екран системи



Рисунок Г.2 – Форма авторизації користувача

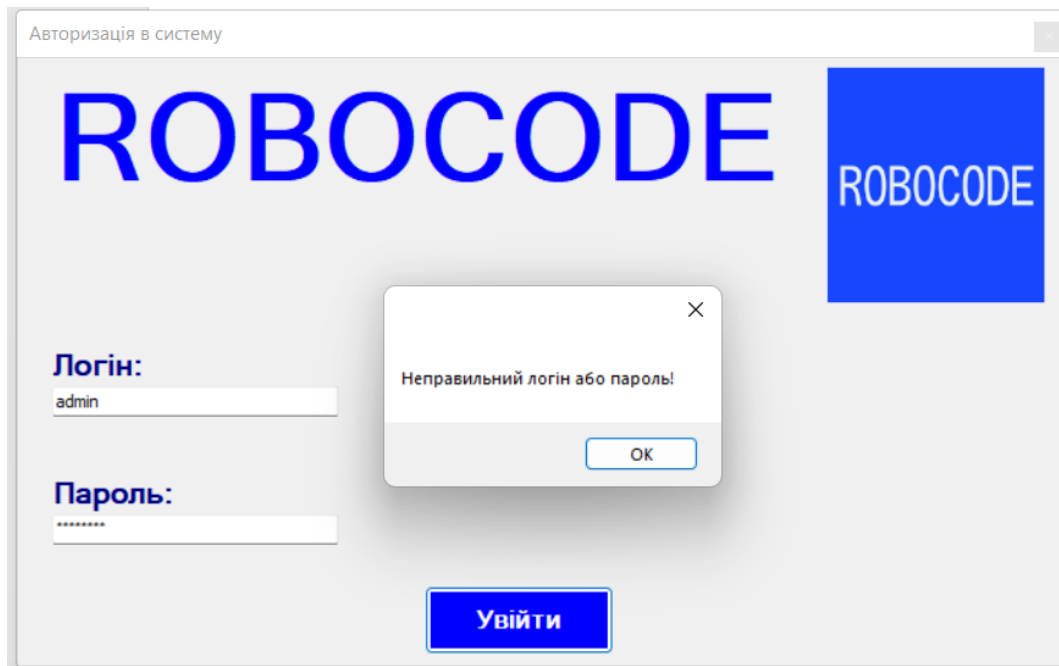


Рисунок Г.3 – Невірна спроба авторизації в систему

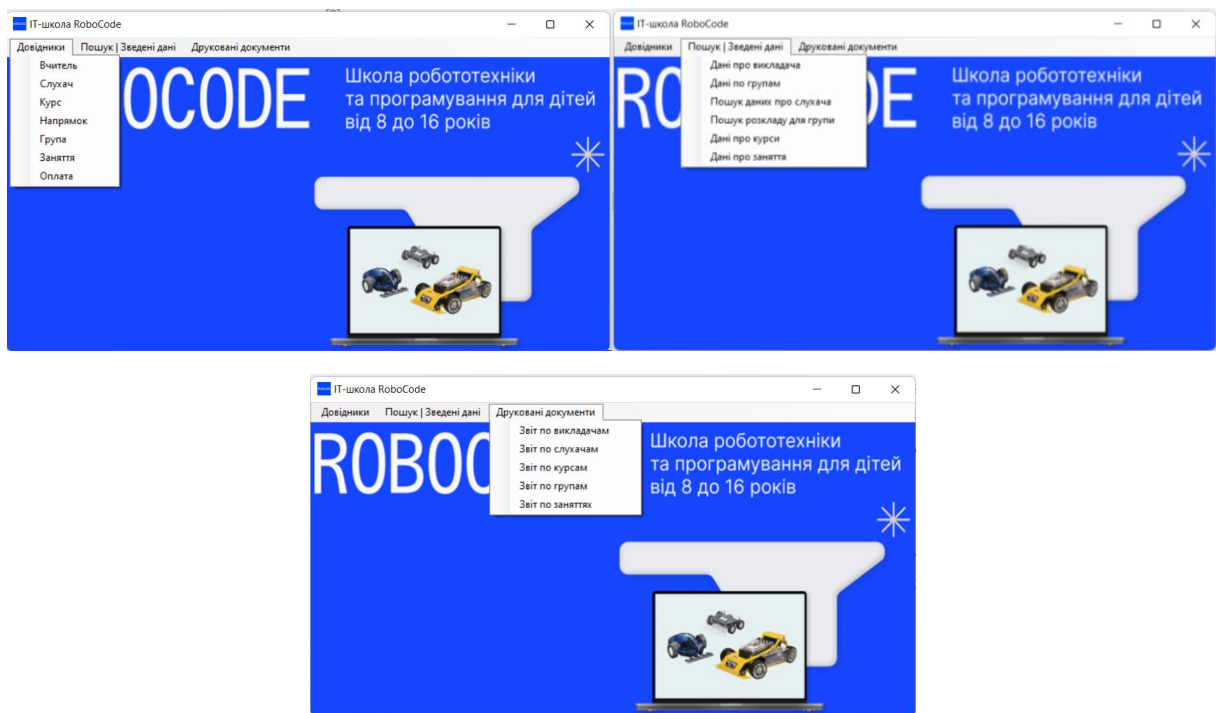


Рисунок Г.4 – Головне меню системи


Вчителі

1 для 4

ID категорії: 1

Назва категорії: Junior teach

Ставка за урок: 180



Відкрити форму для пошуку

Введіть назву категорії:

S-Junior teach

Відфільтрувати

Зняти фільтр

Відкрити

Вчителі:

	Прізвище	Вік	Дата_народження	Телефон	Назва категорії
▶	Петров	35	15.05.1989	+380987654321	Junior teach
	Іванова	28	20.10.1996	+380987654322	Junior teach
*					

Рисунок Г.5 – Форма для введення даних до таблиці «Вчителі»

Слухачі

1 для 22

ID слухача: 2

Ім'я: Анна


Прізвище: Литвиненко

Вік: 12

Дата народження: Thursday, 15 Marcl

Телефон: +380987651111

Назва групи: СБ14:00 Arduino Kids f



Введіть прізвище слухача:

Пошук

Оцінки:

	ID_оцінки	Прізвище слухача	Номер заняття	Оцінка
▶	1	Литвиненко	1	90
	5	Литвиненко	1	92
*				

Рисунок Г.6 – Форма для введення даних до таблиці «Слухачі»

Курси

1 для 19

ID курсу:

Назва курсу:

Ціна курсу:

Назва напрямку:

Введіть назву курсу:

Відфільтрувати

Зняти фільтр

Теми занять:

	ID_теми	Назва_теми	Назва курсу	Номер_теми
▶	1	Введення у Scratch 2	Arduino Kids	1
	2	S4A. Введення в роб...	Arduino Kids	8
*				

Рисунок Г.7 – Форма для введення даних до таблиці «Курси»

Напрямки

1 для 5

ID напрямку:

Назва напрямку:

Введіть назву напрямку:

Рисунок Г.8 – Форма для введення даних до таблиці «Напрямки»

Групи


1 для 9

**ID групи:**

**Назва групи:**

**Прізвище вчителя:**

**Назва курсу:**



**Введіть назву групи:**

**Пошук**

**Розклад занять:**

ID_розкладу	День_тижня	Дата_заняття	Час_заняття	Назва_групи
1	Субота	16.03.2024	10:00:00	СБ10:00 Arduino ...
*				

Рисунок Г.9 – Форма для введення даних до таблиці «Групи»

Заняття

1 для 5

**ID заняття:**

**Номер заняття:**


**Назва теми:**

**Дата заняття:**

**Прізвище вчителя:**

**Тип заняття:**

**Назва групи:**



**Введіть дату заняття:**

**Пошук**

Рисунок Г.10 – Форма для введення даних до таблиці «Заняття»

Чеки з оплатами

1 для 19

ID чека:

Сума:

Дата оплати:

Прізвище слухача:

Назва курсу:

Введіть дату оплати:

Пошук

Рисунок Г.11 – Форма для введення даних до таблиці «Чеки з оплатами»

Пошук вчителя

2 для 5

Введіть прізвище вчителя:

Пошук

	ID_вчителя	Ім_я	Прізвище	Вік	Дата_народження
	2	Олексій	Петров	35	15.05.1989
▶	3	Марія	Іванова	28	20.10.1996
	4	Василь	Сидоров	40	02.03.1982
	5	Оксана	Іванова	45	18.12.1977
	6	Андрій	Григоренко	30	25.07.1992

Рисунок Г.12 – Результат пошуку вчителя по його прізвищу

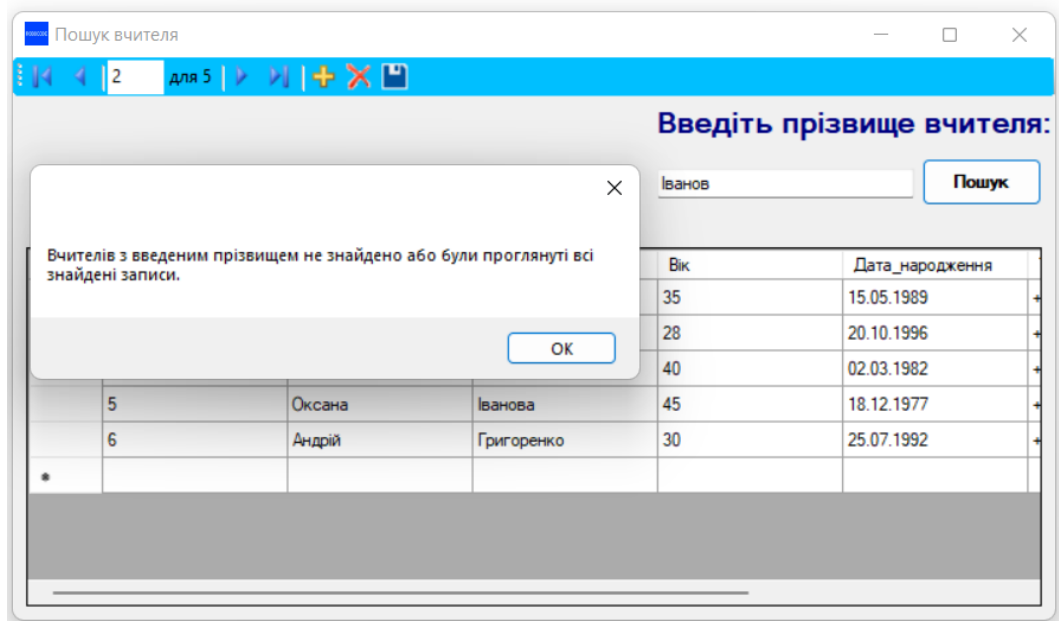


Рисунок Г.13 – Помилка пошуку, якщо вчителя з заданим прізвищем немає в БД

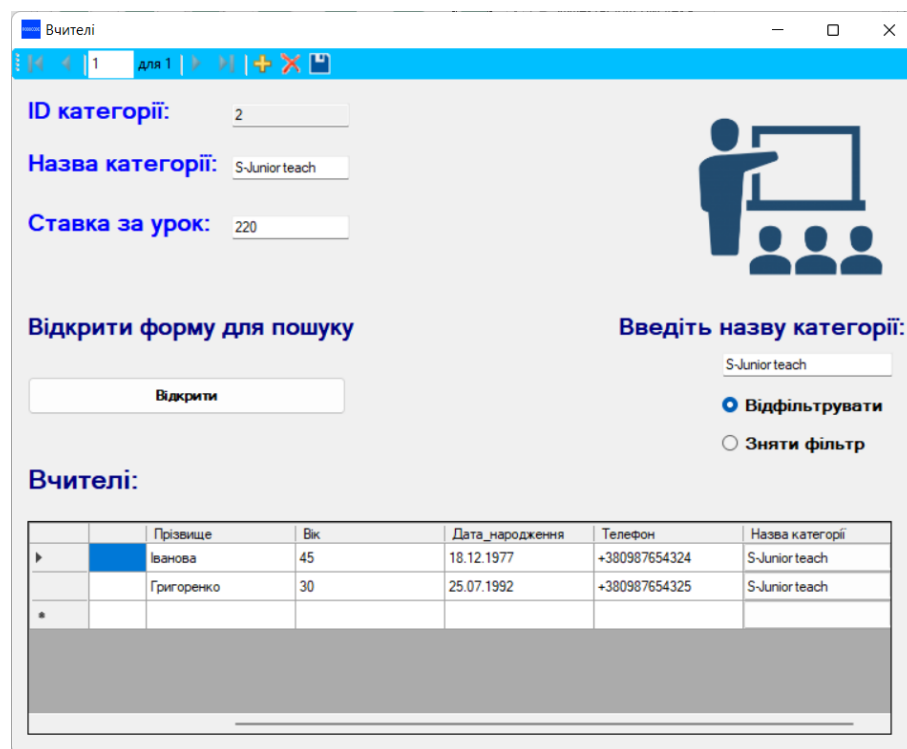


Рисунок Г.14 – Результат застосування фільтру по назві категорії

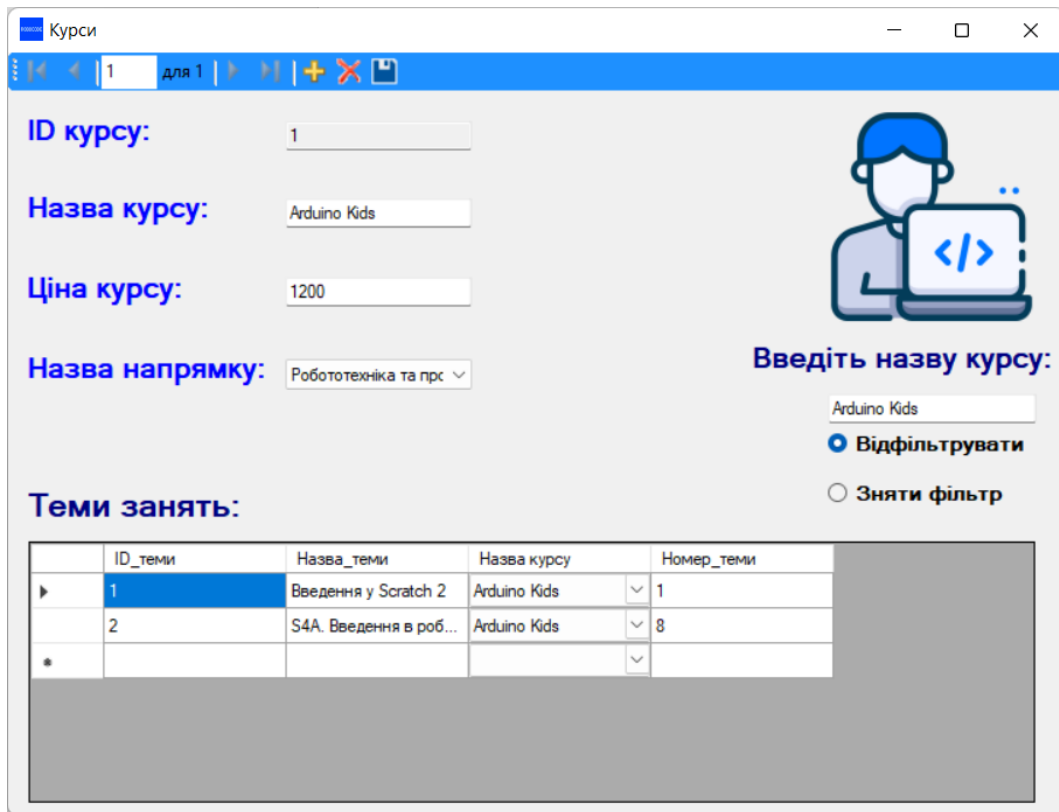


Рисунок Г.15 – Результат фільтрування по назві курсу

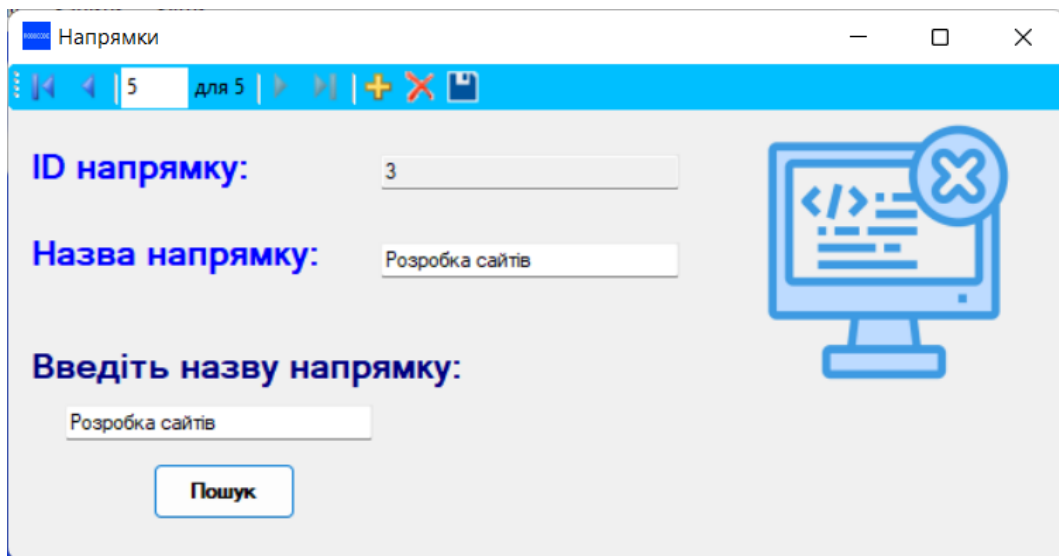


Рисунок Г.16 – Результат пошуку по назві напрямку програмування

Групи


2 для 9

**ID групи:**

**Назва групи:**

**Прізвище вчителя:**

**Назва курсу:**



**Введіть назву групи:**

**Розклад занять:**

ID_розкладу	День_тижня	Дата_заняття	Час_заняття	Назва_групи
2	Субота	17.03.2024	12:00:00	СБ12:00 Arduino ...
*				

Рисунок Г.17 – Результат пошуку групи по її назві

Заняття

4 для 5

**ID заняття:**

**Номер заняття:**


**Назва теми:**

**Дата заняття:**

**Прізвище вчителя:**

**Тип заняття:**

**Назва групи:**



**Введіть дату заняття:**

Рисунок Г.18 – Результат пошуку даних про заняття по даті проведення

Чеки з платежами

7 для 19

ID чека:

Сума:

Дата оплати:

Прізвище слухача:

Назва курсу:

Введіть дату оплати:




Рисунок Г.19 – Результат пошуку чеку оплати по даті оплати

Дані про викладача

1 для 4

Імя	Вік	Телефон	Назва_категорії	Кількість_проведених_	Зарплата
Аксій	35	+380987654321	Junior teach	2	360
Лія	28	+380987654322	Junior teach	1	180
Миль	40	+380987654323	Middle teach	1	250
Слана	45	+380987654324	S-Junior teach	1	220




Рисунок Г.20 – Форма запити, який виводить дані про викладача

Дані по групам

1 для 8

Назва_групи	Назва_напряму	Назва_курсу	Кількість_слухачів
СБ10:00 Arduino Kids	Робототехніка та про...	Arduino Kids	4
СБ12:00 Arduino Kids	Робототехніка та про...	Arduino Kids	2
НД:12:00 Arduino Kids	Робототехніка та про...	Arduino Kids	3
СБ14:00 Arduino Kids ...	Робототехніка та про...	Arduino KidsPro	2
НД:14:00 Arduino Kid...	Робототехніка та про...	Arduino KidsPro	1
СБ16:00 Construct	Розробка ігор	Construct	3
НД:16:00 Unity Begin...	Розробка ігор	Unity Beginner	4
СБ18:00 Web Beginner	Розробка сайтів	Web Beginner	3




Рисунок Г.21 – Форма запити, який виводить дані по групам

Пошук даних про слухача

Оберіть прізвище слухача:

1 для 2

Прізвище	Вік	Дата_народження	Телефон	Назва_групи	Середня_оцінка
Литвиненко	12	15.03.2012	+380987651111	СБ14:00 Arduino Kids ...	91
Литвиненко	15	20.06.2009	+380987652222	СБ14:00 Arduino Kids ...	95

Рисунок Г.22 – Форма запити з результатом пошуку учня, та показом його успішності

Дані про розклад груп

Оберіть назву групи:

1 для 1

Назва_групи	День_тижня	Дата_заняття	Час_заняття	Ім_я	Прізвище
СБ10:00 Arduino Kids	Субота	16.03.2024	10:00:00	Олексій	Петров

Рисунок Г.23 – Форма запити з результатом пошуку розкладу для групи

Дані про курси

1 для 19

Назва_курсу	Ціна_курсу	Назва_напряму	Сума_продажів
Arduino Kids	1200	Робототехніка та про...	7200
Arduino Junior	1200	Робототехніка та про...	0
Arduino KidsPro	1400	Робототехніка та про...	4200
Middle Embedded	1450	Робототехніка та про...	0
Pro Embedded	1500	Робототехніка та про...	0
Senior Embedded	1500	Робототехніка та про...	0
Expert Embedded	1500	Робототехніка та про...	0
Construct	1200	Розробка ігор	3600
Unity Beginner	1200	Розробка ігор	4800




Рисунок Г.24 – Форма запити, який виводить дані про курси

Номер_заняття	Дата_заняття	Тип_заняття	Прізвище_вчителя	Назва_групи
1			Петров	СБ10:00 Arduino Kid
8	01.09.2024	Офлайн	Іванова	СБ12:00 Arduino Kid
13	09.11.2024	Офлайн	Сидоров	СБ14:00 Arduino Kid
18	03.03.2024	Онлайн	Іванова	НД:16:00 Unity Begi
2	08.04.2024	Онлайн	Петров	СБ10:00 Arduino Kid

Рисунок Г.25 – Форма запити, який виводить дані про заняття

Прізвище	Ім'я	Вік	Телефон	Назва категорії	Кількість проведених занять	Зарплата
Іванова	Марія	28	+380987654322	Junior teach	1	180
	Оксана	45	+380987654324	S-Junior teach	1	220
Петров	Олександр	35	+380987654321	Junior teach	2	380
Сидоров	Василь	40	+380987654323	Middle teach	1	250

Рисунок Г.26 – Звіт про роботу викладачів

Ім'я	Прізвище	Вік	Дата народження	Телефон	Назва групи	Середня оцінка
Анна	Литвиненко	12	15.03.2012 0:00:00	+380987651111	СБ14:00 Arduino Kids Pro	91
Олег	Литвиненко	15	20.06.2009 0:00:00	+380987652222	СБ14:00 Arduino Kids Pro	95
Павло	Бондар	7	03.12.2014 0:00:00	+380987660000	СБ12:00 Arduino Kids	88

Рисунок Г.27 – Звіт по успішності учнів

Звіт по курсам

Назва курсу	Ціна курсу	Назва напрямку	Сума продажів
Arduino Junior	1200	Робототехніка та програмування	0
Arduino Kids	1200	Робототехніка та програмування	7200
Arduino KidsPro	1400	Робототехніка та програмування	4200
Construct	1200	Розробка ігор	3600
Expert Embedded	1500	Робототехніка та програмування	0
Java Beginner	1200	Розробка програмного забезпечення	0
Java Junior	1400	Розробка програмного забезпечення	0
Java Middle	1500	Розробка програмного забезпечення	0
Middle Embedded	1450	Робототехніка та програмування	0
Pro Embedded	1500	Робототехніка та програмування	0
Python Beginner	1200	Програмування Python	0
Senior Embedded	1500	Робототехніка та програмування	0
Unity 2D	1400	Розробка ігор	0

Рисунок Г.28 – Звіт з продажів курсів

Звіт по групам

Назва групи	Назва напрямку	Назва курсу	Кількість слухачів
НД: 12:00 Arduino Kids	Робототехніка та програмування	Arduino Kids	3
НД: 14:00 Arduino Kids Pro	Робототехніка та програмування	Arduino KidsPro	1
НД: 16:00 Unity Beginner	Розробка ігор	Unity Beginner	4
СБ 10:00 Arduino Kids	Робототехніка та програмування	Arduino Kids	4
СБ 12:00 Arduino Kids	Робототехніка та програмування	Arduino Kids	2
СБ 14:00 Arduino Kids Pro	Робототехніка та програмування	Arduino KidsPro	2
СБ 16:00 Construct	Розробка ігор	Construct	3
СБ 18:00 Web Beginner	Розробка сайтів	Web Beginner	3

Рисунок Г.29 – Звіт по групам

**Додаток Д. Частина коду програми*****Код процедури форми завантаження***

```
public Form_loading()
{
    InitializeComponent();
    timer1.Start();
}
private void timer1_Tick(object sender, EventArgs e)
{
    progressBar1.Increment(1);
    int value = progressBar1.Value;
    label_loading.Text = "Завантаження... " + value + "%";
    label_loading.Visible = true;
    if (progressBar1.Value == 100)
    {
        timer1.Stop();
        label_loading.Text = "Завантаження... " + value + "% Завантажено";
        form_authorization.Show();
        this.Hide();
    }
}
```

***Код процедури авторизації:***

```
private string connectionString = "Data Source=DESKTOP-ANDREY;Initial
Catalog=Users_Robocode;Persist Security Info=True;User ID=sa;Password=sa";

private void button1_Click(object sender, EventArgs e)
{
    string login = textBox_login.Text;
    string password = textBox_password.Text;
```

```

string query = "SELECT Адмін FROM Користувачі WHERE Логін =
@Login AND Пароль = @Password";
using (SqlConnection connection = new SqlConnection(connectionString))
{
    SqlCommand command = new SqlCommand(query, connection);
    command.Parameters.AddWithValue("@Login", login);
    command.Parameters.AddWithValue("@Password", password);
    try
    {
        connection.Open();
        object result = command.ExecuteScalar();
        if (result != null && result != DBNull.Value && (bool)result)
        {
            Form_main form_main = new Form_main();
            form_main.FormClosed += Form_main_FormClosed;
            form_main.Show();
            this.Hide();
        }
        else
        {
            MessageBox.Show("Неправильний логін або пароль!");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Помилка під час авторизації: " + ex.Message);
    }
}
private void Form_main_FormClosed(object sender, FormClosedEventArgs e)

```

```

{
    this.Close();
}
private void Form_Authorization_FormClosed(object sender,
FormClosedEventArgs e)
{
    Application.Exit();
}

```

***Код для забезпечення навігації:***

```

private void вчительToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form_teacher form_teacher = new Form_teacher();
    form_teacher.ShowDialog();
}
private void слухачToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form_student form_student = new Form_student();
    form_student.ShowDialog();
}
private void напрямкиТаКурсиToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Form_courses form__courses = new Form_courses();
    form__courses.ShowDialog();
}
private void групаToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form_group form_group = new Form_group();
    form_group.ShowDialog();
}
private void оцінкаToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form_rating form_rating = new Form_rating();
    form_rating.ShowDialog();
}
private void оплатаToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form_pay form_pay = new Form_pay();
    form_pay.ShowDialog();
}

```

```

}
private void курсиToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form_direction form_direction = new Form_direction();
    form_direction.ShowDialog();
}
private void заняттяToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form_lesson form_lesson = new Form_lesson();
    form_lesson.ShowDialog();
}
private void заробітнаПлатаЗаУрокиToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Form_req_data_teacher form_req_data_teacher = new
Form_req_data_teacher();
    form_req_data_teacher.ShowDialog();
}
private void toolStripMenuItem1_Click(object sender, EventArgs e)
{
    Form_schedule form_schedule = new Form_schedule();
    form_schedule.ShowDialog();
}
private void даніПоГрупамToolStripMenuItem_Click(object sender, EventArgs
e)
{
    Form_req_groups form_req_groups = new Form_req_groups();
    form_req_groups.ShowDialog();
}
private void даніПроСлухачаToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Form_req_student form_req_student = new Form_req_student();
    form_req_student.ShowDialog();
}
private void даніПроРозкладГрупToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Form_req_schedule form_req_schedule = new Form_req_schedule();
    form_req_schedule.ShowDialog();
}
private void успішністьУчнівToolStripMenuItem_Click(object sender,
EventArgs e)
{

```

```

    Form_report_teachers form_report_teachers = new Form_report_teachers();
    form_report_teachers.ShowDialog();
}
private void toolStripMenuItem1_Click_1(object sender, EventArgs e)
{
    Form_report_groups form_report_groups = new Form_report_groups();
    form_report_groups.ShowDialog();
}
private void Form_main_Load(object sender, EventArgs e)
{
}
private void Form_main_FormClosed(object sender, FormClosedEventArgs e)
{
    Application.Exit();
}
private void даніПроКурсиToolStripMenuItem_Click(object sender, EventArgs
e)
{
    Form_req_courses form_req_courses = new Form_req_courses();
    form_req_courses.Show();
}
private void даніПроЗаняттяToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Form_req_lesson form_req_lesson = new Form_req_lesson();
    form_req_lesson.Show();
}
private void звітПоСлухачахToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Form_report_student form_report_student = new Form_report_student();
    form_report_student.Show();
}
private void звітПоКурсамToolStripMenuItem_Click(object sender, EventArgs
e)
{
    Form_report_course form_report_course = new Form_report_course();
    form_report_course.Show();
}
private void звітПоЗаняттяхToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Form_report_lesson form_report_lesson = new Form_report_lesson();

```

```

    form_report_lesson.Show();
}

```

***Код функції пошуку по прізвищу викладача:***

```

private int lastFoundIndex = -1;
private void button_searchTeacher_Click(object sender, EventArgs e)
{
    string searchText = textBox_searchTeacher.Text.Trim();

    if (string.IsNullOrEmpty(searchText))
    {
        MessageBox.Show("Будь ласка, введіть прізвище для пошуку.");
        return;
    }
    int startIndex = lastFoundIndex + 1;
    if (startIndex >= вчительBindingSource.Count)
    {
        startIndex = 0;
        lastFoundIndex = -1;
    }
    int nextFoundIndex = -1;

    for (int i = startIndex; i < вчительBindingSource.Count; i++)
    {
        DataRowView rowView = (DataRowView)вчительBindingSource[i];
        string surname = rowView["Прізвище"].ToString().Trim();

        if (surname.IndexOf(searchText, StringComparison.OrdinalIgnoreCase) >=
0)
        {
            nextFoundIndex = i;
            break;
        }
    }
    if (nextFoundIndex != -1)
    {
        вчительBindingSource.Position = nextFoundIndex;
        lastFoundIndex = nextFoundIndex;
    }
    else
    {
        MessageBox.Show("Вчителів з введеним прізвищем не знайдено.");
        lastFoundIndex = -1;
    }
}

```

```
}
```

***Код функції пошуку по прізвищу викладача:***

```
private int lastFoundIndex = -1;

private void button_SearchStudent_Click(object sender, EventArgs e)
{
    string searchText = textBox_searchStudent.Text.Trim();
    if (string.IsNullOrEmpty(searchText))
    {
        MessageBox.Show("Будь ласка, введіть прізвище для пошуку.");
        return;
    }
    int startIndex = lastFoundIndex + 1;
    if (startIndex >= слухачBindingSource.Count)
    {
        startIndex = 0;
        lastFoundIndex = -1;
    }
    int nextFoundIndex = -1;
    for (int i = startIndex; i < слухачBindingSource.Count; i++)
    {
        DataRowView rowView = (DataRowView)слухачBindingSource[i];
        string surname = rowView["Прізвище"].ToString().Trim();
        if (surname.IndexOf(searchText, StringComparison.OrdinalIgnoreCase) >=
0)
        {
            nextFoundIndex = i;
            break;
        }
    }
    if (nextFoundIndex != -1)
```

```

    {
        слухачBindingSource.Position = nextFoundIndex;
        lastFoundIndex = nextFoundIndex;
    }
else
    {
        MessageBox.Show("Слухачів з введеним прізвищем не знайдено.");
        lastFoundIndex = -1;
    }
}

```

***Код функції фільтрування по назві курсу:***

```

private void radioButton1_CheckedChanged(object sender, EventArgs e)
    {
        this.курсBindingSource.Filter = $"Назва_курсу = '{textBox_filtr_nameCourse.Text}'";
    }
private void radioButton_filtrOff_CheckedChanged(object sender, EventArgs e)
    {
        this.курсBindingSource.RemoveFilter();
    }

```

***Код функції пошуку по назві напрямку:***

```

private void button_SearchDirection_Click(object sender, EventArgs e)
    {
        int itemDirection = this.напрямокBindingSource.Find("Назва_напрямку",
textBox_searchDirection.Text);
        if (itemDirection != -1)
            {
                напрямокBindingSource.Position = itemDirection;
            }
else

```

```

    {
        MessageBox.Show("Напрямок з введеною назвою не знайдено.");
    }
}

```

***Код функції пошуку по назві групи:***

```

private void button_SearchGroup_Click(object sender, EventArgs e)
{
    int    itemGroup    =    this.групаBindingSource.Find("Назва_групи",
textBox_searchGroup.Text);
    if (itemGroup != -1)
    {
        групаBindingSource.Position = itemGroup;
    }
    else
    {
        MessageBox.Show("Групу з введеною назвою не знайдено.");
    }
}

```

***Код функції пошуку заняття по даті:***

```

private int lastFoundIndex = -1;

private void button_SearchLesson_Click(object sender, EventArgs e)
{
    string searchText = textBox_searchLesson.Text.Trim();
    if (string.IsNullOrEmpty(searchText))
    {
        MessageBox.Show("Будь ласка, введіть дату для пошуку.");
        return;
    }
    if (!DateTime.TryParse(searchText, out DateTime searchDate))
    {

```

```
        MessageBox.Show("Будь ласка, введіть дату у правильному форматі  
(yyyy-MM-dd).");  
        return;  
    }  
    int startIndex = lastFoundIndex + 1;  
    if (startIndex >= заняттяBindingSource.Count)  
    {  
        startIndex = 0;  
        lastFoundIndex = -1;  
    }  
    int nextFoundIndex = -1;  
    for (int i = startIndex; i < заняттяBindingSource.Count; i++)  
    {  
        DataRowView rowView = (DataRowView)заняттяBindingSource[i];  
        DateTime date = Convert.ToDateTime(rowView["Дата_заняття"]);  
        if (date.Date == searchDate.Date)  
        {  
            nextFoundIndex = i;  
            break;  
        }  
    }  
    if (nextFoundIndex != -1)  
    {  
        заняттяBindingSource.Position = nextFoundIndex;  
        lastFoundIndex = nextFoundIndex;  
    }  
    else  
    {  
        MessageBox.Show("Записів з введеною датою заняття не знайдено.");  
        lastFoundIndex = -1;  
    }  
}
```

```

    }
}

```

***Код функції пошуку чеку оплати по даті оплати:***

```

private int lastFoundIndex = -1;

private void button_SearchPay_Click(object sender, EventArgs e)
{
    string searchText = textBox_searchPay.Text.Trim();
    if (string.IsNullOrEmpty(searchText))
    {
        MessageBox.Show("Будь ласка, введіть дату для пошуку.");
        return;
    }
    if (!DateTime.TryParse(searchText, out DateTime searchDate))
    {
        MessageBox.Show("Будь ласка, введіть дату у правильному форматі (yyyy-MM-dd).");
        return;
    }
    int startIndex = lastFoundIndex + 1;
    if (startIndex >= чек_з_оплатоюBindingSource.Count)
    {
        startIndex = 0;
        lastFoundIndex = -1;
    }
    int nextFoundIndex = -1;
    for (int i = startIndex; i < чек_з_оплатоюBindingSource.Count; i++)
    {
        DataRowView                rowView                =
(DataRowView)чек_з_оплатоюBindingSource[i];
        DateTime date = Convert.ToDateTime(rowView["Дата_оплати"]);

```

```

if (date.Date == searchData.Date)
{
    nextFoundIndex = i;
    break;
}
}
if (nextFoundIndex != -1)
{
    чек_з_оплатоюBindingSource.Position = nextFoundIndex;
    lastFoundIndex = nextFoundIndex;
}
else
{
    MessageBox.Show("Записів з введеною датою оплати не знайдено.");
    lastFoundIndex = -1;
}
}

```

***Код функції пошуку слухача по прізвищу через параметричний запит:***

```

private void reqStudentButton_Click(object sender, EventArgs e)
{
    try
    {
        this.запит_параметр_слухачTableAdapter.Fill(this._IT_школаDataSet.запит_параметр_слухач, comboBox_reqStudent.SelectedValue.ToString());
    }
    catch(System.Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
}

```

***Код функції пошуку розкладу для групи через параметричний запит:***

```
private void reqScheduleButton_Click(object sender, EventArgs e)
{
    try
    {
        this.запит_параметр_розкладTableAdapter.Fill(this._IT_школаDataSet.Запит_параметр_розклад, comboBox_reqSchedule.SelectedValue.ToString());
    }
    catch (System.Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
}
```

***Код налаштування валідації:***

```
private void групаBindingNavigator_Click_1(object sender, EventArgs e)
{
    this.Validate();
    this.групаBindingSource.EndEdit();
    try
    {
        this.tableAdapterManager.UpdateAll(this._IT_школаDataSet);
    }
    catch (SqlException ex)
    {
        MessageBox.Show("Помилка: " + ex.Message, "Помилка");
    }
}
```