

**ВІННИЦЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ МИХАЙЛА КОЦЮБИНСЬКОГО**

**ФАКУЛЬТЕТ МАТЕМАТИКИ, ФІЗИКИ І КОМП'ЮТЕРНИХ НАУК**

**Кафедра математики та інформатики**

**КУРСОВА РОБОТА**

з дисципліни «Бази даних та інформаційні системи»

на тему: **«Проектування та розробка інформаційної системи електронного журналу для обліку успішності та відвідуваності учнів середньої школи»**

Студента 3 курсу групи СОІ

Освітньої програми Середня освіта. Інформатика, математика

Предметної спеціальності 014.09 Середня освіта (Інформатика)

Галузі знань 01 Освіта / Педагогіка

Ступеня вищої освіти бакалавра

\_\_\_\_\_ Глухоти Максима Анатолійовича \_\_\_\_\_  
(прізвище, ім'я, по батькові в родовому відмінку)

Керівник \_\_\_\_\_ Ключко Оксана Віталіївна \_\_\_\_\_  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Оцінка за розширеною шкалою \_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_

Голова комісії \_\_\_\_\_ О.В.Ключко  
(підпис)

Члени комісії \_\_\_\_\_ О.П.Косовець  
(підпис)

\_\_\_\_\_ Я.В.Крупський  
(підпис)

## ЗМІСТ

<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ .....</b>	<b>3</b>
<b>ВСТУП .....</b>	<b>4</b>
<b>РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ В ОСВІТІ .....</b>	<b>7</b>
1.1. Поняття та класифікація інформаційних систем в освіті .....	7
1.2. Огляд наукових джерел і досліджень щодо електронних журналів .....	8
1.3. Аналіз існуючих інформаційних систем обліку успішності та відвідуваності .....	9
<b>ВИСНОВКИ ДО РОЗДІЛУ 1 .....</b>	<b>11</b>
<b>РОЗДІЛ 2. МЕТОДОЛОГІЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ .....</b>	<b>12</b>
2.1. Характеристика предметної області та аналіз інформаційних потоків .....	12
2.2. Обґрунтування вибору засобів розробки та СУБД .....	13
2.3. Проєктування інфологічної моделі бази даних .....	16
2.4. UML-діаграма послідовності роботи з електронним журналом .....	19
<b>ВИСНОВКИ ДО РОЗДІЛУ 2 .....</b>	<b>21</b>
<b>РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ «ЕЛЕКТРОННИЙ ЖУРНАЛ» .....</b>	<b>22</b>
3.1. Обґрунтування вибору засобів розробки .....	22
3.2. Проєктування структури бази даних .....	23
3.3. Реалізація SQL-запитів для основної логіки .....	24
3.4. Програмна реалізація інтерфейсу користувача та бізнес-логіки системи .....	26
<b>ВИСНОВКИ ДО РОЗДІЛУ 3 .....</b>	<b>35</b>
<b>ВИСНОВКИ .....</b>	<b>36</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>38</b>
<b>ДОДАТОК А .....</b>	<b>41</b>
<b>ДОДАТОК Б .....</b>	<b>42</b>
<b>ДОДАТОК В .....</b>	<b>45</b>
<b>ДОДАТОК Г .....</b>	<b>46</b>

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ**

<b>АРМ</b>	Автоматизоване робоче місце
<b>БД</b>	База даних
<b>ІС</b>	Інформаційна система
<b>МОН</b>	Міністерство освіти і науки України
<b>НУШ</b>	Нова українська школа
<b>СУБД</b>	Система управління базами даних
<b>MVVM</b>	Model-View-ViewModel (патерн проєктування)
<b>SQL</b>	Structured Query Language (Мова структурованих запитів)
<b>UML</b>	Unified Modeling Language (Уніфікована мова моделювання)
<b>WPF</b>	Windows Presentation Foundation

## ВСТУП

**Актуальність дослідження.** Стрімка цифровізація освітнього простору України зумовлює докорінний перегляд підходів до управління навчальним процесом у закладах загальної середньої освіти. Сучасна парадигма розвитку освіти, закріплена у Законі України «Про повну загальну середню освіту» [18] та наукових працях провідних учених, висуває нові вимоги до оперативності збору, обробки та аналізу даних про успішність учнів. Традиційні паперові форми обліку поступово втрачають актуальність, поступаючись місцем автоматизованим системам, впровадження яких регламентується «Інструкцією з ведення ділової документації в електронній формі» (Наказ Міністерством освіти і науки України (далі – МОН) № 707) [7]. Актуальність переходу на цифрові носії підтверджується тим, що ведення журналів у цифровому форматі значно спрощує роботу педагогів, дає змогу зручно заповнювати їх у будь-який час та з будь-якого місця, водночас захищаючи дані від стороннього доступу [5].

Аналіз ринку свідчить про домінування хмарних екосистем (наприклад, «Мрія» [13]), проте ризики залежності від інтернет-з'єднання та зовнішніх провайдерів актуалізують потребу в гібридних або локальних архітектурах для гарантування безперервності навчального процесу. Водночас, в умовах зростання кіберзагроз та можливих перебоїв у роботі глобальних мереж, критичного значення набуває питання інформаційної безпеки та автономності освітніх даних. Як зазначають дослідники [2; 35], повна залежність від хмарних провайдерів створює ризики втрати доступу до академічних записів у надзвичайних ситуаціях. Це актуалізує потребу у створенні гібридних або локальних рішень (Intranet), які гарантують безперервність навчального процесу та збереження даних всередині захищеного периметра закладу освіти, незалежно від зовнішніх факторів.

**Мета і завдання дослідження.** Метою роботи є проектування та розробка інформаційної системи електронного журналу для автоматизації обліку успішності та відвідуваності учнів середньої школи засобами технологій .NET.

Досягнення сформульованої мети передбачає вирішення таких завдань:

1. Провести системний аналіз предметної області, нормативно-правової бази та існуючих програмних аналогів автоматизації шкільного обліку.

2. Спроекувати інфологічну модель Entity–Relationship diagram (далі – ER-діаграму) та динамічну модель взаємодії компонентів системи Unified Modeling Language (далі – UML).

3. Реалізувати фізичну структуру бази даних та набір оптимізованих Structured Query Language (далі – SQL) -запитів у середовищі MS SQL Server.

4. Розробити графічний інтерфейс та бізнес-логіку клієнтського застосунку (АРМ вчителя) мовою C# з використанням технології Windows Presentation Foundation (далі – WPF) та патерну Model-View-ViewModel (далі – MVVM).

**Об’єкт дослідження** – процес інформаційного забезпечення та автоматизації обліку навчальних досягнень учнів.

**Предмет дослідження** – методи та засоби проєктування інформаційної системи електронного журналу на основі реляційних баз даних.

**Матеріал дослідження.** Інформаційною базою слугували нормативні документи МОН України [7], теоретичні положення проєктування баз даних [16] та документація платформи .NET [31; 39].

**Методи дослідження.** У роботі використано комплекс методів: теоретичні – аналіз наукової літератури [2; 21] та нормативних документів для визначення вимог до системи; емпіричні – моделювання предметної області (ER-діаграми) для проєктування структури даних; практичні – об’єктно-орієнтоване програмування [37] для реалізації функціоналу системи.

Наукова новизна одержаних результатів полягає у подальшому розвитку методів проєктування інформаційних систем освітнього моніторингу шляхом розробки адаптивної моделі подання даних. На відміну від типових рішень, запропонований підхід дозволяє динамічно формувати структуру електронного журналу відповідно до вимог оцінювання компетентностей НУШ (групи результатів), забезпечуючи при цьому цілісність даних в умовах автономного функціонування локальної мережі закладу освіти.

Практичне значення одержаних результатів визначається створенням повнофункціонального програмного продукту — десктопного застосунку (файл .exe), реалізованого на основі клієнт-серверної архітектури з використанням технологій MS SQL Server та WPF.

Розроблена система готова до впровадження у закладах середньої освіти для автоматизації робочого місця вчителя. Запропоноване рішення дозволяє автоматизувати рутинні процеси, підвищити прозорість оцінювання, а також гарантувати автономність роботи електронного журналу, мінімізуючи залежність навчального процесу від стабільності інтернет-з'єднання.

Крім того, отримані результати, зокрема спроектована архітектура бази даних та набір оптимізованих SQL-запитів, мають універсальний характер і можуть слугувати основою для подальших наукових розробок студентів, наприклад, для створення веб-версії «Кабінету учня» або інтеграції з іншими освітніми платформами.

**Структура та обсяг роботи.** Курсова робота складається зі вступу, трьох розділів, висновків, списку використаних джерел (40 найменувань) та додатків. Загальний обсяг роботи становить 46 сторінок. Оформлення пояснювальної записки виконано згідно з вимогами ДСТУ 3008:2015 [4].

## **РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ В ОСВІТІ**

### **1.1 Поняття та класифікація інформаційних систем в освіті**

Сучасний етап розвитку освітньої галузі нерозривно пов'язаний із процесами глобальної цифровізації. Як зазначено у Концепції «Нова українська школа» [10] та фахових виданнях [12], проникнення інформаційно-комунікаційних технологій у навчальний процес та адміністративну діяльність закладів освіти є необхідною умовою підвищення якості освіти. Це зумовлює необхідність впровадження спеціалізованих інформаційних систем (далі – ІС), що регламентується Законом України «Про освіту» [17] та Державною програмою «Освіта» [3].

В загальному розумінні, інформаційна система — це взаємопов'язана сукупність засобів, методів і персоналу, що використовується для зберігання, обробки та видачі інформації з метою вирішення завдань управління [1; 2]. При переході до освітньої сфери, інформаційна система в освіті (далі – ІСО) адаптує цю концепцію для вирішення специфічних педагогічних та адміністративних завдань. Головною метою ІСО є підвищення ефективності управління освітнім процесом та надання доступу до інформаційних ресурсів усім його учасникам: адміністрації, вчителям, учням та батькам [5].

Існуючі ІСО можна класифікувати за їх функціональним призначенням:

1) Системи управління навчанням (Learning Management Systems (далі – LMS)). Це платформи, орієнтовані безпосередньо на дидактичний процес. Вони надають інструменти для розміщення навчальних матеріалів, онлайн-тестування, комунікації між викладачем та учнем. Найвідомішими прикладами є Moodle [29] та Google Classroom [26].

2) Студентські інформаційні системи (Student Information Systems, SIS). Цей клас систем забезпечує облік даних про учнів, їх академічну успішність та відвідуваність. До цієї категорії належать електронні журнали та щоденники, такі як «Нові знання» [14] та державна система «Мрія» [13].

3) Системи управління закладом освіти (School Management Systems (далі – SMS)). Це переважно адміністративні системи, покликані автоматизувати

управлінські процеси школи. До їхніх функцій належить ведення баз даних особових справ учнів та персоналу, управління розкладом занять, фінансовий облік та ведення шкільної документації [35].

4) Спеціалізовані інформаційні системи. Це вузькоспрямовані рішення для конкретних завдань, наприклад: ІС «Бібліотека», ІС «Тестування знань», ІС «Облік приміщень університету» або системи формування розкладу [9].

Електронний журнал успішності та відвідуваності, що є предметом даного дослідження, займає проміжне положення. Він є ключовим модулем як для систем управління навчанням (оскільки фіксує результати), так і для систем адміністративного управління (оскільки є основним інструментом звітності та контролю).

## **1.2 Огляд наукових джерел і досліджень щодо електронних журналів**

Питання цифровізації шкільної документації розглядається як у нормативно-правовій площині, так і в наукових дослідженнях.

Згідно з нормативною базою, електронний класний журнал — це окремий електронний документ ділової документації закладу освіти, у якому фіксують результати навчальних досягнень учнів, їхнє відвідування та виконання навчальних програм. Він може функціонувати як самостійна електронна освітня інформаційна система (далі – ОІС) або як її модуль [5]. Можливість вести електронні журнали замість паперових передбачено наказом МОН № 707, яким затверджено «Інструкцію з ведення ділової документації в закладах загальної середньої освіти в електронній формі» [7].

Порядок діловодства в закладі загальної середньої освіти визначається керівником відповідно до законодавства про освіту. У межах своєї автономії педагогічна рада може ухвалити рішення про вибір тієї чи іншої ОІС [5]. Це створює правове підґрунтя для розробки та впровадження авторських програмних рішень у конкретному закладі освіти.

Аналіз наукових джерел дозволяє виділити кілька ключових напрямів досліджень цієї проблематики:



1) Педагогічні аспекти. Дослідники Н. В. Морзе [12] та О. В. Співаковський [19] аналізують вплив електронних журналів на підвищення об'єктивності та прозорості оцінювання. Встановлено, що відкритість даних про успішність позитивно впливає на мотивацію учнів та ефективність комунікації у триаді «вчитель—учень—батьки».

2) Технологічні аспекти. Значна увага приділяється технічній реалізації ІС. Автори розглядають переваги хмарних технологій [20], методи проєктування реляційних баз даних [15], а також питання захисту персональних даних.

3) Ергономічні аспекти. Науковці підкреслюють, що успіх системи залежить не лише від функціоналу, але й від зручності інтерфейсу (User Interface (далі – UI/UX)) [22] та мінімізації часу, який вчитель витрачає на внесення даних.

Попри широкий спектр досліджень, більшість робіт фокусується на використанні готових веб-платформ. Ніша розробки локальних клієнт-серверних рішень, які б поєднували надійність десктопних застосунків (WPF) та потужність професійних систем управління базами даних (далі – СУБД) (MS SQL Server), залишається недостатньо висвітленою, що підтверджує актуальність теми курсової роботи.

### **1.3 Аналіз існуючих інформаційних систем обліку успішності та відвідуваності**

Для визначення функціональних вимог до проєктованої системи було проведено аналіз ринку програмного забезпечення. На сьогодні на edtech-ринку України представлено понад 15 приватних рішень — як платних, так і безоплатних [5]. Кожен заклад визначає самостійно, яку ОІС використовувати.

Розглянемо найпоширеніші системи:

1) «Нові Знання» (nz.ua). Одна з наймасовіших всеукраїнських систем [14]. Перевагою є комплексність та єдиний простір для всіх учасників. Недоліком є жорстка структура, перевантаженість інтерфейсу та повна залежність від інтернет-з'єднання.

2) «HUMAN». Позиціонується як освітня соціальна мережа [30]. Має сучасний дизайн, але орієнтованість на SaaS-модель (плата за підписку) та складність функціоналу може бути перешкодою для впровадження у невеликих школах.

3) «E-Schools» («Єдина школа»). Комплексна система управління, що охоплює адміністративні та фінансові задачі [6]. Широкий функціонал системи, що охоплює фінансові та адміністративні модулі, може бути надлишковим для вчителя, якому необхідний лише журнал успішності.

Важливим аспектом є інтеграція. ОІС, що відповідають правилам безпеки, можуть під'єднуватися до системи Автоматизований інформаційний комплекс освітнього менеджменту (далі – АІКОМ), що дозволяє значно спростити звітність [5]. Як безоплатна державна альтернатива доступна цифрова екосистема «Мрія» [13], до якої під'єднано вже понад 2000 закладів освіти. Це ініціатива Президента України, яку реалізують Мінцифра та МОН. Вона доступна у формі мобільного застосунку та вебпорталу, забезпечуючи зручний цифровий простір для взаємодії між школою і родинами.

Підсумок аналізу: Більшість існуючих рішень — це хмарні веб-платформи, які ставлять заклад у залежність від провайдера та наявності Інтернету. Це обґрунтовує доцільність розробки альтернативного рішення — спеціалізованої клієнт-серверної системи (АРМ Вчителя), що функціонує локально, забезпечує автономність даних та високу швидкодію [20].

## ВИСНОВКИ ДО РОЗДІЛУ 1

У першому розділі було розглянуто теоретичні основи розробки інформаційних систем в освіті, що дозволило сформулювати цілісне уявлення про їх сутність, класифікацію та сучасний стан розвитку. Встановлено, що інформаційні системи є ключовим елементом цифрової трансформації освітнього процесу, оскільки забезпечують збір, обробку, зберігання та передачу даних, необхідних для управління навчанням і прийняття рішень. Еволюція систем від паперових форм до сучасних хмарних сервісів підтверджує закономірність переходу освіти на цифрові технології, що відповідає концепції Нової української школи.

Аналіз наукових джерел та нормативно-правової бази показав, що електронні журнали розглядаються не лише як інструмент обліку успішності та відвідуваності, але й як засіб підвищення прозорості оцінювання та ефективності комунікації. Дослідження акцентують увагу на педагогічних, технологічних та ергономічних аспектах. Водночас встановлено, що чинне законодавство (зокрема Наказ МОН № 707) надає закладам освіти автономію у виборі програмних засобів, що створює правове поле для впровадження власних розробок.

Аналіз існуючих інформаційних систем (таких як «Нові знання», «HUMAN», «Єдина школа» та державна екосистема «Мрія») показав, що переважна більшість рішень є хмарними та залежними від стабільності інтернет-з'єднання. Вони забезпечують широкий функціонал і доступність, проте створюють ризики для закладів із нестабільним доступом до мережі. Потреба у локальних клієнт-серверних рішеннях, які б гарантували автономність роботи, високу швидкість та захист даних усередині периметра закладу, залишається актуальною.

Таким чином, результати першого розділу підтверджують, що існуючі на ринку системи мають як переваги, так і суттєві обмеження. Це створює обґрунтоване підґрунтя для розробки власної інформаційної системи на базі технологій .NET та MS SQL Server, яка поєднає переваги автоматизації із вимогами надійності та автономності.

## РОЗДІЛ 2 МЕТОДОЛОГІЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 2.1 Характеристика предметної області та аналіз інформаційних потоків

Проектування будь-якої інформаційної системи розпочинається із системного аналізу предметної області. У контексті даної роботи розглядається процес обліку успішності та відвідуваності учнів у закладі загальної середньої освіти. Згідно з принципами системного підходу [8; 21], школа розглядається як складна організаційна система, де циркулюють інформаційні потоки між учасниками освітнього процесу.

На сьогодні у більшості шкіл, які не перейшли на цифрові платформи, облік ведеться у паперовому вигляді (модель «AS-IS» — «як є»). Цей процес має низку суттєвих недоліків: дублювання інформації у класному журналі, щоденниках та табелях; висока ймовірність механічних помилок при розрахунку середнього балу; складність формування аналітичних звітів для адміністрації; відсутність оперативного доступу батьків до оцінок. Метою розробки нової системи (модель «TO-BE» — «як має бути») є автоматизація цих рутинних операцій.

Аналіз бізнес-процесів дозволив класифікувати основні інформаційні потоки системи:

- 1) Навчальні дані: оцінки, теми уроків, домашні завдання (таблиці Lessons, Grades у базі даних).
- 2) Адміністративні дані: списки учнів, навантаження вчителів, кадрова інформація (Students, TeachingAssignments).
- 3) Структурні дані: розклад занять, перелік предметів та класів (FixedSchedule, Subjects).

На основі аналізу нормативних вимог [7] виділено ключові сутності предметної області, необхідні для побудови реляційної моделі даних:

- 1) Суб'єкти освітнього процесу:

- Вчителі: Ключові користувачі системи. Кожен вчитель характеризується персональними даними, спеціалізацією та кваліфікаційною категорією. Вчитель також може виконувати роль класного керівника.

- Учні: Об'єкти обліку. Для кожного учня зберігаються персональні дані, інформація про батьків (для комунікації) та приналежність до класу.

- Класи: Організаційні одиниці, що групують учнів та мають закріпленого класного керівника.

## 2) Об'єкти обліку:

- Навчальне навантаження: Розподіл предметів між вчителями у конкретних класах (хто, де і що викладає).

- Уроки: Динамічні події, що мають дату, тему, тип (контрольна, самостійна тощо) та домашнє завдання.

- Результати навчання: Оцінки успішності та відмітки про відвідуваність, які мають бути суворо прив'язані до конкретного учня та уроку.

Специфіка предметної області вимагає суворого дотримання цілісності даних: оцінка не може існувати без прив'язки до учня та уроку, а урок повинен відповідати затвердженому навчальному навантаженню. Це висуває підвищені вимоги до проектування схеми бази даних, яка має відповідати третій нормальній формі (далі – 3НФ) для уникнення надлишковості та аномалій при модифікації даних [27].

## 2.2 Обґрунтування вибору засобів розробки та СУБД

Вибір технологічного стеку є критичним етапом проектування, що визначає подальшу ефективність та масштабованість системи [16]. Для реалізації поставленої мети було проведено порівняльний аналіз сучасних інструментальних засобів.

Для реалізації підсистеми збереження даних було проведено порівняльний аналіз чотирьох найпоширеніших реляційних СУБД: SQLite, Microsoft Access, MySQL та Microsoft SQL Server. Вибір здійснювався на основі критеріїв сумісності з платформою .NET, підтримки мережевого доступу та можливостей забезпечення цілісності даних. Результати аналізу наведено нижче:

- SQLite: Компактна вбудована база даних. Вона ідеальна для мобільних додатків, але не підтримує повноцінний мережевий багатокористувацький режим.

Оскільки система «Електронний журнал» передбачає потенційну можливість одночасної роботи кількох вчителів з різних комп'ютерів, файлова архітектура SQLite може призвести до блокувань файлу бази даних та втрати інформації при одночасному запису.

- Microsoft Access: Популярна настільна СУБД, що входить до пакету MS Office. Її перевагою є простота створення баз даних початкового рівня. Проте, Access використовує файл-серверну архітектуру, що створює значне навантаження на локальну мережу при збільшенні кількості користувачів. Крім того, Access має обмежені можливості щодо захисту даних та не підтримує розширені можливості мови SQL (зокрема, тригери та складні збережувані процедури), які необхідні для реалізації бізнес-логіки журналу [16].

- MySQL: Популярна безкоштовна СУБД, яка є стандартом для веб-розробки (стек LAMP). Вона є потужною та швидкою, проте її інтеграція з технологіями Microsoft .NET потребує встановлення додаткових драйверів (MySQL Connector) і не є нативною. Це ускладнює розгортання програми на шкільних комп'ютерах, де права адміністратора можуть бути обмежені.

- Microsoft SQL Server: Потужна об'єктно-реляційна система керування базами даних клієнт-серверного типу.

Для реалізації курсової роботи було обрано саме цю СУБД з огляду на такі фактори:

1) Глибока інтеграція з .NET: Використання бібліотеки System.Data.SqlClient забезпечує найвищу продуктивність та стабільність з'єднання в середовищі розробки Visual Studio [31].

2) Підтримка T-SQL: Можливість використання діалекту Transact-SQL дозволяє перенести частину обчислювальної логіки (наприклад, розрахунок середнього балу або валідацію даних) безпосередньо на сервер за допомогою збережуваних процедур та функцій, що відповідає сучасним стандартам проектування ІС [24].

3) Надійність та безпека: Підтримка ACID-транзакцій гарантує, що оцінки не будуть втрачені або записані частково у випадку збою електроживлення чи розриву з'єднання [40].

4) Масштабованість: Використання редакції LocalDB або Express дозволяє розгорнути систему безкоштовно на звичайному ПК вчителя, але при необхідності базу даних можна легко перенести на потужний сервер Enterprise без зміни програмного коду.

Обґрунтування вибору засобів розробки клієнтської частини. Для створення інтерфейсу користувача (АРМ вчителя) обрано мову програмування C# та технологію WPF.

1) Мова C#: Це сучасна об'єктно-орієнтована мова зі строгою типізацією, яка є основним інструментом розробки у екосистемі Microsoft. Використання C# дозволяє застосовувати передові практики програмування, такі як Language Integrated Query (далі – LINQ) (для роботи з колекціями даних) та асинхронне програмування (async/await), що підвищує чутливість інтерфейсу [23; 37].

2) Технологія WPF: На відміну від застарілої технології Windows Forms, яка використовує піксельну графіку, WPF базується на векторній графіці та використовує мову розмітки Extensible Application Markup Language (далі – XAML). Це дозволяє:

- Розділити дизайн (XAML) та логіку програми (C# code-behind).
- Забезпечити коректне відображення інтерфейсу на екранах з будь-якою роздільною здатністю (що важливо для шкільних моніторів різної якості).
- Використовувати прив'язку даних (Data Binding), що значно спрощує відображення таблиць з оцінками [28; 33].

Архітектурний патерн MVVM : Використання шаблону MVVM є стандартом де-факто для WPF-додатків. Він дозволяє відокремити графічний інтерфейс (View) від бізнес-логіки (Model) через проміжний шар (ViewModel). Це спрощує тестування програми та дозволяє легко змінювати дизайн без переписування коду логіки [22; 36].

## 2.3 Проектування інфологічної моделі бази даних

Інфологічна модель — це формалізоване представлення предметної області у вигляді логічної структури даних, що дозволяє визначити сутності, їх атрибути та взаємозв'язки. Вона є основою для побудови фізичної моделі бази даних та реалізації програмної логіки інформаційної системи. У межах даної курсової роботи інфологічна модель реалізована у вигляді діаграми «сутність–зв'язок» (Entity–Relationship Diagram, ERD), яка відображає структуру бази даних «Електронний журнал».

Загальна структура моделі. На основі аналізу інформаційних потоків було виділено ключові сутності предметної області, які охоплюють учасників освітнього процесу, навчальні події, результати навчання, довідники та допоміжні таблиці. Взаємозв'язки між сутностями реалізовані через зовнішні ключі (FOREIGN KEY) та обмеження цілісності, що забезпечує логічну узгодженість даних [11; 16].

Фрагмент концептуальної схеми бази даних, що відображає основний процес обліку успішності, наведено на рисунку 2.1.

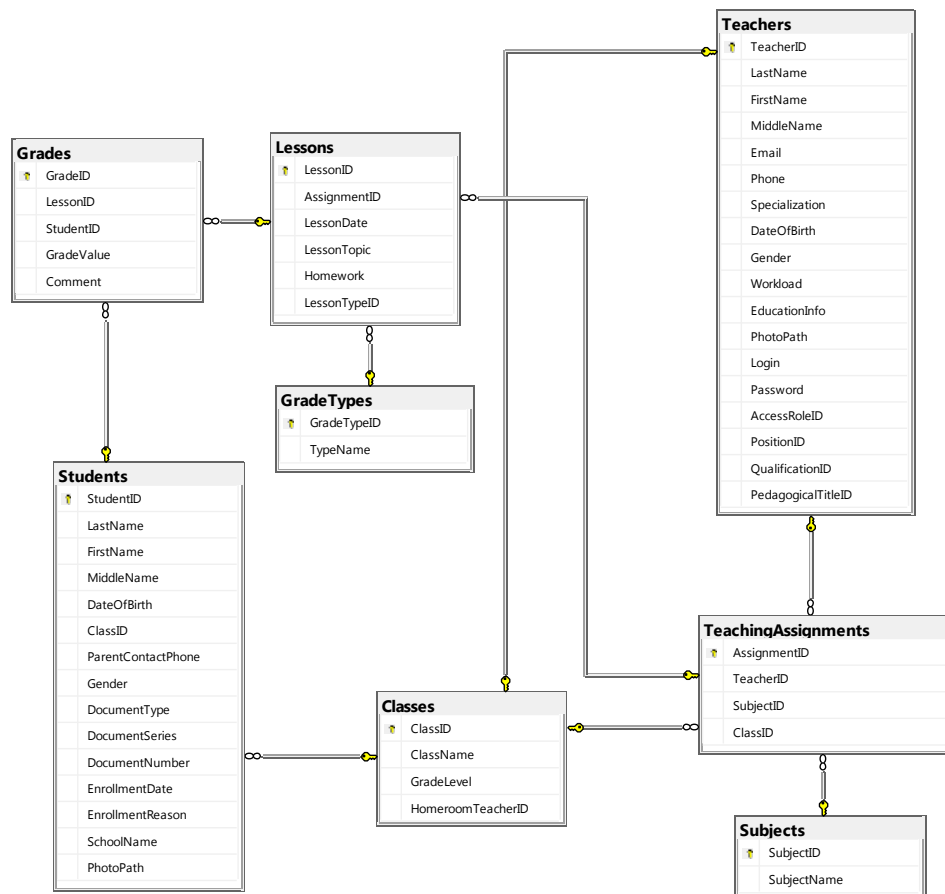


Рисунок 2.1 – Фрагмент інфологічної моделі підсистеми «Електронний журнал»



Опис основних сутностей:

1) **Учні (Students):** Містять персональні дані, дату народження, стать, документи, школу (SchoolName) та фото (PhotoPath). Зв'язок із таблицею Класи (Classes) реалізовано через поле ClassID, що забезпечує типову модель «багато-до-одного»: один клас — багато учнів.

2) **Класи (Classes):** Організаційні одиниці з назвою (ClassName), рівнем (GradeLevel) та класним керівником (HomeroomTeacherID). Для реалізації бізнес-правила «один вчитель — один клас» використано обмеження UNIQUE на поле HomeroomTeacherID.

3) **Вчителі (Teachers):** Зберігають інформацію про педагогічний склад: ПІБ, контакти, спеціалізацію, посаду (PositionID), кваліфікацію (QualificationID), педагогічне звання (PedagogicalTitleID), роль доступу (AccessRoleID), логін та пароль. Ці атрибути дозволяють реалізувати авторизацію та розмежування прав доступу.

4) **Предмети (Subjects):** Довідкова таблиця навчальних дисциплін. Поле SubjectName має обмеження UNIQUE, що запобігає дублюванню назв предметів.

5) **Навчальне навантаження (TeachingAssignments):** Асоціативна сутність, яка зв'язує вчителя (TeacherID), предмет (SubjectID) та клас (ClassID). Це дозволяє реалізувати зв'язок «багато-до-багатьох» між вчителями та класами. Обмеження UNIQUE (TeacherID, SubjectID, ClassID) забезпечує унікальність призначень.

6) **Уроки (Lessons):** Фіксують факт проведення заняття. Кожен запис містить дату (LessonDate), тему (LessonTopic), домашнє завдання (Homework) та тип уроку (LessonTypeID). Зв'язок із таблицею TeachingAssignments через AssignmentID дозволяє автоматично визначити, хто проводив урок і з якого предмета.

7) **Оцінки (Grades):** Замикаюча сутність, яка фіксує результат учня (StudentID) за конкретний урок (LessonID). Поле GradeValue містить оцінку, а Comment — додаткові пояснення. Такий каскадний зв'язок забезпечує логічну цілісність: оцінка не може існувати без уроку та учня.

8) **Типи оцінок (GradeTypes):** Довідкова таблиця, що класифікує види оцінювання: поточна, тематична, семестрова, групи результатів (ГР1–ГР4),

підсумкова тощо. Винесення типів у окрему таблицю відповідає вимогам третьої нормальної форми (3НФ) [27].

9) **Розклад (FixedSchedule):** Визначає день тижня (DayOfWeek) та номер уроку (LessonNumber) для кожного призначення (AssignmentID). Це дозволяє формувати тижневий розклад без дублювання предметів і вчителів.

10) **Батьки (Parents):** Контактні особи, пов'язані з учнями. Зв'язок реалізовано через StudentID, а обмеження ON DELETE CASCADE забезпечує автоматичне видалення записів при видаленні учня.

11) **Оголошення (Announcements):** Таблиця для зберігання повідомлень, які відображаються на головній сторінці.

12) **Довідники (Positions, Qualifications, PedagogicalTitles, AccessRoles):** Допоміжні таблиці, що забезпечують гнучкість у налаштуванні профілю вчителя та прав доступу.

Нормалізація та цілісність даних. Усі таблиці бази даних відповідають вимогам третьої нормальної форми (3НФ), що дозволяє уникнути надлишковості, дублювання даних та аномалій при оновленні [27]. Зовнішні ключі забезпечують референційну цілісність, а обмеження NOT NULL, UNIQUE, DEFAULT та CHECK — структурну узгодженість. Наприклад:

- Оцінка (Grades) не може бути записана без існуючого уроку (Lessons) та учня (Students).

- Урок (Lessons) не може бути створений без призначення (TeachingAssignments), яке визначає вчителя, предмет і клас.

- Класний керівник (HomeroomTeacherID) може бути призначений лише до одного класу.

Таким чином, розроблена інфологічна модель бази даних «Електронний журнал» охоплює всі ключові аспекти освітнього процесу: структуру класів, педагогічний склад, навчальні події, оцінювання, розклад, комунікацію з батьками та адміністративні довідники. Запропонована структура забезпечує логічну цілісність, масштабованість та зручність у реалізації запитів. Повна розгорнута

схема наведена у Додатку А, а її фізична реалізація у вигляді SQL-коду — у Додатку Б.

## **2.4 UML-діаграма послідовності роботи з електронним журналом**

Для моделювання динамічної взаємодії між користувачем і компонентами системи «Електронний журнал» використано діаграми послідовності UML (Sequence Diagram). Вони відображають часову логіку обміну повідомленнями між об'єктами системи, що особливо важливо для розуміння внутрішніх процесів обробки даних [25].

У системі реалізовано два ключові сценарії взаємодії, діаграми яких наведено у Додатку В:

1) Завантаження журналу (Рисунок В.1): отримання списку учнів, тем уроків та існуючих оцінок.

2) Збереження оцінки (Рисунок В.2): введення, валідація даних та запис у базу даних.

Сценарій 1: Завантаження електронного журналу. Цей процес ініціюється, коли вчитель обирає клас і предмет для перегляду. Інтерфейс надсилає запит до шару бізнес-логіки, який виконує серію оптимізованих SQL-запитів. Алгоритм взаємодії включає такі кроки (див. Рисунок В.1):

- 1) Вчитель обирає параметри журналу (Клас, Предмет).
- 2) ViewModel (модуль логіки) виконує асинхронні запити до бази даних для отримання списків учнів, уроків та оцінок.
- 3) Отримані «сирі» дані агрегуються в колекцію ObservableCollection.
- 4) Інтерфейс відображає заповнену таблицю користувачеві.

Сценарій 2: Збереження оцінки. Процес моделює внесення нової оцінки або редагування існуючої. Ключовим етапом тут є валідація даних перед відправкою на сервер (див. Рисунок В.2). Послідовність дій:

- 1) Вчитель вводить значення у комірку.
- 2) Система виконує валідацію: перевіряє, чи є значення допустимим (числа 1–12, «н», «хв»).

3) Якщо дані коректні, формується SQL- запит (INSERT або UPDATE) і надсилається до бази даних [24].

4) Після успішного виконання транзакції інтерфейс виводить повідомлення «Дані збережено».

5) Якщо валідація не пройдена, система блокує запит і підсвічує помилку, не навантажуючи базу даних.

Архітектурні особливості. Використання діаграм послідовності дозволило спроектувати систему згідно з патерном MVVM, чітко розділивши інтерфейс (View) та логіку (ViewModel) [22; 36]. Також передбачено асинхронний режим роботи (async/await), що забезпечує стабільність інтерфейсу під час виконання складних запитів до SQL Server [23].

## ВИСНОВКИ ДО РОЗДІЛУ 2

У другому розділі здійснено комплексне дослідження та проєктування інформаційної системи «Електронний журнал». Основні результати роботи полягають у наступному:

1. Проведено системний аналіз предметної області та визначено ключові інформаційні потоки (оцінки, розклад, навантаження). Сформульовано функціональні вимоги до системи, що дозволило перейти від моделі «AS-IS» (паперовий облік) до моделі «TO-BE» (автоматизована система).

2. Обґрунтовано вибір технологічного стеку: СУБД Microsoft SQL Server обрано за підтримку транзакцій та збережуваних процедур, а зв'язку C# / WPF — за можливості створення сучасного адаптивного інтерфейсу та підтримку патерну MVVM.

3. Розроблено нормалізовану інфологічну модель бази даних (відображену в Додатку А), яка включає 14 взаємопов'язаних сутностей. Забезпечено третю нормальну форму (3НФ) для уникнення надлишковості даних.

4. Реалізовано фізичну структуру бази даних у вигляді оптимізованого SQL-скрипту, який наведено у Додатку Б.

5. Спроектовано динамічну поведінку системи за допомогою UML-діаграм послідовності (наведені у Додатку В). Деталізовано алгоритми завантаження журналу та збереження оцінок, що дозволило виявити необхідність асинхронної взаємодії з сервером.

Таким чином, у даному розділі закладено надійну архітектурну основу для програмної реалізації системи. Отримані проєктні рішення гарантують цілісність даних, масштабованість системи та зручність подальшої розробки, опис якої буде наведено у третьому розділі.

## РОЗДІЛ 3 ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ «ЕЛЕКТРОННИЙ ЖУРНАЛ»

### 3.1 Обґрунтування вибору засобів розробки

Для програмної реалізації спроектованої системи було обрано інтегроване середовище розробки (IDE) Microsoft Visual Studio 2022 [39]. Цей вибір зумовлений повною підтримкою платформи .NET, наявністю потужних інструментів налагодження (debugger), вбудованим дизайнером XAML для WPF та зручною роботою з рішеннями, що складаються з багатьох файлів.

У якості цільової платформи обрано .NET 8.0. Це остання версія з довгостроковою підтримкою (LTS) від Microsoft [34; 38], яка забезпечує кросплатформеність, найвищу продуктивність серед попередніх версій та покращену роботу з пам'яттю, що є критично важливим для стабільної роботи на шкільних комп'ютерах.

Для реалізації функціоналу системи до проєкту було підключено ряд зовнішніх бібліотек та просторів імен:

1) `System.Data.SqlClient` — для забезпечення прямого з'єднання з базою даних MS SQL Server та виконання SQL-команд [31].

2) `System.Collections.ObjectModel` — для використання колекції `ObservableCollection<T>`, яка забезпечує автоматичне оновлення інтерфейсу при зміні даних (Data Binding) [32].

3) `System.Windows.Controls` — для роботи з графічними компонентами WPF (`DataGrid`, `TextBox`, `Button`) [28].

Структура розробленого рішення (Solution) побудована згідно з патерном MVVM [22; 36] і включає розгалужений комплекс файлів вихідного коду, які логічно розподілені по папках:

- Model: класи, що описують сутності бази даних (зокрема `Student.cs`, `Teacher.cs`, `Grade.cs`).

- View: набір XAML-файлів, що відповідають за візуальне відображення вікон (`LoginView.xaml`, `JournalView.xaml` тощо).

- ViewModel: класи, що містять бізнес-логіку та обробку команд користувача (наприклад, MainViewModel.cs, AuthViewModel.cs).

- Helpers: допоміжні класи для шифрування паролів, роботи з базою даних та валідації.

Повна структура проєкту, що відображає ієрархію папок та файлів у середовищі Visual Studio, наведена у Додатку Г.

Окрему увагу приділено візуальному оформленню застосунку. Для створення сучасного інтерфейсу використано стиль Material Design, який реалізовано через кастомні стилі (Styles) та шаблони елементів керування (ControlTemplates) у файлі App.xaml. Це дозволило забезпечити єдиний вигляд кнопок, полів введення та таблиць, незалежно від версії операційної системи Windows користувача.

Така організація проєкту забезпечує високу модульність системи, дозволяє легко орієнтуватися у структурі рішення та значно спрощує подальшу підтримку і масштабування програмного продукту.

### **3.2 Проєктування структури бази даних**

Фізична реалізація бази даних системи «Електронний журнал» виконана у середовищі Microsoft SQL Server. На етапі проєктування структури було створено скрипт мовою T-SQL, який автоматизує процес розгортання бази даних (див. Додаток Б) [24].

Процес створення структури включає наступні етапи:

- 1) Створення контейнера бази даних: Використано команду CREATE DATABASE DB\_SchoolJournal, яка ініціалізує нову базу даних із налаштуваннями за замовчуванням. Для уникнення конфліктів передбачено перевірку на існування бази з таким іменем перед створенням.

- 2) Визначення типів даних: Для зберігання текстової інформації (прізвища, імена, теми уроків) обрано тип NVARCHAR. На відміну від VARCHAR, він підтримує кодування Unicode, що є критично важливим для коректного відображення кирилиці (української мови) [24]. Для первинних ключів

використано цілочисельний тип INT із властивістю IDENTITY(1,1), що забезпечує автоматичну нумерацію записів (автоінкремент) та гарантує унікальність ідентифікаторів.

3) Створення таблиць та обмежень цілісності: У структурі бази даних реалізовано жорсткі обмеження (Constraints) для забезпечення валідності даних [16; 27]:

- PRIMARY KEY: Кожна таблиця має первинний ключ (наприклад, StudentID, LessonID) для однозначної ідентифікації записів.

- FOREIGN KEY: Використано для зв'язку між таблицями. Наприклад, у таблиці TeachingAssignments поля TeacherID, SubjectID та ClassID є зовнішніми ключами, що посиляються на відповідні довідники. Це унеможливорює видалення вчителя, якщо за ним закріплені уроки.

- UNIQUE: Для запобігання дублюванню даних встановлено унікальні індекси. Зокрема, у таблиці TeachingAssignments композитний ключ UNIQUE (TeacherID, SubjectID, ClassID) гарантує, що одному вчителю не можна двічі призначити той самий предмет у тому ж класі.

- ON DELETE CASCADE: У таблиці Parents використано каскадне видалення. Це означає, що при видаленні учня з бази автоматично видаляється інформація про його батьків, що запобігає накопиченню «сміттєвих» даних.

Крім того, для підвищення швидкодії виконання пошукових запитів, на поля LastName (прізвище учня) та LessonDate (дата уроку) неявно створено кластерні індекси. Це дозволяє миттєво знаходити потрібні записи навіть при наявності десятків тисяч рядків у таблицях оцінок та відвідуваності.

### **3.3 Реалізація SQL-запитів для основної логіки**

Для забезпечення взаємодії клієнтського застосунку з базою даних розроблено набір SQL-запитів, які реалізують основні сценарії роботи користувача. Логіка запитів побудована на мові маніпулювання даними (DML) [24].

Розглянемо реалізацію ключових операцій:



1) Завантаження списку учнів (Формування журналу): Коли вчитель відкриває журнал конкретного класу, система виконує запит на вибірку учнів, відсортованих за алфавітом.

```
SELECT StudentID, LastName, FirstName
FROM Students
WHERE ClassID = @SelectedClassID
ORDER BY LastName;
```

Використання параметра @SelectedClassID захищає систему від SQL-ін'єкцій [31].

2) Отримання сітки уроків: Для відображення колонок із датами та темами уроків використовується вибірка з таблиці Lessons, яка фільтрується за конкретним призначенням (AssignmentID).

```
SELECT LessonID, LessonDate, LessonTopic, LessonTypeID
FROM Lessons
WHERE AssignmentID = @CurrentAssignmentID
ORDER BY LessonDate;
```

3) Збереження та оновлення оцінок: Процес виставлення оцінки реалізовано через пряму вставку даних у таблицю Grades. Оскільки оцінка прив'язана до конкретного уроку та учня, запит має вигляд:

```
INSERT INTO Grades (LessonID, StudentID, GradeValue, Comment)
VALUES (@LessonID, @StudentID, @GradeValue, @Comment);
```

У випадку редагування оцінки застосовується оператор UPDATE з умовою пошуку по LessonID та StudentID [24].

4) Генерація звітних даних: Для формування семестрових звітів використовуються запити з агрегатними функціями та об'єднанням таблиць (JOIN). Наприклад, для підрахунку середнього балу учня система вибирає всі оцінки певного типу (тематичні) та обчислює їх середнє арифметичне значення на стороні сервера або клієнта [24].

5) Забезпечення атомарності операцій: При виконанні складних дій, таких як видалення вчителя або переведення учнів між класами, використовується механізм транзакцій (BEGIN TRANSACTION ... COMMIT). Це гарантує, що якщо під час виконання операції станеться помилка (наприклад, збій мережі), всі зміни будуть скасовані (ROLLBACK), і база даних залишиться у цілісному стані.

Реалізація цих запитів на рівні бібліотеки `System.Data.SqlClient` у кодї `C#` дозволяє забезпечити високу швидкодію системи навіть при великій кількості записів у базі даних [31].

### **3.4 Програмна реалізація інтерфейсу користувача та бізнес-логіки системи**

На основі спроектованої архітектури було розроблено комплексний клієнтський застосунок «School Journal App». Програмний продукт реалізовано мовою `C#` з використанням технології WPF (Windows Presentation Foundation), що дозволило створити гнучкий, масштабований та інтуїтивно зрозумілий інтерфейс. Архітектура рішення базується на принципах MVVM (Model-View-ViewModel), де логіка взаємодії з користувачем відокремлена від бізнес-логіки та доступу до даних.

Нижче наведено детальний опис реалізованих функціональних модулів системи.

1. Модуль авторизації та конфігурації підключення. Система безпеки реалізована у класі `LoginView`. На відміну від стандартних рішень, модуль входу підтримує два режими роботи:

- Конфігурація з'єднання: Якщо система виявляє відсутність збереженого рядка підключення (`AppSession.ConnectionString`), користувачеві пропонується інтерфейс для налаштування параметрів сервера (IP-адреса або ім'я хоста) та назви бази даних. Це дозволяє розгортати програму в будь-якій мережевій інфраструктурі без перекомпіляції коду.

- Автентифікація користувача: Реалізовано механізм перевірки облікових даних. Паролі не зберігаються у відкритому вигляді; натомість використовується алгоритм хешування SHA-256 (метод `AppSession.HashPassword`), що унеможливорює злам акаунтів навіть при доступі злоумисника до бази даних.

- Функція «Запам'ятати мене»: Реалізована через клас `LocalSettings`, який безпечно зберігає логін останнього користувача у локальному файлі конфігурації, спрощуючи щоденний вхід у систему.

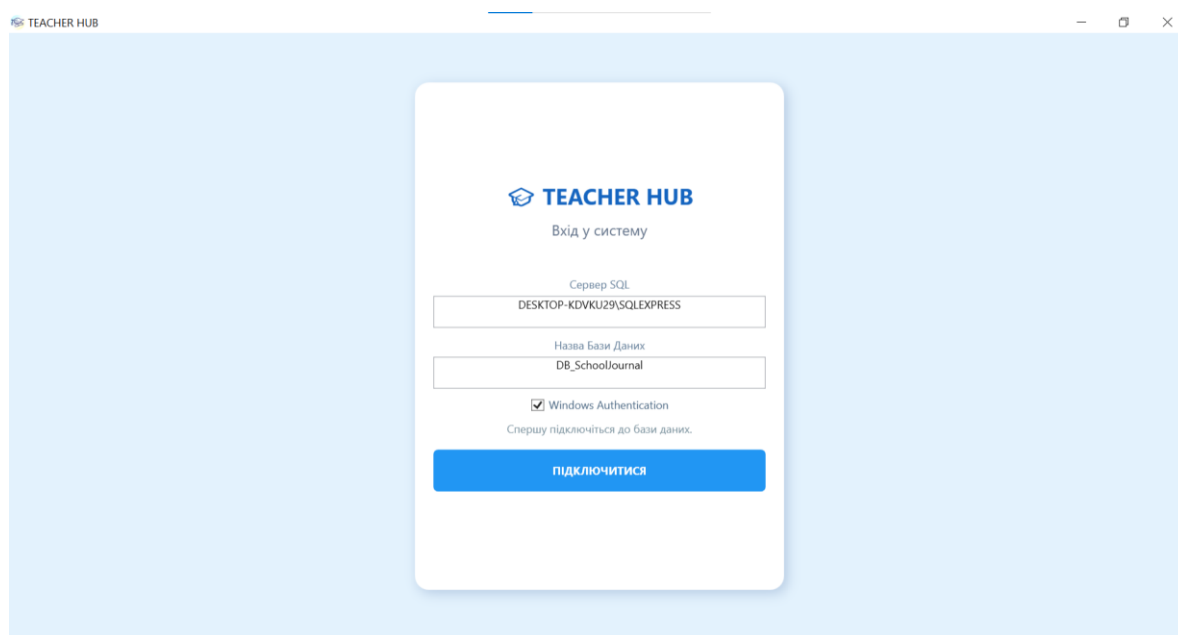


Рисунок 3.1 – Вікно авторизації: конфігурація підключення до SQL Server

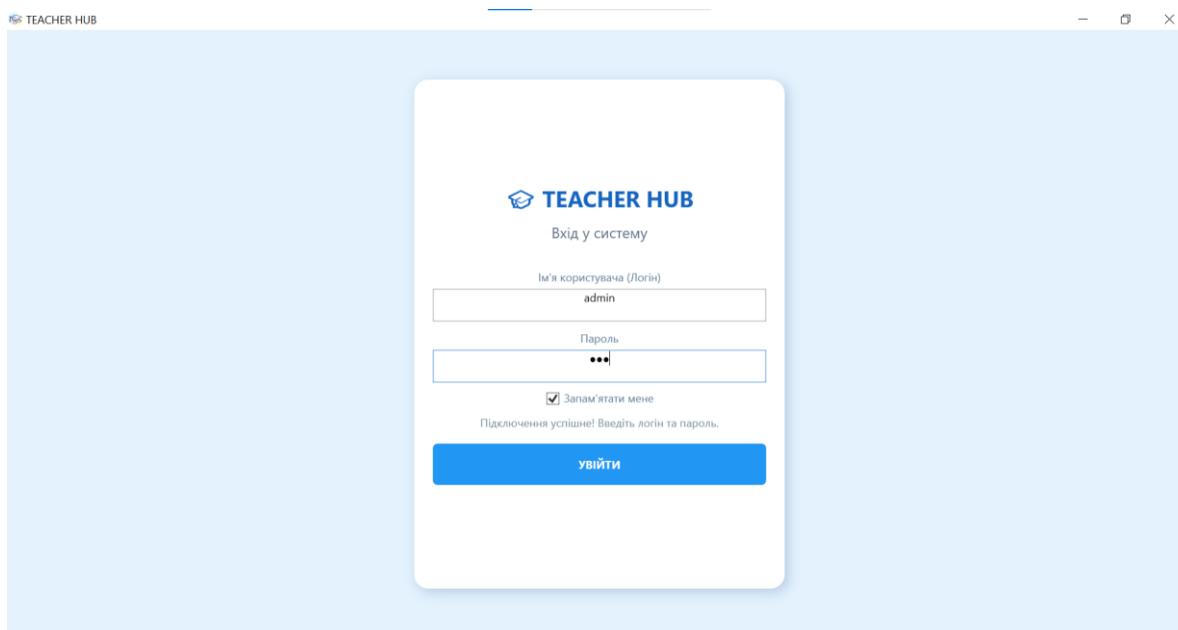


Рисунок 3.2 – Вікно авторизації: введення облікових даних

2. Головна панель керування (Dashboard) Після успішної авторизації завантажується DashboardView. Логіка цього модуля є адаптивною і залежить від ролі користувача (CurrentRoleID), яка визначається на етапі входу:

- Рольова модель: Якщо користувач є адміністратором, система виконує SQL-запити для підрахунку глобальної статистики (загальна кількість учнів, класів). Якщо це вчитель, запити автоматично фільтруються за його TeacherID, відображаючи лише закріплені за ним класи та предмети.

Інтерактивний розклад: Реалізовано віджет «Уроки сьогодні», який виконує складний SQL-запит з JOIN-об'єднанням таблиць Lessons, Subjects та Classes, відображаючи список актуальних занять на поточну дату.

Система оголошень: Реалізовано внутрішній канал комунікації. Адміністратори мають змогу додавати та видаляти загальношкільні оголошення, які миттєво з'являються на панелях усіх користувачів.

Окрему увагу приділено ергономіці та візуальному оформленню головної панелі. Ключові показники ефективності (KPI), такі як кількість учнів та активність класів, представлені у вигляді інтерактивних карток («плиток»). Для їх реалізації використано елементи Border із закругленими кутами та градієнтним фоном (LinearGradientBrush), а також ефект тіні (DropShadowEffect), що надає інтерфейсу об'єму та сучасності.

Крім того, у правому верхньому куті реалізовано блок ідентифікації, який динамічно відображає прізвище та ініціали поточного користувача на основі даних сесії (AppSession). Це дозволяє уникнути помилок при роботі кількох вчителів за одним комп'ютером. Оновлення всієї статистики відбувається асинхронно при кожному переході на вкладку «Панель», що забезпечує актуальність даних у реальному часі.

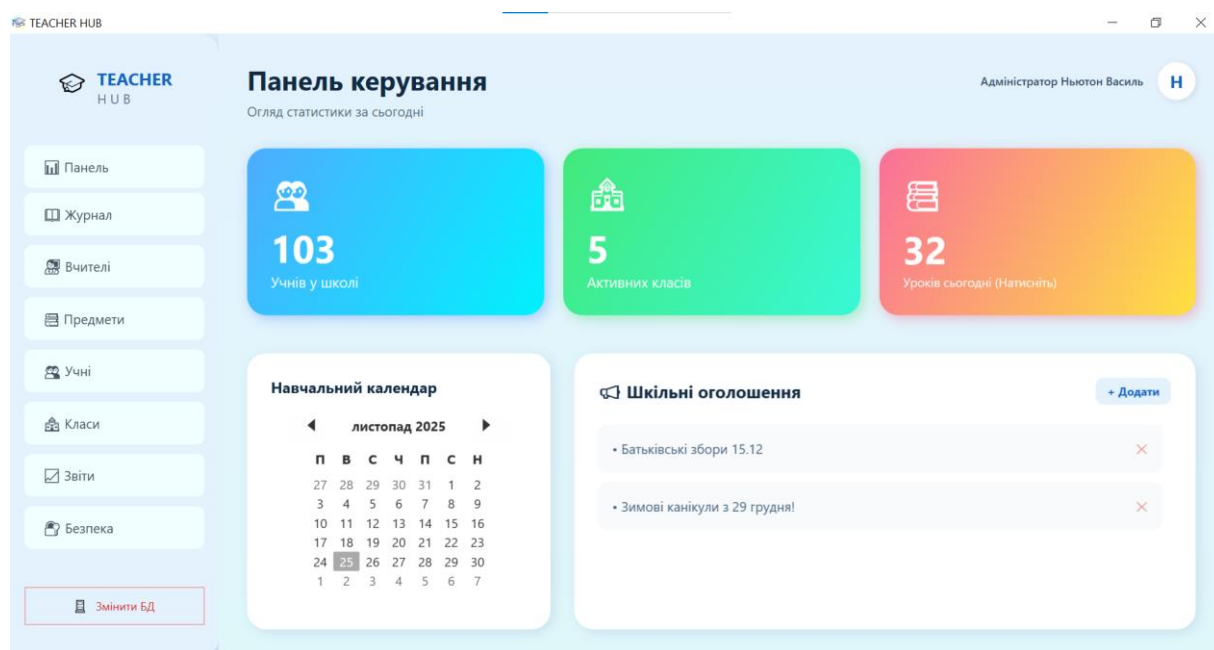


Рисунок 3.3 – Головна панель керування (режим адміністратора)

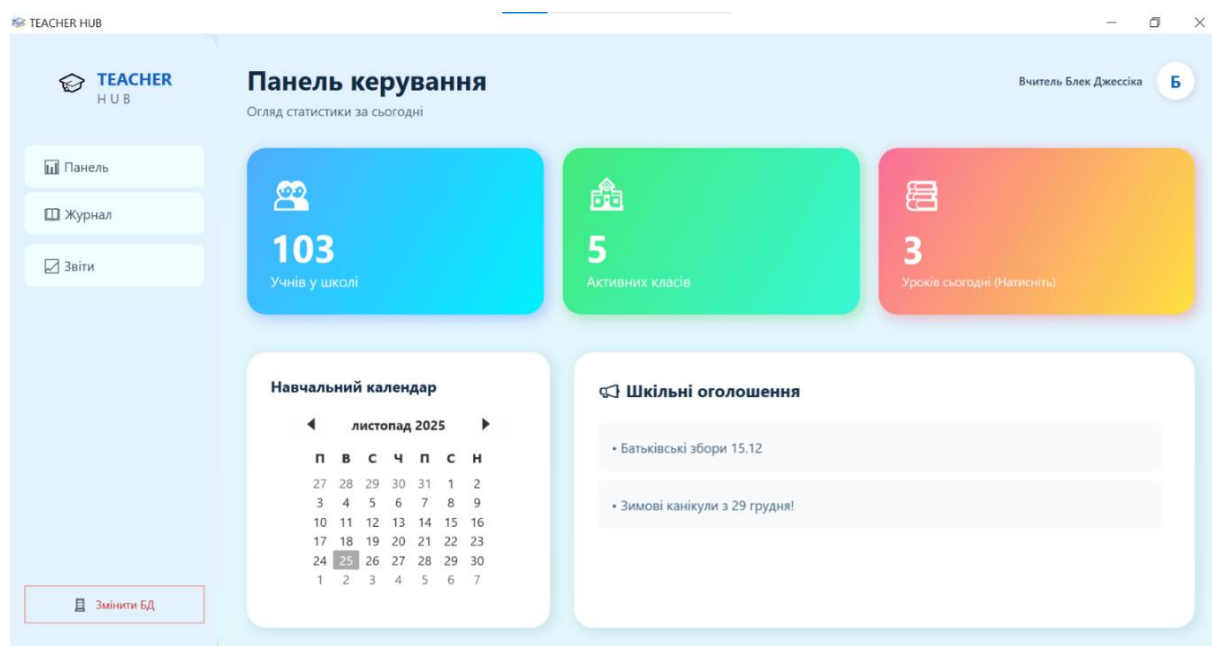


Рисунок 3.4 – Головна панель керування (режим вчителя)

3. Електронний журнал (Journal Core). Це найбільш технологічно складний модуль системи (JournalView), який реалізує функціонал динамічного відображення даних.

- Динамічна побудова колонок: Оскільки кількість уроків не є фіксованою, інтерфейс таблиці (DataGrid) формується програмно. Алгоритм проходить через результат SQL-вибірки уроків і для кожного запису створює нову колонку, у заголовок якої виводиться дата та тип уроку.

- Підтримка НУШ (Нова українська школа): У коді реалізовано словник `_pushDescriptions`, який містить перелік компетентностей (Груп результатів) для кожної освітньої галузі (Математична, Мовно-літературна, Природнича тощо). При виборі предмета система автоматично визначає галузь і пропонує вчителю відповідні типи оцінювання.

- Візуалізація успішності: Розроблено власний конвертер значень `GradeColorConverter`, який у реальному часі аналізує введені оцінки. Низькі бали (1–3) підсвічуються червоним, середні — жовтим, а високі (10–12) — зеленим кольором, що дозволяє педагогу миттєво оцінити загальну картину успішності класу.

- Інтеграція з Google Meet: Для дистанційного навчання реалізовано функцію збереження та швидкого виклику посилання на відеоконференцію. Посилання прив'язується до контексту уроку і відкривається у браузері за замовчуванням одним кліком.

Технічна реалізація динамічного журналу базується на використанні колекцій Dictionary<string, int>, які зберігають відповідність між візуальною колонкою таблиці та унікальним ідентифікатором уроку (LessonID) у базі даних. Це дозволяє системі безпомилково визначати контекст редагування при кліку на будь-яку клітинку, незалежно від кількості колонок чи їх порядку. Заголовки стовпців також генеруються програмно, поєднуючи дату проведення заняття та його тип у зручному для сприйняття форматі.

Для підвищення ергономічності процесу оцінювання відмовлено від стандартного ручного введення тексту. Натомість розроблено спеціалізоване спливаюче вікно (Popup), яке активується при виборі комірки. Воно пропонує вчителю стандартизований набір балів та відміток, що виключає можливість введення некоректних даних та значно прискорює роботу з журналом.

**Електронний журнал**

Клас: 9 Предмет: Інформатика Завантажити 25.11.2025 + Урок Meet

№	Учень	04.09 ГР 1	11 ГР 2	18 ГР 3	25 ГР 4	02.10 ГР 1	09 ГР 2	16 ГР 3	23 ГР 4	30 ГР 1	06.11 (Тем) ГР 3	13 ГР 4	20 ГР 1	27 ГР 2	04.12 ГР 3	11 ГР 4	18 ГР 1	25 ГР 2
1	Бабич Єгор						5	4	12		10	хв	10					
2	Демченко Юлія			4	7		9		12	7	8	12	6					
3	Демченко Владислав			Н		хв	9	хв			7	5	10					
4	Коваленко Катерина			4		6	5		10		11	хв						
5	Кузьменко Соломія					11	11			10	7	5						
6	Кузьменко Андрій			11			4		4		10	хв	8					
7	Кузьменко Анастасія	хв			11			9	11		7	11	9					
8	Лисенко Мілана				8	6	8	10		8	10	12	4					
9	Лисенко Владислав			хв		12	4		12	хв	6	9	7					

**Теми уроків та домашні завдання**

Дата	№	Тема уроку	Домашнє завдання
04.09.2025	1	Інформація та повідомлення (Працює з інформацією, даними, моделями)	Опрацювати матеріал
11.09.2025	2	Комп'ютерні пристрої (Створює інформаційні продукти)	Опрацювати матеріал
18.09.2025	3	ОС Windows (Працює в цифровому середовищі)	Опрацювати матеріал

Імпорт (Excel) Експорт (Excel) PDF

Рисунок 3.5 – Загальний вигляд електронного журналу з динамічними колонками

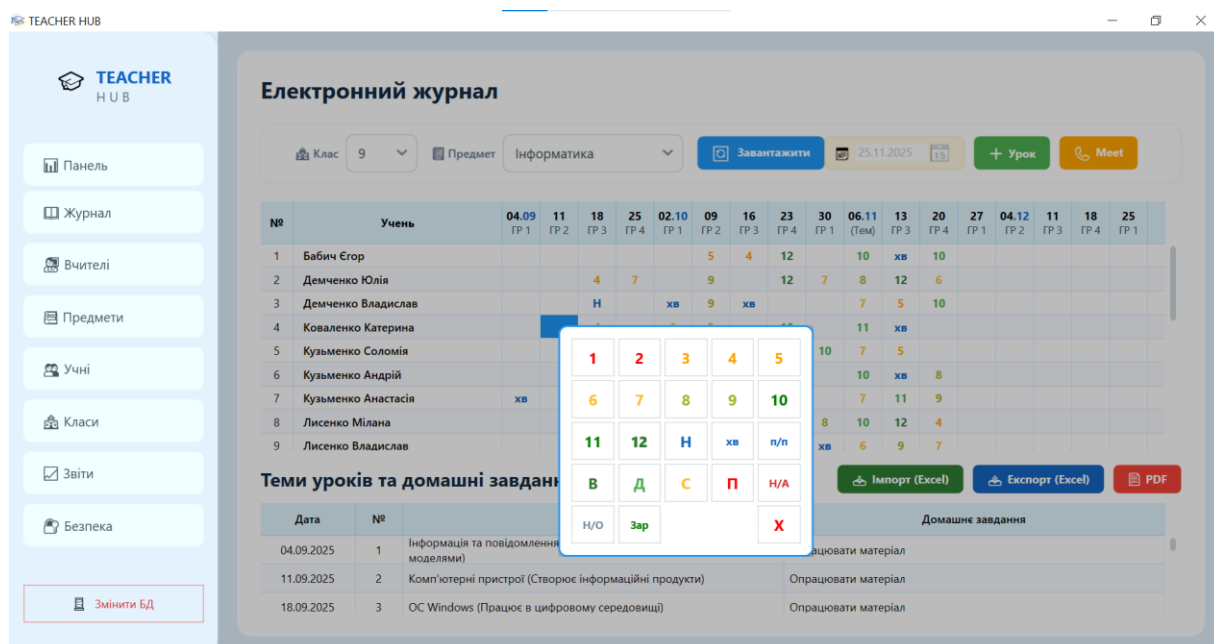


Рисунок 3.6 – Інтерфейс електронного журналу: режим виставлення оцінок

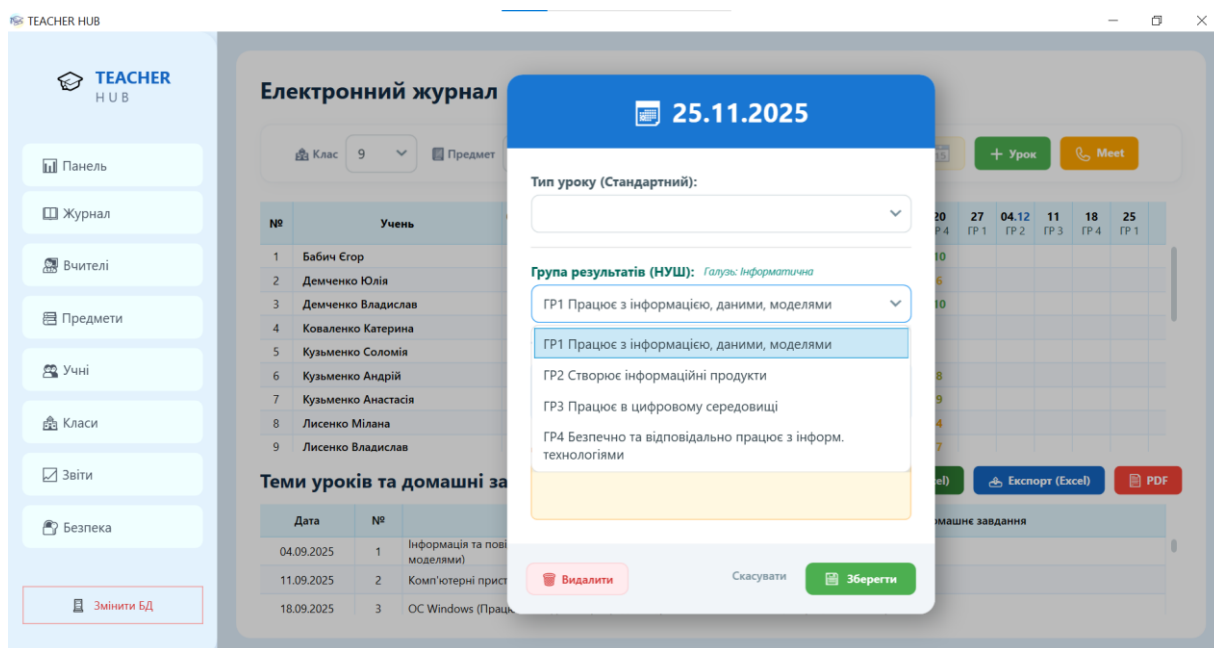


Рисунок 3.7 – Модуль редагування уроку з вибором компетентностей НУШ

4. Керування контингентом та транзакції. Модулі StudentFormWindow та ClassStudentsWindow забезпечують повний цикл керування даними учнів.

- Транзакційна цілісність: При виконанні складних операцій, таких як переведення учня до іншого класу або видалення профілю вчителя, використовується механізм SqlTransaction. Це гарантує, що зміни будуть застосовані атомарно: або всі пов'язані записи (оцінки, батьки, навантаження)

будуть успішно оновлені, або, у випадку помилки, відбудеться повний відкат (Rollback) змін, що запобігає пошкодженню бази даних.

- Інтерфейс переміщення учнів: Для формування класів розроблено зручний двопанельний інтерфейс (TransferListStyle), що дозволяє переміщувати учнів між списками «Доступні» та «У класі» за допомогою кнопок, автоматично оновлюючи ClassID у базі даних.

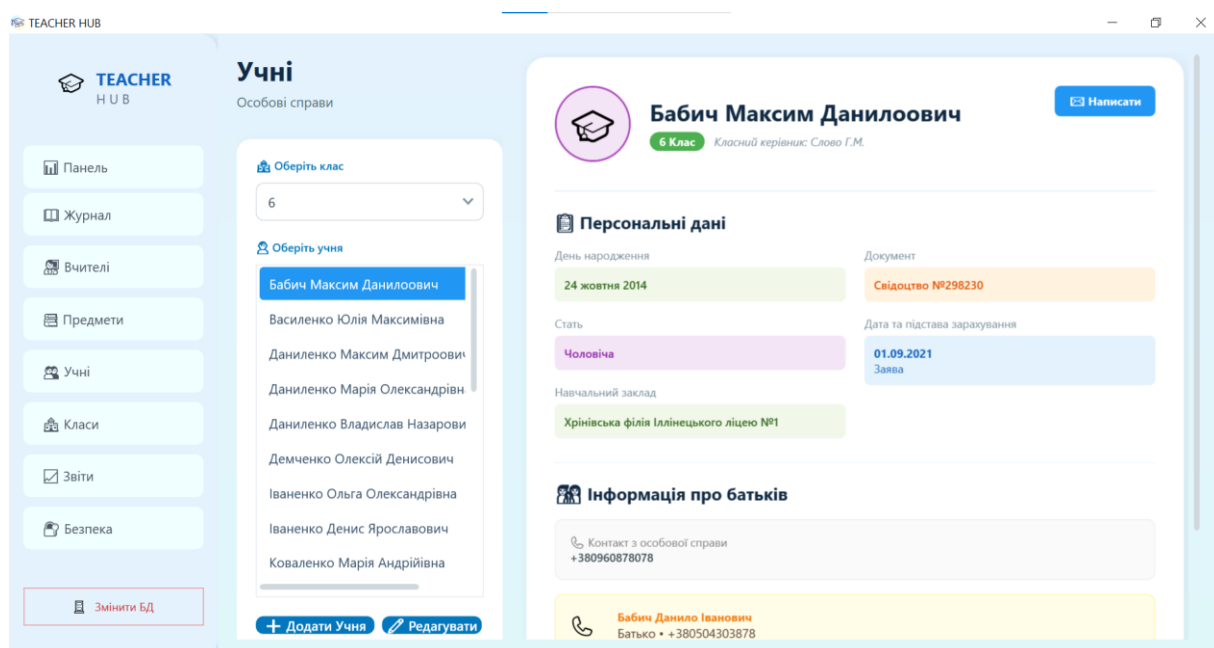


Рисунок 3.8 – Модуль керування контингентом: перегляд особової справи учня

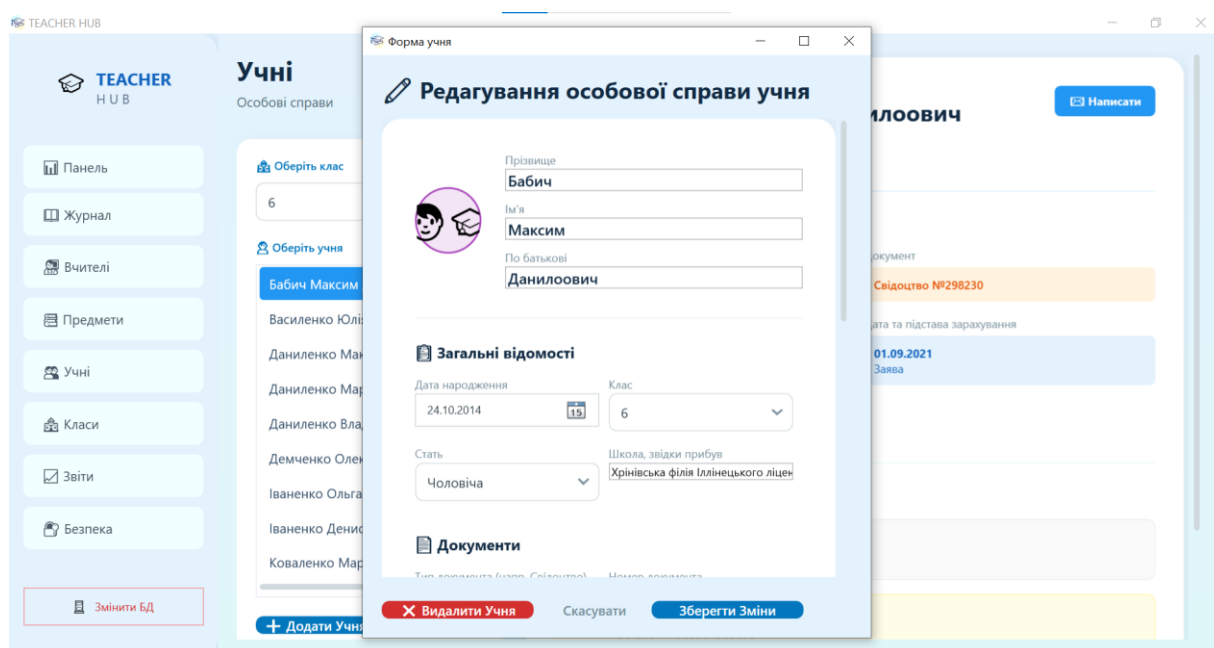


Рисунок 3.9 – Форма редагування даних учня



5. Модуль адміністратора та безпека. Спеціалізоване вікно AccessControlView надає адміністратору повний контроль над доступом до системи.

- Керування ролями: Реалізовано можливість зміни ролі користувача (Вчитель / Завуч / Адміністратор) «на льоту» через оновлення зовнішнього ключа AccessRoleID.

- Блокування доступу: Функція відкликання прав дозволяє миттєво заблокувати вхід звільненому працівнику, очищуючи його логін та пароль у базі даних, але зберігаючи історію його роботи.

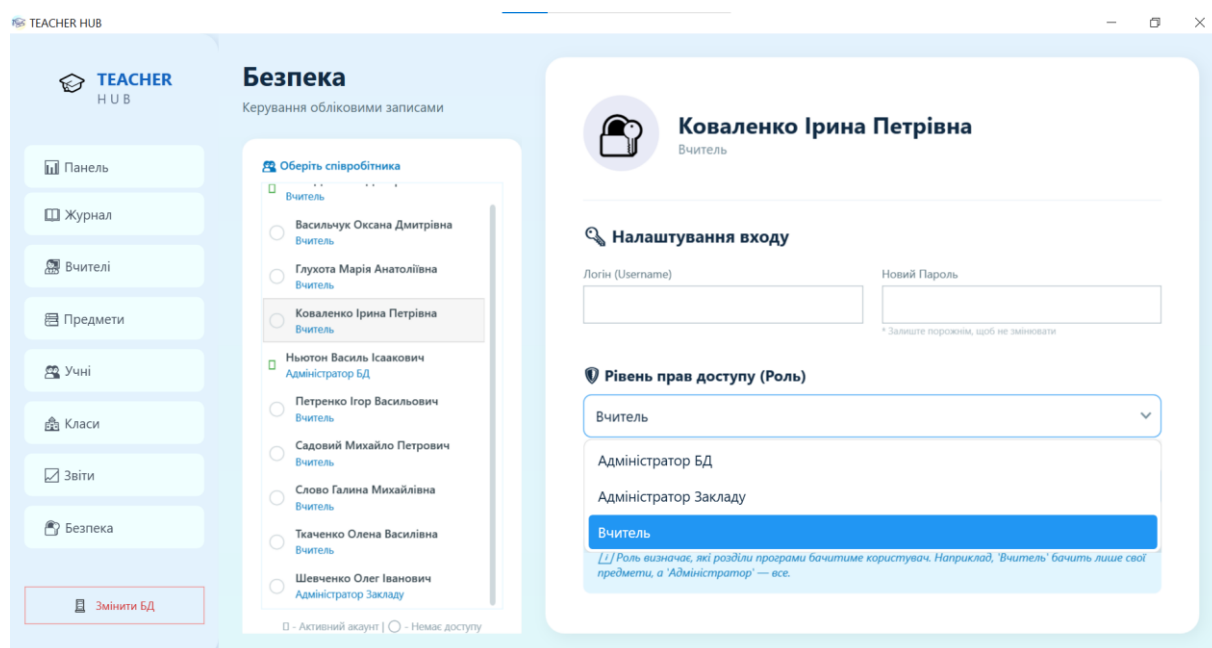


Рисунок 3.10 – Панель адміністрування облікових записів та прав доступу

6. Генерація звітності та експорт. Модуль ReportsView використовує асинхронні методи (async/await) для формування складних аналітичних звітів без блокування інтерфейсу користувача.

- Алгоритм побудови PDF: Розроблено унікальний алгоритм розбиття великих таблиць на сторінки (FlowDocument). Система автоматично розраховує ширину колонок, і якщо звіт (наприклад, семестровий журнал) не вміщується на одній сторінці, він коректно переноситься на наступну з дублюванням шапки таблиці та прізвищ учнів.

- Экспорт у Excel: Для подальшої аналітичної обробки реалізовано експорт будь-якого звіту у формат CSV, який коректно розпізнається Microsoft Excel.

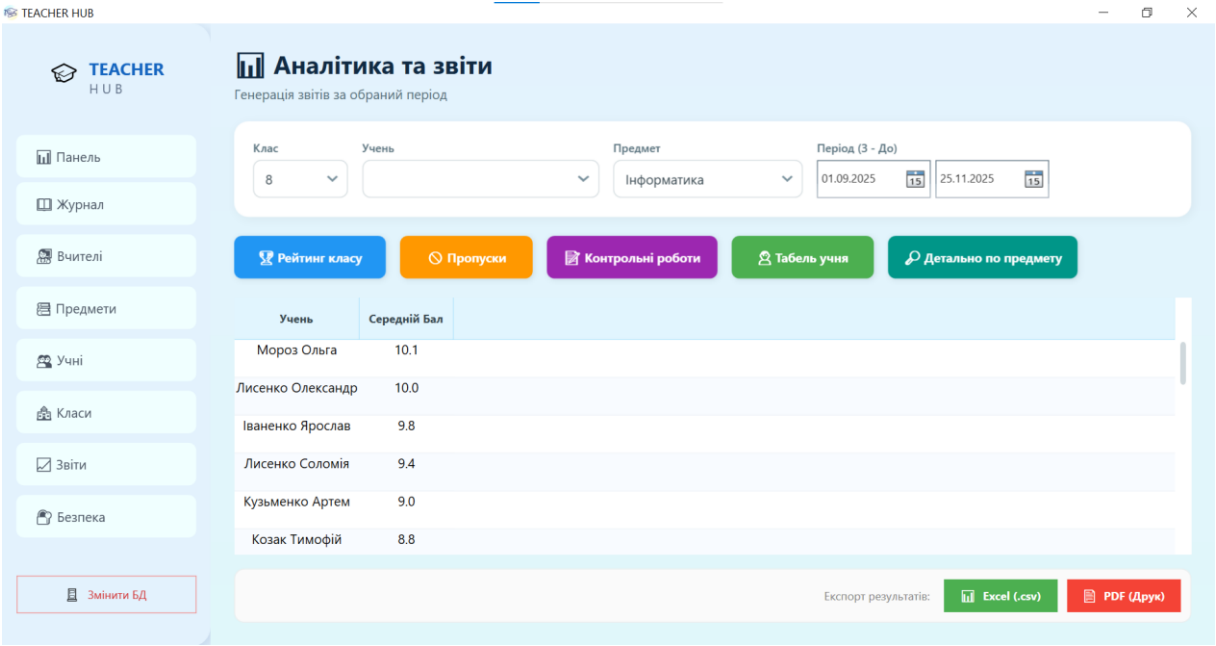


Рисунок 3.11 – Модуль генерації аналітичних звітів: перегляд рейтингу успішності класу

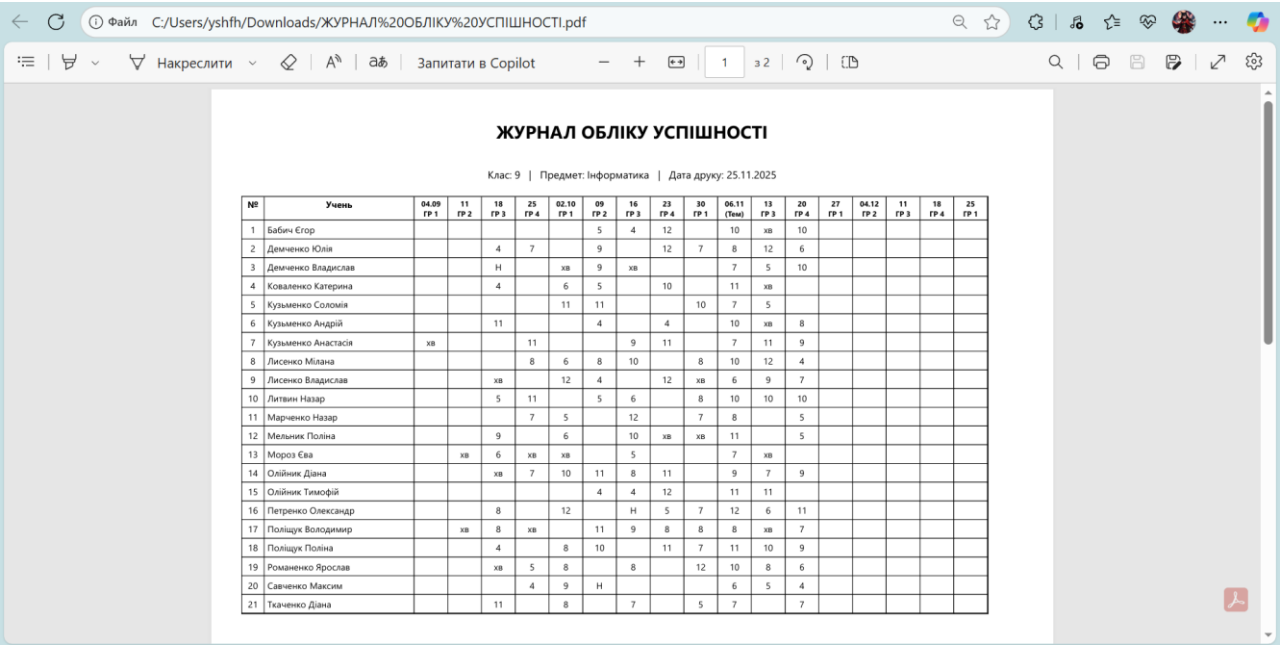


Рисунок 3.12 – Друкована форма класного журналу, згенерована системою (формат PDF)

### ВИСНОВКИ ДО РОЗДІЛУ 3

У третьому розділі описано процес програмної реалізації інформаційної системи «Електронний журнал». На основі спроектованої архітектури створено комплексний програмний продукт, що складається з клієнтської частини (WPF) та серверної бази даних (MS SQL Server).

Основні результати реалізації:

1. База даних: Розгорнуто фізичну структуру БД, яка включає 15 таблиць (14 взаємопов'язаних та одну службову) із налаштованими зв'язками та обмеженнями цілісності. Реалізовано набір оптимізованих SQL-запитів для вибірки даних, що забезпечує швидку роботу навіть при великій кількості записів.

2. Архітектура застосунку: Програма побудована за модульним принципом із використанням патерну MVVM (Model-View-ViewModel). Код проєкту структуровано за логічними компонентами (моделі, представлення, допоміжні класи), що забезпечує легкість підтримки та масштабування.

3. Функціональність: Реалізовано повний набір функцій для вчителя: авторизація з хешуванням паролів, динамічний електронний журнал, підтримка оцінювання за стандартами НУШ, керування контингентом учнів та генерація аналітичних звітів.

4. Інтерфейс та швидкодія: Використання технології WPF дозволило створити сучасний адаптивний інтерфейс. Застосування асинхронного програмування (async/await) гарантує стабільну роботу програми та відсутність «зависань» під час виконання складних операцій з базою даних.

Таким чином, розроблена система є діючим функціональним прототипом, який успішно вирішує поставлені задачі автоматизації навчального процесу та готовий до дослідної експлуатації в умовах закладу освіти.

## ВИСНОВКИ

У курсовій роботі розв'язано актуальне науково-практичне завдання проєктування та розробки інформаційної системи електронного журналу для закладів загальної середньої освіти. На основі проведеного дослідження зроблено такі висновки:

1. Проведено системний аналіз предметної області та нормативно-правової бази. Встановлено, що існуючі хмарні аналоги створюють залежність від стабільності інтернет-з'єднання. Обґрунтовано доцільність створення локальної клієнт-серверної системи, яка забезпечує автономність роботи, швидкий доступ до даних та захист інформації всередині периметра закладу.

2. Спроектовано інфологічну модель бази даних у вигляді ER-діаграми, що включає 14 взаємопов'язаних сутностей, та динамічну модель взаємодії компонентів системи (UML-діаграми послідовності). Це дозволило деталізувати логіку обробки даних ще на етапі проєктування та уникнути архітектурних помилок.

3. Реалізовано фізичну структуру бази даних у середовищі MS SQL Server. Створено набір оптимізованих SQL-запитів та збережуваних процедур, які забезпечують цілісність даних, каскадне видалення пов'язаних записів та високу швидкість вибірки інформації для журналів і звітів.

4. Розроблено графічний інтерфейс та бізнес-логіку клієнтського застосунку «Teacher Hub» мовою C# з використанням технології WPF та патерну MVVM. Застосування асинхронного програмування та модульної архітектури дозволило створити чутливий інтерфейс, який не блокується під час виконання складних операцій з базою даних.

Практичне значення роботи полягає у розробці функціонального прототипу інструментарію, який демонструє можливість автоматизації рутинної роботи вчителя, мінімізації механічних помилок при підрахунку балів та забезпечення надійного зберігання академічних записів незалежно від зовнішніх факторів.

Перспективи подальшого розвитку розробленої системи вбачаються у створенні мобільного додатку для оперативного доступу батьків до оцінок, а також

у реалізації модуля штучного інтелекту для аналізу успішності та надання персональних рекомендацій учням. Крім того, архітектура проєкту дозволяє в майбутньому інтегрувати систему з хмарними сховищами для автоматичного резервного копіювання бази даних.

Окрім технічних переваг, впровадження системи має важливий соціально-педагогічний ефект. Автоматизація рутинних процесів звільняє робочий час вчителя для творчої підготовки до уроків та індивідуальної роботи з учнями. Прозорість оцінювання та оперативний доступ до результатів навчання сприяють підвищенню мотивації здобувачів освіти та покращують комунікацію між школою та родиною, що повністю відповідає концепції людиноцентризму в сучасній українській освіті.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гайн А. Г. Інформатика та інформаційні технології. Київ: Освіта, 2019. 302 с.
2. Глоба Л. С. Розробка інформаційних систем: навч. посіб. Київ: НТУУ «КПІ», 2018. 321 с.
3. Державна національна програма "Освіта" (Україна ХХІ століття). URL: <https://zakon.rada.gov.ua/laws/show/896-93-п>, 29.05.1996.
4. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Київ: ДП «УкрНДНЦ», 2016. 26 с.
5. Електронні класні журнали та щоденники. Міністерство освіти і науки України. URL: <https://mon.gov.ua/news/elektronnyi-zhurnal-u-shkoli-pro-holovne>, 06.09.2025.
6. Єдина школа: інформаційна система. URL: <https://eschool-ua.com>, 2025.
7. Інструкція з ведення ділової документації у закладах загальної середньої освіти в електронній формі: затв. Наказом Міністерства освіти і науки України № 707 від 08.08.2022. URL: <https://zakon.rada.gov.ua/laws/show/z1029-22>.
8. Ключко О. В. Математичне моделювання систем і процесів в освіті/педагогіці: навч. посіб. Вінниця: ВДПУ імені Михайла Коцюбинського, 2019. 127 с.
9. Концепція цифрової трансформації освіти і науки на період до 2026 року: схвалено розпорядженням Кабінету Міністрів України від 19 серпня 2021 р. № 988-р.
10. Концепція реалізації державної політики у сфері реформування загальної середньої освіти «Нова українська школа» на період до 2029 року: схв. розп. КМУ від 14.12.2016 р. № 988.
11. Литвин В. В. Бази даних та знань: підручник. Львів: Магнолія-2006, 2019. 368 с.
12. Морзе Н. В., Варченко-Троценко Л. О. Електронне освітнє середовище як інструмент забезпечення якості освітнього процесу. Інформаційні технології і засоби навчання. 2019. Т. 71, № 3. С. 235–249.

13. Мрія: освітня екосистема. URL: <https://mriia.gov.ua/app>, 2025.
14. Нові знання: електронні щоденники та журнали. URL: <https://nz.ua>, 2024.
15. Пасічник В. В. Організація баз даних та знань. Київ: BHV, 2020. 384 с.
16. Петкович В. М. Проектування баз даних: теорія та практика. Київ: Ліра-К, 2021. 280 с.
17. Про освіту: Закон України від 05.09.2017 № 2145-VIII. URL: <https://zakon.rada.gov.ua/laws/show/2145-19>.
18. Про повну загальну середню освіту: Закон України від 16.01.2020 № 463-IX. URL: <https://zakon.rada.gov.ua/laws/show/463-20>.
19. Співаковський О. В., Петухова Л. Є. Інформаційно-комунікаційні технології в початковій школі: навч.-метод. посібник. Херсон: ХДУ, 2018. 230 с.
20. Тарасенко А. В. Розробка клієнт-серверних застосунків на платформі .NET. Харків: ХНУРЕ, 2022. 156 с.
21. Федорчук А. Г. Системний аналіз в інформаційних технологіях. Житомир: ЖДТУ, 2018. 210 с.
22. Шевчук А. Патерн MVVM у WPF: професійний підхід до розробки. Програмування та комп'ютерні мережі. 2023. № 2. С. 45–52.
23. Albahari J. C# 10 in a Nutshell: The Definitive Reference. O'Reilly Media, 2022. 1060 p.
24. Ben-Gan I. T-SQL Fundamentals. 3rd Edition. Microsoft Press, 2016. 464 p.
25. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide. Addison-Wesley, 2018. 496 p.
26. Google Classroom. URL: <https://classroom.google.com>, 2025.
27. Harrington J. L. Relational Database Design and Implementation. 4th Edition. Morgan Kaufmann, 2016. 434 p.
28. Griffiths I. Programming WPF. 2nd Edition. O'Reilly Media, 2019. 886 p.
29. Moodle: Open Source Learning Platform. URL: <https://moodle.org>, 2025.
30. HUMAN: освітня система. URL: <https://www.human.ua>, 2023.

31. Microsoft Documentation. ADO.NET Architecture. URL: <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet>, 2025.
32. Microsoft Documentation. Windows Presentation Foundation (WPF). URL: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf>, 2025.
33. Nathan A. WPF 4.5 Unleashed. Sams Publishing, 2013. 864 p.
34. Price M. J. C# 10 and .NET 6 – Modern Cross-Platform Development. Packt Publishing, 2021. 820 p.
35. Ray B. School Management Systems: A Guide for Administrators. TechEd Publishing, 2020. 210 p.
36. Smith J. Advanced MVVM Techniques. New York: Apress, 2021. 300 p.
37. Troelsen A., Japikse P. Pro C# 10 with .NET 6. Apress, 2022. 1400 p.
38. Versluis G. Desktop Development with .NET. Microsoft Press, 2023. 350 p.
39. Visual Studio 2022 Documentation. URL: <https://visualstudio.microsoft.com>, 2025.
40. What are ACID properties in Database? GeeksforGeeks. URL: <https://www.geeksforgeeks.org/acid-properties-in-dbms>, 17.11.2025.



## ДОДАТОК А

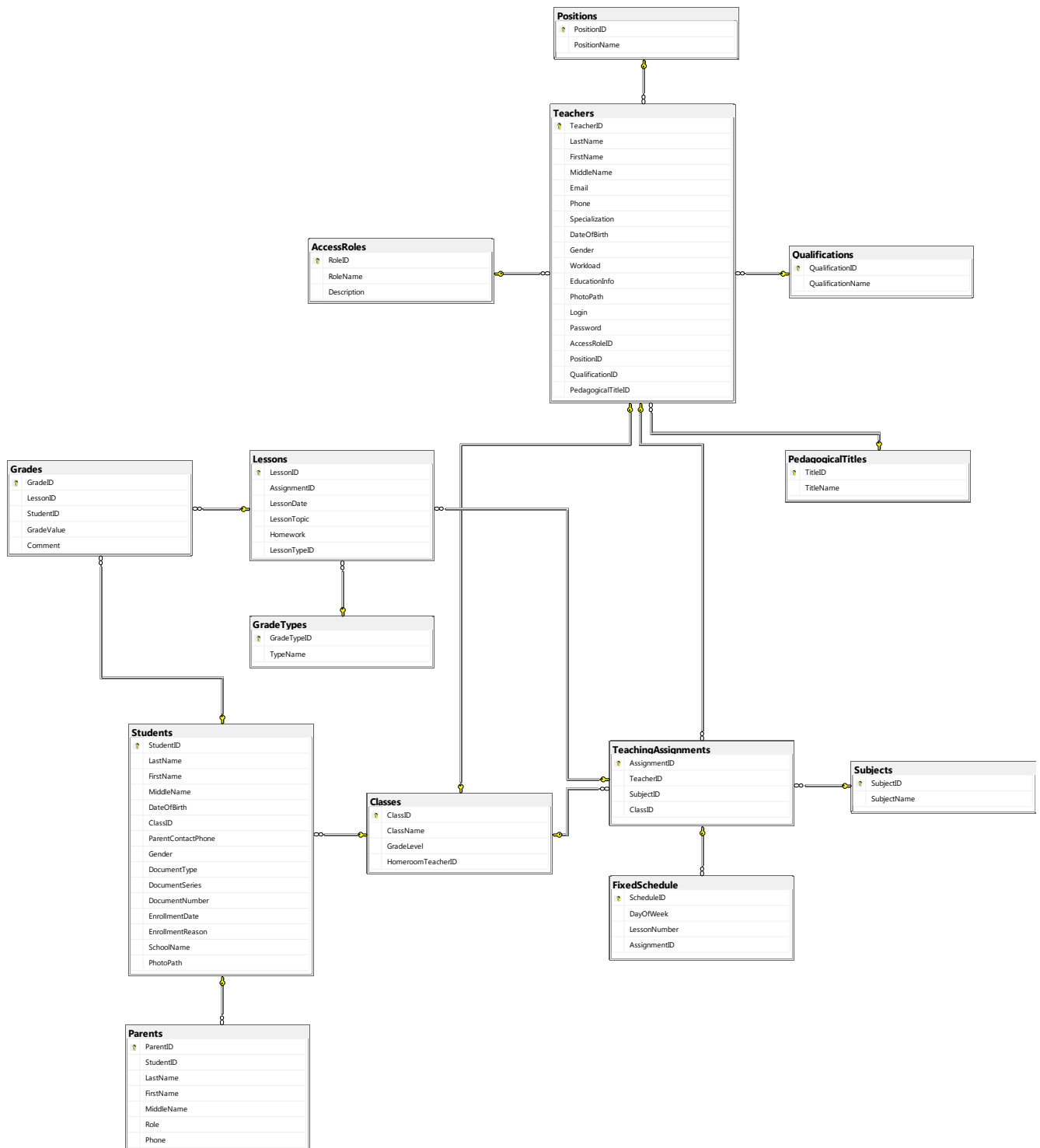


Рисунок А.1 – Повна інфологічна модель бази даних

## ДОДАТОК Б

### SQL-код створення структури бази даних

```

CREATE DATABASE DB_SchoolJournal;

USE DB_SchoolJournal;

CREATE TABLE AccessRoles (
    RoleID INT PRIMARY KEY,
    RoleName NVARCHAR(50) NOT NULL,
    Description NVARCHAR(255)
);

CREATE TABLE Positions (
    PositionID INT PRIMARY KEY IDENTITY(1,1),
    PositionName NVARCHAR(100) NOT NULL UNIQUE
);

CREATE TABLE Qualifications (
    QualificationID INT PRIMARY KEY IDENTITY(1,1),
    QualificationName NVARCHAR(100) NOT NULL UNIQUE
);

CREATE TABLE PedagogicalTitles (
    TitleID INT PRIMARY KEY IDENTITY(1,1),
    TitleName NVARCHAR(100) NOT NULL UNIQUE
);

CREATE TABLE GradeTypes (
    GradeTypeID INT PRIMARY KEY IDENTITY(1,1),
    TypeName NVARCHAR(50) NOT NULL
);

CREATE TABLE Announcements (
    AnnouncementID INT PRIMARY KEY IDENTITY(1,1),
    Content NVARCHAR(255) NOT NULL,
    DateCreated DATETIME DEFAULT GETDATE()
);

CREATE TABLE Teachers (
    TeacherID INT PRIMARY KEY IDENTITY(1,1),
    LastName NVARCHAR(50) NOT NULL,
    FirstName NVARCHAR(50) NOT NULL,
    MiddleName NVARCHAR(50),
    Email NVARCHAR(100) UNIQUE,
    Phone NVARCHAR(20),
    Specialization NVARCHAR(100),
    DateOfBirth DATE,
    Gender NVARCHAR(10) NOT NULL,
    Workload DECIMAL(5,2),
    EducationInfo NVARCHAR(MAX),
    PhotoPath NVARCHAR(MAX),
    Login NVARCHAR(50) NULL,
    Password NVARCHAR(MAX) NULL,
    AccessRoleID INT DEFAULT 3,
    PositionID INT NOT NULL,
    QualificationID INT,
    PedagogicalTitleID INT,
    FOREIGN KEY (PositionID) REFERENCES Positions(PositionID),
    FOREIGN KEY (QualificationID) REFERENCES Qualifications(QualificationID),
    FOREIGN KEY (PedagogicalTitleID) REFERENCES PedagogicalTitles(TitleID),
    FOREIGN KEY (AccessRoleID) REFERENCES AccessRoles(RoleID)
);

```

```

);

CREATE TABLE Subjects (
    SubjectID INT PRIMARY KEY IDENTITY(1,1),
    SubjectName NVARCHAR(100) NOT NULL UNIQUE
);

CREATE TABLE Classes (
    ClassID INT PRIMARY KEY IDENTITY(1,1),
    ClassName NVARCHAR(10) NOT NULL,
    GradeLevel INT NOT NULL,
    HomeroomTeacherID INT UNIQUE NOT NULL,
    FOREIGN KEY (HomeroomTeacherID) REFERENCES Teachers(TeacherID)
);

CREATE TABLE Students (
    StudentID INT PRIMARY KEY IDENTITY(1,1),
    LastName NVARCHAR(50) NOT NULL,
    FirstName NVARCHAR(50) NOT NULL,
    MiddleName NVARCHAR(50),
    DateOfBirth DATE,
    ClassID INT NOT NULL,
    ParentContactPhone NVARCHAR(20),
    Gender NVARCHAR(10),
    DocumentType NVARCHAR(50),
    DocumentSeries NVARCHAR(10),
    DocumentNumber NVARCHAR(20),
    EnrollmentDate DATE,
    EnrollmentReason NVARCHAR(200),
    SchoolName NVARCHAR(200) DEFAULT N'Хрпінівська філія Іллінецького ліцею №1',
    PhotoPath NVARCHAR(MAX),
    FOREIGN KEY (ClassID) REFERENCES Classes(ClassID)
);

CREATE TABLE Parents (
    ParentID INT PRIMARY KEY IDENTITY(1,1),
    StudentID INT NOT NULL,
    LastName NVARCHAR(50),
    FirstName NVARCHAR(50),
    MiddleName NVARCHAR(50),
    Role NVARCHAR(50),
    Phone NVARCHAR(20),
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID) ON DELETE CASCADE
);

CREATE TABLE TeachingAssignments (
    AssignmentID INT PRIMARY KEY IDENTITY(1,1),
    TeacherID INT NOT NULL,
    SubjectID INT NOT NULL,
    ClassID INT NOT NULL,
    FOREIGN KEY (TeacherID) REFERENCES Teachers(TeacherID),
    FOREIGN KEY (SubjectID) REFERENCES Subjects(SubjectID),
    FOREIGN KEY (ClassID) REFERENCES Classes(ClassID),
    UNIQUE (TeacherID, SubjectID, ClassID)
);

CREATE TABLE FixedSchedule (
    ScheduleID INT PRIMARY KEY IDENTITY(1,1),
    DayOfWeek INT,
    LessonNumber INT,
    AssignmentID INT NOT NULL,
    FOREIGN KEY (AssignmentID) REFERENCES TeachingAssignments(AssignmentID)
);

```

```
CREATE TABLE Lessons (
    LessonID INT PRIMARY KEY IDENTITY(1,1),
    AssignmentID INT NOT NULL,
    LessonDate DATE NOT NULL,
    LessonTopic NVARCHAR(255),
    Homework NVARCHAR(1000),
    LessonTypeID INT DEFAULT 1,
    FOREIGN KEY (AssignmentID) REFERENCES TeachingAssignments(AssignmentID),
    FOREIGN KEY (LessonTypeID) REFERENCES GradeTypes(GradeTypeID)
);
```

```
CREATE TABLE Grades (
    GradeID INT PRIMARY KEY IDENTITY(1,1),
    LessonID INT NOT NULL,
    StudentID INT NOT NULL,
    GradeValue NVARCHAR(3) NOT NULL,
    Comment NVARCHAR(255),
    FOREIGN KEY (LessonID) REFERENCES Lessons(LessonID),
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID)
);
```

```
INSERT INTO AccessRoles (RoleID, RoleName, Description) VALUES
(1, N'Адміністратор БД', N'Повний доступ'),
(2, N'Адміністратор Закладу', N'Директор/Завуч'),
(3, N'Вчитель', N'Доступ до своїх предметів');
```

```
INSERT INTO GradeTypes (TypeName) VALUES
(N'Поточна'), (N'Зошит'), (N'Самостійна робота'), (N'Контрольна робота'),
(N'Практична робота'), (N'Семестрова'), (N'Річна'), (N'Підсумкова'),
(N'Група результатів 1 (ГР1)'), (N'Група результатів 2 (ГР2)'),
(N'Група результатів 3 (ГР3)'), (N'Група результатів 4 (ГР4)');
```

```
INSERT INTO Subjects (SubjectName) VALUES
(N'Українська мова'), (N'Математика'), (N'Інформатика'), (N'Англійська мова'),
(N'Історія України'), (N'Фізика'), (N'Хімія'), (N'Біологія');
```

```
INSERT INTO Positions (PositionName) VALUES (N'Вчитель'), (N'Директор'),
(N'Заступник');
INSERT INTO Qualifications (QualificationName) VALUES (N'Вища категорія'),
(N'Спеціаліст');
```

ДОДАТОК В

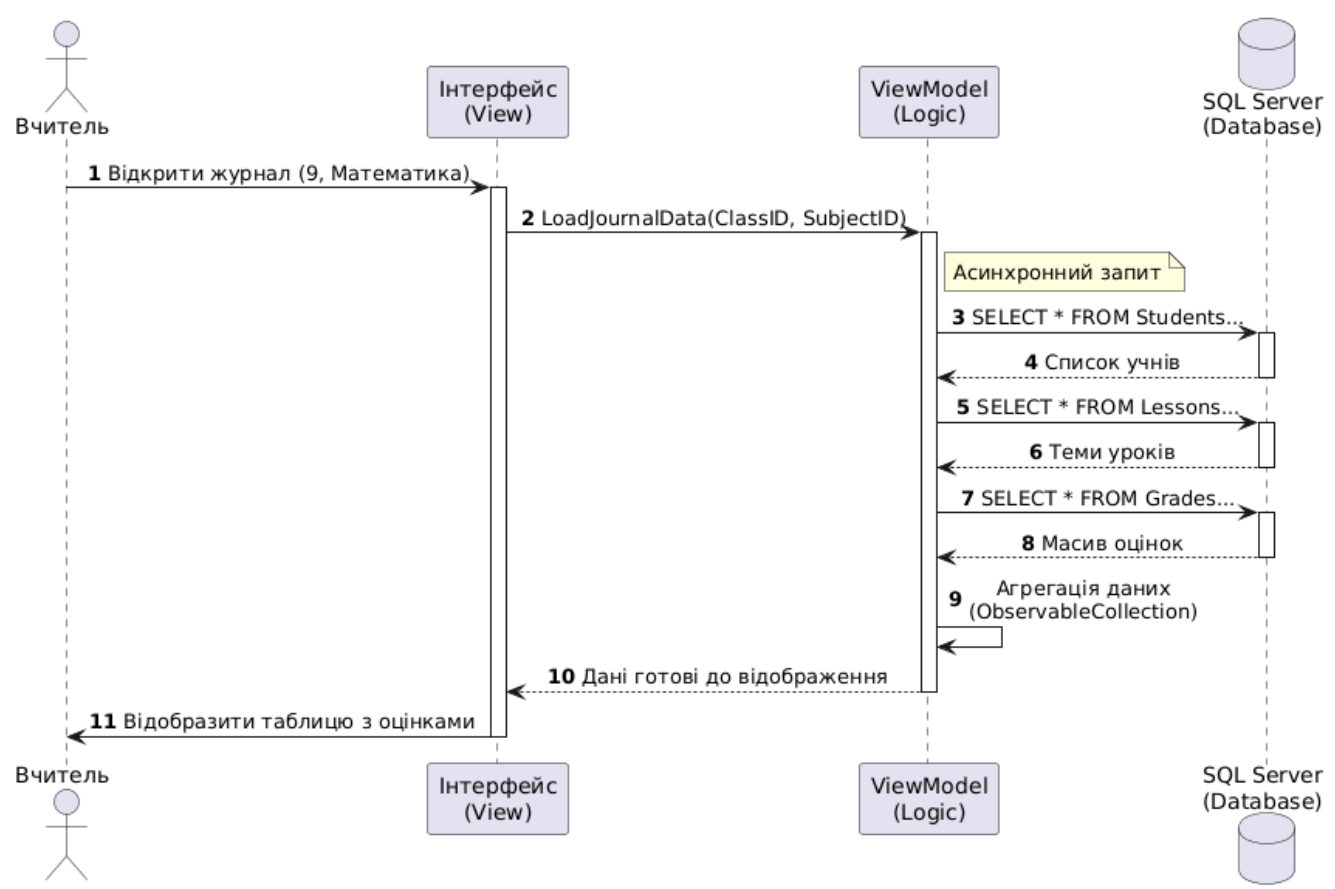


Рисунок В.1 – Діаграма послідовності: Завантаження даних

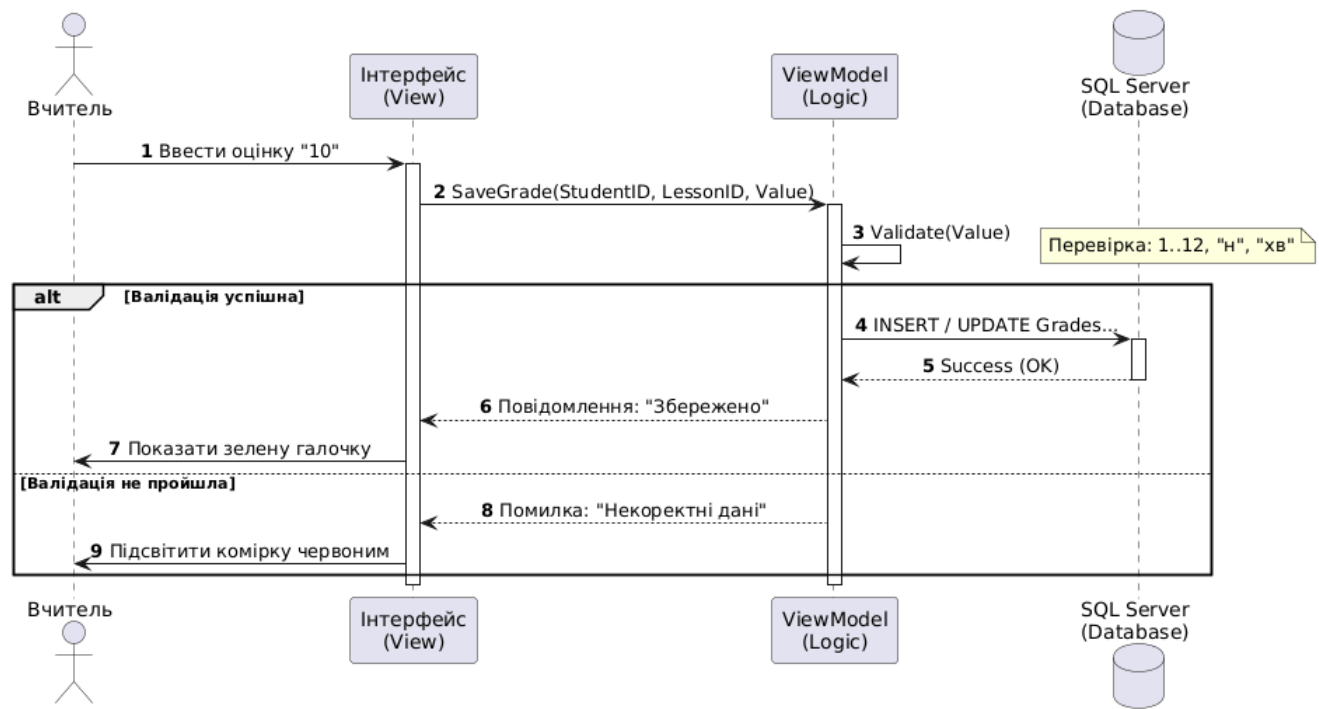


Рисунок В.2 – Діаграма послідовності: Збереження оцінки

## ДОДАТОК Г

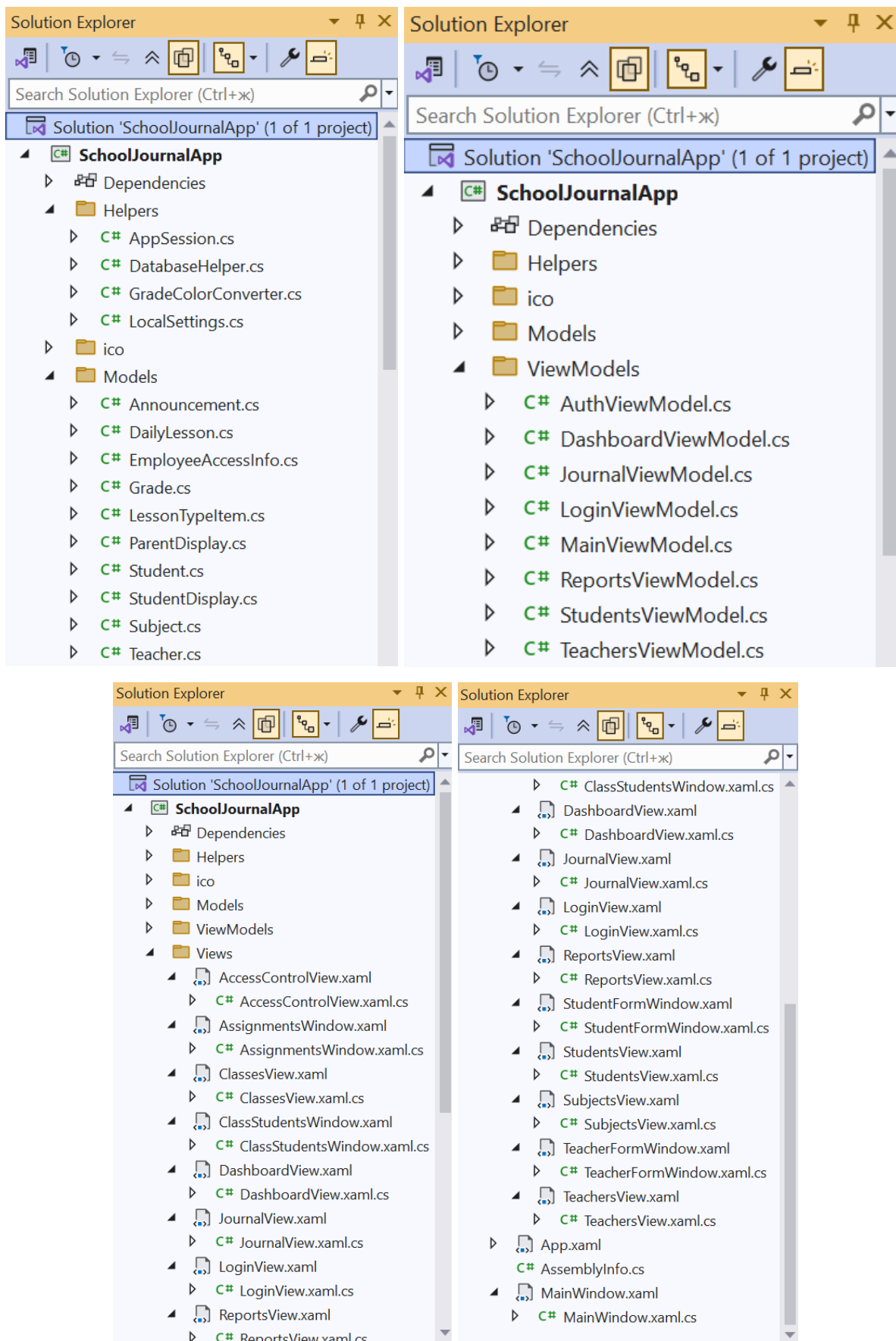


Рисунок Г.1 – Структура файлів проекту (Solution Explorer)