МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Бакалаврська кваліфікаційна робота

на тему: Розробка застосування працівників відділу кадрів
як додатку ІС «Особистий кабінет викладача»

Виконав студент 4 курсу групи К-41
Спеціальність 122 комп'ютерні науки
Аджамов Рамін Адалат огли

Керівник к.ф.-м.н., доцент
Козловська Валентина Петрівна

Консультант

Рецензент к.т.н., доцент
Великодний Станіслав Сергійович

Одеса 2020

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ 6
ВСТУП 7
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ 8
1.1 Користувачі інформаційної системи 8
1.2 Вимоги до інформаційної системи «Особистий кабінет викладача» 9
2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ «ОСОБИСТИЙ КАБІНЕТ ВИКЛАДАЧА» 11
3 ВИБІР СКБД ТА СЕРЕДОВИЩА РОЗРОБКИ КЛІЄНТСЬКОГО ДОДАТКУ 15
3.1 Реляційні бази даних 15
3.1.1 Мова T-SQL 16
3.1.2 Збережені процедури у мові T-SQL 17
3.2 Вибір СКБД для БД ІС «Особистий кабінет викладача» 18
3.3 Вибір середовища розробки додатку та мови програмування 19
4 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 22
5 СЕРВЕРНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ БД ІС «ОСОБИСТИЙ КАБІНЕТ ВИКЛАДАЧА» 25
6 ОПИС ЗАСТОСУВАННЯ ПРАЦІВНИКА КАФЕДРИ 31
6.1 Загальні відомості 31
6.2 Функціональне призначення 31
6.3 Керівництво програміста 31
6.3 Посібник користувача 33
7 ОПИС ЗАСТОСУВАННЯ ПРАЦІВНИКА ВІДДІЛУ КАДРІВ 38
7.1 Загальні відомості 38
7.2 Функціональне призначення 38
7.3 Керівництво програміста 38
7.3 Посібник користувача 40
ВИСНОВКИ 48
ПЕРЕЛІК ПОСИЛАНЬ 49

ДОДАТОК А Кадрові вимоги 51
ДОДАТОК Б Лист самоатестації викладача 54
ДОДАТОК В Логічна схема бази даних ІС 58
ДОДАТОК Г Вихідний код збережених процедур БД 59
ДОДАТОК L Вихідний код клієнтського додатку 61

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних

ІС – інформаційна система

ПЗ – програмне забезпечення

РБД – реляційна база даних

СКБД – система керування базами даних

ВСТУП

Основне призначення закладів вищої освіти – надання освітніх послуг здобувачам освіти рівня бакалавр та магістр. Крім навчання студентів у відповідності до навчальних планів викладачі університету повинні вести наукову, методичну, виховну і організаційну роботу, а також керувати науковою роботою студентів.

Всі види діяльності викладача враховуються при визначенні його рейтингу, також наукова, організаційна, методична діяльність викладача визначає, наскільки він відповідає ліцензійним умовам, що накладаються на викладацький склад при отриманні вишем акредитації.

Для спрощення звітності викладачів щодо своєї різнобічної діяльності, та скорочення часу та зусиль по обробці даних і отриманні балів рейтингу викладацького складу університету, а також списків викладачів з переліком виконаних ними пунктів ліцензійних умов, потрібно розробити та впровадити інформаційну систему (ІС) «Особистий кабінет викладача».

Метою даної кваліфікаційної роботи є моделювання та розробка підсистеми «Група користувачів – відділ кадрів» у складі інформаційної системи «Особистий кабінет викладача».

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

В ліцензійних вимогах оцінюється наукова та громадська активність викладачів. Методика оцінки цієї активності складається з перевірки кількості пунктів кадрових вимог, які виконав викладач за останні 5 років. Серед цих вимог є пункти, що оцінюють діяльність викладача на науковому, методичному та організаційному напрямках [1], [2] [1)].

1.1 Користувачі інформаційної системи

Для роботи ІС «Особистий кабінет викладача» потрібно, щоб викладачі вносили свої дані щодо всіх видів діяльності у базу даних (БД) інформаційної системи. Таким чином, викладачі є однією з груп основних користувачів ІС, що розробляється. Ця група користувачів повинна мати права на повне маніпулювання: додавання, перегляд, оновлення, видалення, – своїми особистими даними, які відносяться до професійної діяльності. Тобто, викладачі повинні бути авторизованими користувачами.

Щоб викладача отримав відповідні права, його потрібно внести у групу користувачів «Викладачі». Тобто потрібен користувач, який вводить в систему дані: прізвище, ім’я, по-батькові, науковий ступінь та наукове звання, де і коли отримав вищу освіту, де і на протязі якого періоду працював за фахом, на якій посаді працює та на яку частину ставки, тощо.

Як видно з перелічених даних, веденням цих даних на підприємствах займаються відділи кадрів. Але навантажувати робітників відділу кадрів роботу з ІС «Особистий кабінет викладача» не є оптимальним.

Внесення у БД кадрових даних викладачів можна розбити на декілька етапів:

[1)] [1] Про затвердження Ліцензійних умов провадження освітньої діяль-ності закладів освіти. Постанова КМУ № 1187 від 30.12.15 року. http://mdu.edu.ua/wpcontent/-uploads/postanova%E2%84%961187.pdf

[2] Про затвердження Ліцензійних умов провадження освітньої діяльності. Документ 1187-2015-п, чинний, поточна редакція – Редакція від 23.05.2018, підстава – 347-2018-п. https://zakon.rada.gov.ua/laws/show/1187-2015-%D0%BF

1. Відповідальний користувач кафедри вносить в БД всіх викладачів відповідної кафедри: прізвище, ім’я, по-батькові, науковий ступінь та наукове звання на якій посаді працює та на яку частину ставки;
2. Кожен викладач додає про себе іншу кадрову інформацію: де і коли навчався, які ступені та звання має;
3. Відповідальний користувач відділу кадрів додає у БД дані з трудової книжки, яка зберігається у відділі кадрів, а саме, де, на яких посадах, та протягом якого терміну викладач працював.

Внесення даних, які становлять ліцензійні вимоги та пункти таблиці розрахунку рейтингу відбувається при проектуванні та впровадження ІС «Особистий кабінет викладача». При експлуатації інформаційної системи правами оновлення цих даних має володіти відповідальний користувач методичного відділу університету. Викладачі повинні мати право переглядати пункти ліцензійних вимог та таблиці розрахунку рейтингу, а також вносити дані щодо виконання цих пунктів з прив’язкою своїх даних до пунктів ліцензійних вимог та/або пунктів таблиці рейтингу.

1.2 Вимоги до інформаційної системи «Особистий кабінет викладача»

ІС «Особистий кабінет викладача» повинна:

- надавати можливість вносити у БД нових викладачів університету;
- надавати можливість відповідальним користувачам вносити зміні у кадрові дані викладачів: посаду, частину ставки, звання, тощо;
- надавати всім користувачам можливість перегляду пунктів кадрових вимог та таблиці розрахунку рейтингу викладачів;
- надавати викладачеві зручний інтерфейс для внесення та перевірки його особистих даних;
- надавати можливість оцінки викладачем його відповідності до кадрових вимог (ліцензійних умов);
- надавати можливість оцінки викладачем його рейтингу;

- надавати користувачам, що перевіряють особисті дані викладачів з різних видів діяльності, зручний інтерфейс для контролю;
- надавати користувачеві, що відповідає за повноту та актуальність даних щодо кадрових вимог зручний інтерфейс для перегляду пунктів цих вимог та їх оновлення у разі необхідності;
- надавати користувачеві, що відповідає за розрахунок рейтингу викладачів зручний інтерфейс для перегляду пунктів таблиці розрахунку рейтингу, а також можливість у разі необхідності оновлення цих пунктів.

2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ «ОСОБИСТИЙ КАБІНЕТ ВИКЛАДАЧА»

Для розробки моделі інформаційної системи потрібно спочатку визначити групи користувачів, що будуть працювати з нею. Потрібно виявити, у яких випадках користувачі різних груп будуть звертатись до даної інформаційної системи, та за якими сценаріями вони будуть з нею працювати. Потім потрібно виконати побудову моделі системи, що розробляється [3]–[4] [1)].

Потім потрібно визначитись, чи є необхідним використання бази даних (БД) та системи керування базами даних (СКБД). Якщо база даних потрібна, треба визначити модель даних, яка найкраще буде описувати предметну область інформаційної системи, що розробляється. Потім потрібно розробити концептуальну, логічну та фізичну моделі бази даних.

Основною групою користувачів інформаційної системи «Особистий кабінет викладача» будуть викладачі закладу вищої освіти. Крім них дану ІС будуть використовувати працівники методичного відділу університету, які повинні визначати критерії оцінки роботи викладача та перевіряти звіти викладачів щодо методичної роботи.

Крім того, користувачами ІС, що проектується, будуть відповідальні працівники інших підрозділів, що будуть оцінювати різні види роботи викладача: наукову, організаційну, виховну, громадську.

У першому розділі обґрунтовується необхідність внесення у систему групи користувачів «Відповідальний користувач кафедри» та «Співробітник відділу кадрів»

Розглянемо, які варіанти використання системи потрібні групі користувачів «Відповідальний користувач кафедри»:

[1)] [3] Вендров А. М. CASE-технологии. Современные методы и средства проектирования информационных систем. М: «Финансы и статис-тика», 1998. – 98 с.

[4] Куликов Г. Г., Никулина Н. О., Речкалов А. В. Управление проекта-ми на основе системного моделирования: Учебное пособие. Уфа: УГАТУ, 2009. – 171 с.

а) переглянути список викладачів кафедри: прізвище, ім’я, по-батькові, ступінь, звання, посаду, ставку;

б) оновити при необхідності дані з попереднього пункту щодо викладачів своєї кафедри;

в) видалити дані викладача, що звільнився;

г) переглянути, чи вносили викладачі кафедри дані про всі види своєї роботи за звітний період.

Розглянемо, які варіанти використання системи потрібні групі користувачів «Співробітник відділу кадрів»:

а) переглянути список викладачів всіх кафедри: прізвище, ім’я, по-батькові, ступінь, звання, посаду, ставку;

б) відредагувати у разі помилки дані з попереднього пункту щодо викладачів всіх кафедр;

в) внести в БД дані про роботу викладачів за фахом;

г) переглянути, чи відповідають дані попереднього пункту тим даним, що внесли викладачі про свою роботу за фахом для забезпечення виконання пункту 17 ліцензійних вимог.

Варіанти використання ІС групою користувачів «Викладач» у контексті їх реєстрації в системі та наданні своїх особистих даних:

а) переглянути список викладачів кафедри: прізвище, ім’я, по-батькові, ступінь, звання, посаду, ставку;

б) переглянути, які є в БД дані про роботу викладача за фахом;

в) визначити, чи попередня робота за фахом забезпечує виконання пункту 17 ліцензійних вимог.

Як видно з порівняння наведених варіантів використання ІС вказаними 3 групами користувачів, всі ці групи мають один спільний варіант використання – варіант №1. Таким чином цей варіант використання повинен відноситись до загального користувача, а перераховані 3 групи користувачів

є спадкоємцями цього загального користувача, що мають додаткові варіанти використання системи.

На рис. 1 наведена діаграми варіантів використання (use case diagram) інформаційної системи «Особистий кабінет викладача» для груп користувачів «Відповідальний користувач кафедри», «Співробітник відділу кадрів», «Викладач».

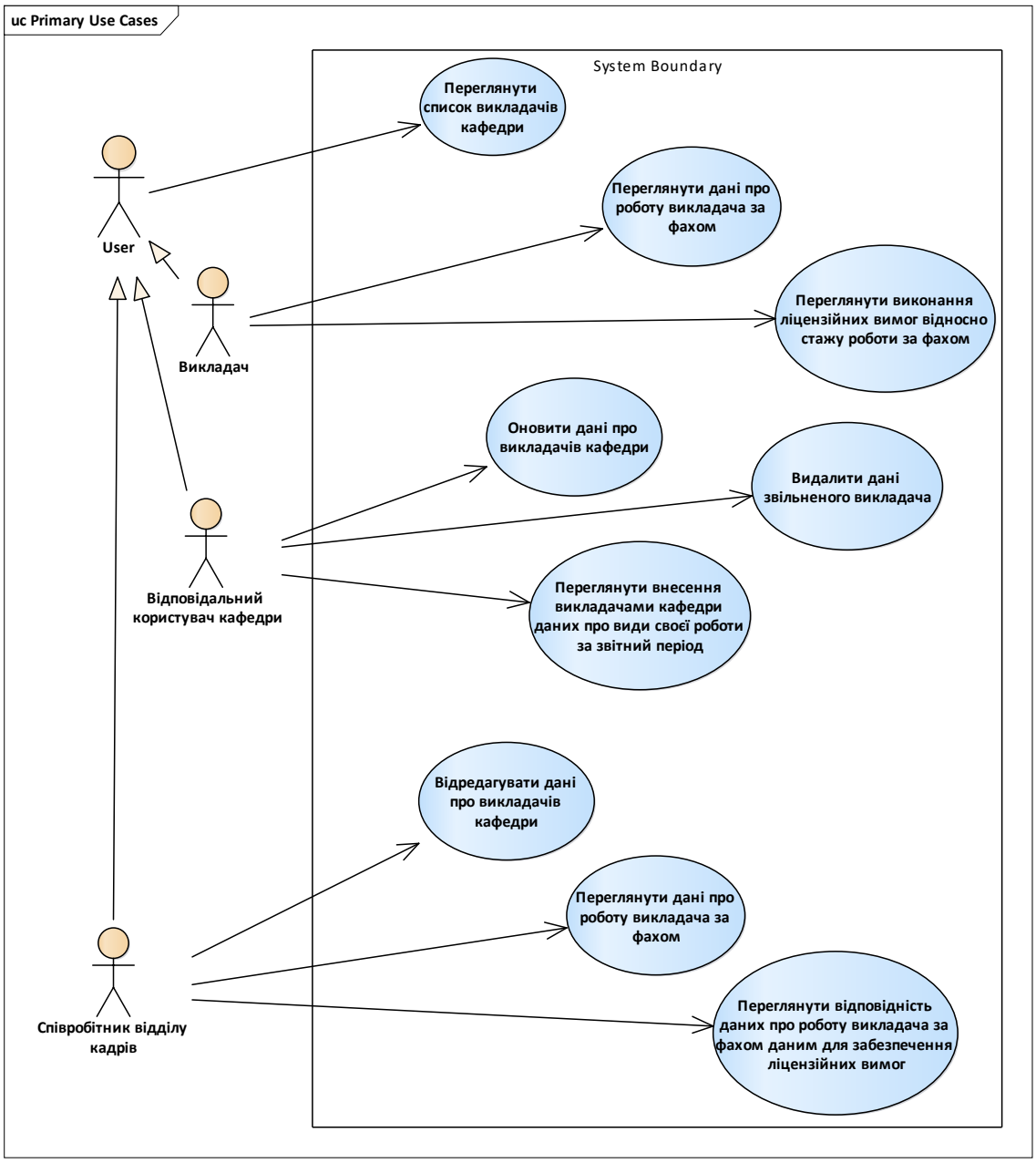

Рисунок 1 – Діаграма варіантів використання інформаційної системи «Особистий кабінет викладача»

Для кожного варіанту використання ІС потрібно розробити сценарій співпраці користувача з системою. Складні сценарії – більше 3-4 пунктів – бажано моделювати з використанням діаграм активності.

Варіанти використання «переглянути список викладачів кафедри: прізвище, ім’я, по-батькові, ступінь, звання, посаду, ставку» та «оновити особисті дані викладачів» не вимагають складних сценаріїв. У базі даних ІС повинна бути базова таблиця, що містить список викладачів. Всі вказані у першому варіантів використання атрибути викладача повинні бути присутніми у цій таблиці. Для можливості визначення до якої кафедри належить викладач, у БД повинна бути базова таблиця списку кафедр університету, а у викладача повинен бути атрибут – посилання на відповідну кафедру.

3 ВИБІР СКБД ТА СЕРЕДОВИЩА РОЗРОБКИ КЛІЄНТСЬКОГО ДОДАТКУ

Оскільки ІС повинна зберігати великий об’єм даних, що будуть постійно оновлюватись, її бажано розробляти на основі бази даних. Перед проектуванням БД потрібно визначити, яку модель даних зручно використовувати для проекту, що розробляється.

На наш час найбільш популярними є СКБД засновані на одній з наступних моделей даних: реляційна, об’єктно-орієнтована, документо-орієнтована [5]–[6] [1)].

База даних ІС «Особистий кабінет викладача» проектувалась та розроблялась у комплексної магістерської роботи 2019 року. Було прийняте рішення реалізовувати БД на основі реляційної моделі даних з використанням реляційної СКБД MS SQL Server.

3.1 Реляційні бази даних

Однією з головних переваг реляційних систем керування базами даних (РСКБД) над системами, заснованими на інших моделях даних є мова структурованих запитів SQL (Structured Query Language). Мову SQL називають мовою реляційних баз даних.

Завдяки цій мові можна виконувати вибірки з баз даних як завгодно складної структури. Мова SQL дозволяє знаходити узагальнені (агреговані) дані, які використовуються, наприклад, в корпоративних сховищах даних.

Переваги SQL:

1) Наявність міжнародних стандартів.

[1)] [5] Реляционные БД vs Объектно-ориентированные БД. URL: https:// habr.com/ru/post/93356 (Дата звернення 08.12.2019)

[6] Объектно-ориентированные базы данных – основные концепции, организация и управление: краткий обзор URL: http://www.cyberguru.ru/ database/database-theory/object-oriented-db-overview.html

2) Незалежність від конкретної СКБД. Незважаючи на наявність діалектів і відмінностей в синтаксисі, в більшості своїй тексти SQL-запитів можуть бути досить легко перенесені з однієї СКБД в іншу.
3) Підтримка архітектури клієнт-сервер.
4) Декларативність. За допомогою SQL програміст описує тільки те, які дані потрібно витягнути або модифікувати. Яким чином це зробити, вирішує СКБД безпосередньо при обробці SQL-запиту.
5) Поширеність.
6) Швидке навчання.

Незважаючи на наявність стандартів, практично в кожній СКБД застосовується свій діалект мови.

3.1.1 Мова T-SQL

Мова T-SQL – це власний діалект мови структурованих запитів, який застосовується в СКБД SQL Server. Мова T-SQL призначений виключно для роботи з СКБД SQL Server, але основна частина операторів, що застосовується у ній, має більш широке використання. Мова T-SQL сумісна зі стандартом ANSI SQL-92 на початковому рівні. Це означає, що в певному обсязі T-SQL сумісна з дуже широким відкритим стандартом. Слід враховувати, що в кожній реляційної СКБД застосовуються різні розширення і способи підвищення продуктивності, що доповнюють зазначений стандарт ANSI і виходять за його рамки. Основна відмінність між SQL та T-SQL є те, що SQL – це мова, яка використовується для зберігання і управління даними в СRБД, а T-SQL є розширеною версією SQL і призначена для виконання операцій на сервері MS SQL.

Додатки можуть зв'язуватися з SQL Server, виконуючи оператори T-SQL. Розробник може писати запити для виконання операцій над таблицями, об'єднання таблиць і додавання обмежень. Він також може виконувати транзакції, писати збережені процедури, подання, індекси і багато іншого.

Існують різні числові, рядкові функції, функції дати. Крім того, існують функції агрегування для виконання операції з набором значень.

При підготовці випуску СКБД SQL Server 2005 мова T-SQL була в значній мірі доопрацьована, і в неї були додані багато нових програмних конструкцій. Крім усього іншого, вона була перетворена в мову, сумісну із загальним середовищем виконання (Common Language Runtime CLR) операційної системи Windows; таким чином, починаючи з цього випуску T-SQL стала однією з мов .NET.

В усіх версіях СКБД SQL Server, починаючи з версії SQL Server 2005, передбачена можливість використовувати для доступу до бази даних будь-яку мову .NET, але в кінцевому підсумку в основі звернення безпосередньо до самих даних завжди лежить якийсь код SQL, тому мова T-SQL залишається базовою мовою при виконанні будь-яких дій в СКБД SQL Server. З кожним випуском SQL Server багато змінюється, але найбільш фундаментальні оператори доступу до даних є практично такими ж, як і раніше.

T-SQL – це процедурне розширення мови SQL компаній Microsoft. SQL була розширена такими додатковими можливостями як:

- керуючі оператори;
- локальні і глобальні змінні;
- різні додаткові функції для обробки рядків, дат, математики тощо;
- підтримка аутентифікації Microsoft Windows.

Мова T-SQL є ключем до використання SQL Server. Всі додатки, які взаємодіють з екземпляром SQL Server, незалежно від їх реалізації і інтерфейсу користувача, відправляють серверу інструкції T-SQL.

3.1.2 Збережені процедури у мові T-SQL

Збережені процедури і функції представляють собою набір SQL-операторів, які можна зберігати на сервері. Якщо сценарій збережений на

сервері, то клієнтам не доведеться повторно ставити одні й ті ж окремі оператори, замість цього вони зможуть звертатися до збереженій процедурі.

Ситуації, коли збережені процедури особливо корисні [7] – [8] [1)]:

- Численні клієнтські програми написані на різних мовах або працюють на різних платформах, але повинні виконувати однакові операції з базами даних.
- Безпека грає першорядну роль. Збережені процедури використовуються для всіх стандартних операцій, що забезпечує сумісність і безпеку середовища, а процедури гарантують належну реєстрацію кожної операції. При такому типі установки додатка, і користувачі не отримують безпосередній доступ до таблиць бази даних і можуть виконувати тільки конкретні процедури, що зберігаються.
- Необхідно знизити мережевий трафік між клієнтом і сервером. Обсяг інформації, що пересилається між сервером і клієнтом істотно знижується, але збільшується навантаження на систему сервера баз даних, так як в цьому випадку на стороні сервера виконується велика частина роботи по обробці даних.

3.2 Вибір СКБД для БД ІС «Особистий кабінет викладача»

У базі даних ІС, що розробляється, будуть зберігатись факти: назви трудів викладачів: статей, підручників, методичних вказівок; рік видання; кількість сторінок; тощо. Тому треба використовувати фактографічну реляційну СКБД.

[1)] [7] Хранимые процедуры. URL: https://metanit.com/sql/sqlserver/11.1.php (Дата звернення 10.05.2020)

[8] CREATE PROCEDURE (Transact-SQL). Microsoft. Документация по SQL. URL: https://docs.microsoft.com/ru-ru/sql/t-sql/statements/create-procedure- transact-sql?view=sql-server-ver15 (Дата звернення 10.05.2020)

Реляційну модель даних слід вибрати також з інших міркувань. Серед пунктів ліцензійних вимог та оцінки рейтингу викладача є такі, що враховують деякій вид діяльності, якщо він представленим декількома даними, наприклад, не менше трьох виданих методичних посібників за 5 останніх років. Тому для підрахунку рейтингу викладача та кількості пунктів ліцензійних вимог, що він виконав, потрібні не тільки детальні дані, а також узагальнені, агреговані дані. Такі дані легко отримати запитами з групуванням, які є складовою частиною мови SQL реляційних СКБД.

Для фізичної реалізації БД була реляційна СКБД MS SQL Server 2017.

3.3 Вибір середовища розробки додатку та мови програмування

На наш час існує багато середовищ швидкої розробки додатків до баз даних, але одним з найбільш вдалих є інтегроване середовище Visual Studio компанії Microsoft, особливо у випадку, коли розробляється додаток до бази даних, що розроблена на СКБД MS SQL Server. Використовуючи середовище Visual Studio програміст може розробляти програму декількома мовами програмування, але у випадку розробки додатку до баз даних найбільш зручним рішенням буде застосування мови C# [9] – [10] [1)].

На сьогоднішній момент мова програмування C# одна з найпотужніших і затребуваних мов в IT-галузі, яка швидко розвивається. На даний момент на C# пишуться найрізноманітніші програми: від невеликих десктопних програмок до великих веб-порталів і веб-сервісів, які обслуговують щодня мільйони користувачів.

У порівнянні з іншими мовами C# досить молода, але в той же час вона вже пройшла великий шлях. Перша версія мови вийшла разом з релізом

[1)] [9] Полное руководство по языку программирования C# 8.0 и платфор-ме .NET Core 3. URL: https://metanit.com/sharp/tutorial/(Дата звернення 10.05.2020)

[10] Руководство по программированию на C#. URL: https://docs.microsoft.com/ ru-ru/dotnet/csharp/programming-guide/ (Дата звернення 10.05.2020)

Microsoft Visual Studio .NET в лютому 2002 року. Поточною версією мови є версія C# 8.0, яка вийшла у вересні 2019 року разом з релізом .NET Core 3.

Середовище Microsoft Visual Studio має потужні можливості для взаємодії з базами даних. Використання наборів даних (DataSet) дозволяє зручно працювати з базою даних, переглядати вміст таблиць, виконувати вибірку даних та за допомогою збережених процедур виконувати обробку даних у БД.

Фреймворк .NET представляє потужну платформу для створення додатків. Можна виділити наступні її основні риси:

– Підтримка декількох мов. Основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), завдяки чому .NET підтримує кілька мов: поряд з C # це також VB.NET, C ++, F #, а також різні діалекти інших мов, прив'язані до .NET, наприклад, Delphi. NET. При компіляції код на будь-якому з цих мов компілюється в збірку спільною мовою CIL (Common Intermediate Language) - свого роду асемблер платформи .NET. Тому ми можемо зробити окремі модулі однієї програми на окремих мовах.

– Кросплатформність. .NET є платформою, що переноситься (з деякими обмеженнями). Наприклад, остання версія платформи на даний момент .NET Core підтримується на більшості сучасних ОС Windows, MacOS, Linux. Використовуючи різні технології на платформі .NET, можна розробляти програми на мові C# для самих різних платформ –Windows, MacOS, Linux, Android, iOS, Tizen.

Потужна бібліотека класів. .NET представляє єдину для всіх підтримуваних мов бібліотеку класів. І який додаток ми не збиралися б писати на C# – текстовий редактор, чат або складний веб-сайт – так чи інакше ми задіємо бібліотеку класів .NET.

Різноманітність технологій. Загальномовне середовище виконання CLR і базова бібліотека класів є основою для цілого стека технологій, які розробники можуть задіяти при побудові тих чи інших додатків. Наприклад,

для роботи з базами даних в цьому стеку технологій призначена технологія ADO.NET і Entity Framework Core. Для побудови графічних додатків з багатим насиченим інтерфейсом – технологія WPF і UWP, для створення більш простих графічних додатків – Windows Forms. Для розробки мобільних додатків – Xamarin. Для створення веб-сайтів – ASP.NET, тощо.

Також ще слід відзначити таку особливість мови C# і фреймворка .NET, як автоматичне прибирання сміття. А це означає, що нам в більшості випадків не доведеться, на відміну від C++, піклуватися про звільнення пам'яті. Вищезазначена загальномовного середовища CLR сама викличе збирач сміття і очистить пам'ять.

Варто відзначити, що .NET довгий час розвивався головним чином як платформа для Windows під назвою .NET Framework. В 2019 вийшла остання версія цієї платформи – .NET Framework 4.8. Вона більше не розвивається

З 2014 року Microsoft став розвивати альтернативну платформу – .NET Core, яка вже призначалася для різних платформ і повинна була увібрати в себе всі можливості застарілого .NET Framework і додати нову функціональність. Тому слід розрізняти .NET Framework, який призначений переважно для Windows, і кросплатформений .NET Core.

Нерідко додаток, створене на C#, називають керованим кодом (managed code). Це означає, що даний додаток створений на основі платформи .NET і тому керується загальномовним середовищем CLR, яке завантажує додаток і при необхідності очищує пам'ять. Але є також додатки, наприклад, створені на мові C ++, які компілюються не в спільну мову CIL, як C# або VB.NET, а в звичайний машинний код. В цьому випадку .NET не керує додатком.

Для розробки програмного застосування для викладачів обирається середовище Microsoft Visual Studio 2019 та мова програмування C# 8.0.

4 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Інформаційна система «Особистий кабінет викладача» повинна містити всі дані, необхідні для розрахунку рейтингу викладача та визначення кількості пунктів ліцензійних умов, яким відповідає діяльність викладача за останні 5 років. Потрібно проаналізувати, які пункти ліцензійних умов та таблиці розрахунку рейтингу викладача можуть перевірятись співробітником відділу кадрів; також потрібно визначити, чи всі дані для розрахунку цих пунктів можуть зберігатись у базі даних наявної схеми. Можливо, потрібно уточнити схему за рахунок додавання нових атрибутів у існуючі відношення, або додавання нових відношень.У відділі кадрів зберігається інформація щодо всього трудового шляху кожного співробітника. У трудовій книжці, оригінал якої зберігається у відділу кадрів, записуються всі місця, де працівник працював на постійній основі, на якій посаді працював, та термін праці – з якої дати прийнятий на роботу, та коли звільнений.

Також у особистій справі працівника зберігаються копії інших документів, що мають вплив на трудову діяльність працівників: документи про освіту, дипломи про присвоєння наукового ступеня та вченого звання, тощо. Копії зберігаються при підтвердження наявності оригінала документа.

Якщо проаналізувати всі 18 пунктів ліцензійних умов, можна зробити висновок, що до сфери відповідальності відділу кадрів відноситься лише один пункт: «17) досвід практичної роботи за спеціальністю не менше п’яти років». У цьому пункті не сказано, що повинна бути безперервна робота протягом п'яти років, головне, щоб за весь час трудової діяльності у працівника набралося не менше 5 років роботи за фахом. Тобто, для виконання цього пункту ліцензійних умов викладачеві потрібно внести в базу даних одне чи більше місце роботи за фахом з вказівкою дати прийому на роботу та дати звільнення. Сумарний стаж роботи за фахом буде розраховувати вибірка даних з групування та сумуванням термінів роботи у різних місцях.

У загальному випадку викладач може набрати необхідні 5 років трудового стажу за фахом, працюючи в кількох місцях у різний час. Тобто, потрібен новий тип сутності РоботаВикладача. Тип сутності викладач зв'язаний з типом сутності РоботаВикладача типом зв'язку «один-до-багатьох» [11] – [12] [1)].

У типу сутності РоботаВикладача повинні бути атрибути:

- ідентифікатор викладача (зовнішній ключ);
- місце роботи;
- посада;
- дата прийому;
- дата звільнення;
- чи є ця робота за фахом.

Дані у таблицю, як і інші особисті дані, додає сам викладач. Але перевіряти ці дані має співробітник відділу кадрів. Співробітник відділу кадрів не зобов'язаний змінювати або видаляти некоректні дані, він тільки повинен відмітити, чи відповідають вони дійсності.

Таким чином, у відношенні РоботаВикладача повинен бути ще один атрибут: «Підтверджено».

Перелічені вище 6 атрибутів заповнює викладач, але він не може задавати або змінювати значення атрибуту «Підтверджено».

Серед пунктів розрахунку рейтингу викладача також є пункти, перевірка виконання яких також відноситься до сфери відповідальності відділу кадрів. Це пункти 2.1 та 2.2 розділу «Підготовка науково-педагогічних кадрів»:

2.1 Захист у звітному календ. році докторської дисертації

2.2 Захист у звітному календ. році кандидатської дисертації

[1)] [11] Томас Конноли, Каролин Бегг. Базы данных. Проектирование, реа-лизация и сопровождение. Теория и практика. 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 1440 с.

[12] Дейт К., Дж. Введение в системы баз данных. 6-е издание: Пер. с англ. – К.; М.; СПб.: Издательский дом «Вильямс». 2000. – 848 с.

Захист дисертації враховується лише один раз у тому році, коли власне відбувся захист. Тому окремої таблиці для зберігання даної інформації створювати не потрібно. Потрібно додати у відношення Викладач атрибути:

– науковий ступінь, що присвоєний після захисту дисертації;
– дата захисту.

Ці дані вносить у базу даних викладач.

Для можливості перевірки даних співробітником відділу кадрів також до відношення Викладач потрібно додати атрибут «Підтверджено».

Додавати нові дані у відношення Викладач, тобто нових викладачів у базу даних, повинні відповідальні співробітники кафедр. Вони вносять основні дані: прізвище, ім’я, по-батькові, посада, – щодо викладачів своєї кафедри. Також за допомогою відповідального співробітника кафедри відбувається авторизація викладачів. Для авторизації викладач повинен внести у базу даних свій логін і пароль. Для виконання цих дій в ІС «Особистий кабінет викладача» повинне бути застосування співробітника кафедри.

Зареєстрований викладач може вносити у БД свої особисті дані. Ці дані може змінювати лише сам викладач. Викладач додає, оновлює і видаляє власні дані за допомогою застосування викладача.

Але для контролю правильності даних викладача у системі потрібні застосування для працівників всіх підрозділів, які перевіряють звіти про виконану викладачами роботу відповідного напрямку. Для перевірки даних, що стосуються працівників відділу кадрів, також потрібне відповідне застосування. Користувач застосування для відділу кадрів не повинен мати можливість додавати або видаляти записи в таблицях, що містять особисті дані викладачів. Цей користувач має тільки перевіряти вже внесені викладачем у базу даних особисті дані та ставити на них відмітку «підтверджено» або «не підтверджено».

5 СЕРВЕРНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ БД ІС «ОСОБИСТИЙ КАБІНЕТ ВИКЛАДАЧА»

В інформаційній системі «Особистий кабінет викладача» має бути велика кількість користувачів: викладачів та відповідальних осіб різних підрозділів, – які мають права маніпулювати даними в БД за допомогою офісних застосувань. У подібних випадках бажано, щоб офісні клієнтські застосування лише надавали користувачам необхідний інтерфейс, а вся обробка даних відбувалася на стороні сервера.

Тобто клієнтські застосування повинні бути «тонкими клієнтами». Для «тонких» клієнтів не бажано генерувати SQL-код команд на додавання, оновлення, усунення записів у таблицях БД у самому додатку. Ці клієнтські застосування повинні лише передавати до серверу виклик збережених процедур, які містять необхідний SQL-код і будуть виконувати потрібні дії по маніпулюванню даними у таблицях БД.

Потрібно розробити збережені процедури для додавання, оновлення та видалення записів в таблицях бази даних, які містять відомості про викладачів та про їх попередні місця роботи за фахом.

Збережена процедура AddTeacher додає запис у таблицю Teachers:

```
CREATE PROC AddTeacher
@id_dep smallint,
@id_teach smallint output,
@surname varchar(30),
@name1 varchar (25),
@name2 varchar (25),
@degree varchar (12),
@k tinyint output,
@msg varchar(100) output
```

Параметри процедури:

@id_dep – ідентифікатор кафедри, де працює викладач;

@id_teach – ідентифікатор викладача;

@surname – прізвище викладача;

@name1 – ім’я викладача;

@name2 – по-батькові викладача;

@degree – вчена ступінь;

@k tinyint – код помилки додавання запису у таблицю;

@msg – повідомлення про помилку.

Збережена процедура AddTeacher перевіряє, чи є у таблиці Teachers запис з атрибутами: id_dep = @id_dep and surname = @surname and name1 = @name1 and name2 = @name2, тобто, чи вносили вже у БД дані цього викладача. Якщо такий запис є – повертається повідомлення про помилку.

Якщо такого запису немає – додається запис з вказаними атрибутами у таблицю Teachers.

Збережена процедура EditTeacher додає запис у таблицю Teachers:

```
CREATE PROC EditTeacher
@id_dep smallint,
@id_teach smallint,
@surname varchar(30),
@name1 varchar (25),
@name2 varchar (25),
@degree varchar (12),
@k tinyint output,
@msg varchar(100) output
```

Параметри процедури EditTeacher ті самі, що у процедури AddTeacher.

Збережена процедура EditTeacher перевіряє, чи є у таблиці Teachers запис з атрибутами: id_dep = @id_dep and surname = @surname and name1 = @name1 and name2 = @name2, та з умовою, що id_teach != @id_teach, тобто, чи є вже у таблиці Teachers запис про іншого викладача з такими самими даними. Якщо такий запис є – повертається повідомлення про помилку.

Якщо такого запису немає – для викладача з ідентифікатором id_teach = @id_teach змінюються інші атрибути запису на вказані параметри збереженої процедури.

Збережена процедура DelTeacher видаляє запис з таблиці Teachers:

```
CREATE PROC DelTeacher
@id_teach smallint output,
@m  tinyint,
@k tinyint output,
@msg varchar(100) output
```

Процедура DelTeacher при значенні параметру @m = 1 одразу видаляє запис з ідентифікаторм id_teach = @id_teach з таблиці Teachers, та всі записи, що з ним зв’язані у підлеглих таблицях.

При параметрі @m = 0 процедура DelTeacher розраховує кількість записів у підлеглих таблицях, що зв’язані з записом з ідентифікаторм id_teach = @id_teach у таблиці Teachers. Якщо таких записів немає, то вказаний запис в таблиці Teachers видаляється.

Якщо такі записи у підлеглих таблицях є, запис не видаляється, а генерується код помилки та повідомлення про помилку.

Збережена процедура EditTeacherDegree оновлює запис у таблиці Teachers:

```
CREATE PROC EditTeacherDegree
@id_teach smallint,
@degree varchar (12),
@date_degree smalldatetime,
@k tinyint output,
@msg varchar(100) output
```

Параметр процедури @date_degree – дата захисту кандидатської або докторської дисертації (останньої за часом). Інші параметри – як в процедурі AddTeacher.

Збережена процедура EditTeacherDegree для запису таблиці Teachers з ідентифікаторм id_teach = @id_teach змінює значення атрибутів date_degree та degree на @date_degree та @degree відповідно.

Збережена процедура EditTeacherCheck оновлює запис у таблиці Teachers:

```
CREATE PROC EditTeacherCheck
@id_teach smallint,
@check_degree bit,
@k tinyint output,
@msg varchar(100) output
```

Параметр процедури @check_degree дорівнює 1, якщо дані викладача про захист дисертації підтверджені відповідним документом, та дорівнює 0, якщо відповідний документ не представлений у відділ кадрів.

Збережена процедура EditTeacherDegree для запису таблиці Teachers з ідентифікаторм id_teach = @id_teach змінює значення атрибуту check_degree на @check_degree.

Збережена процедура MakeTeacherLogin оновлює логін та пароль викладача у базі даних:

```
CREATE PROC  MakeTeacherLogin
@id_teach smallint,
@login varchar(50),
@parol varchar(30),
@k tinyint output,
@msg varchar(100) output
```

Параметри процедури MakeTeacherLogin:

@login – логін користувача-викладача;

@parol – пароль користувача.

Процедура MakeTeacherLogin перевіряє, чи є у БД інший викладач з таким самим логіком. Якщо є – генерується повідомлення про помилку.

Якщо такого користувача немає – для запису з ідентифікатором id_teach = @id_teach змінюються логін та пароль відповідно на @login та @parol.

Збережена процедура CheckDegreeDate встановлює відмітку перевірки працівником відділу кадрів даних викладача щодо захисту ним протягом звітного року кандидатської або докторської дисертації:

```
CREATE PROC CheckDegreeDate
@id_teach smallint,
@degree varchar(12),
@date_degree smalldatetime,
@check bit
```

Параметри процедури:

@id_teach – ідентифікатор викладача;

@degree – назва наукового ступеню, для отримання якого він захистив дисертацію;

@date_degree – дата захисту дисертації;

@check – чи є у відділі кадрів документи, що підтверджують захист дисертації.

Для запису таблиці Teacher з ідентифікатором id_teach = @id_teach присвоюються значення атрибутів degree = @degree, check_degree = @check, date_degree = @date_degree.

Збережена процедура CheckTeachWork встановлює відмітку перевірки працівником відділу кадрів даних викладача щодо його практичної роботи за фахом на протязі всього трудового періоду викладача:

```
create proc CheckTeachWork
@id_teach smallint,
@WorkPlace varchar (100),
@WorkBegin smalldatetime,
@WorkEnd smalldatetime,
@is_spec bit,
@checked bit
```

Параметри процедури:

@id_teach – ідентифікатор викладача;

@WorkPlace – місце роботи;

@WorkBegin – дата прийому на роботц;

@WorkEnd – дата звільнення;

@is_spec – чи є робота за спеціальністю;

@checked – чи підтверджена робота документами.

У таблиці TeacherWork оновлюється запис зі значенням складовового унікального ключа (@id_teach, @WorkPlace, @WorkBegin). Для цього запису присвоюються значення атрибутів is_spec = @is_spec, checked = @checked.

6 ОПИС ЗАСТОСУВАННЯ ПРАЦІВНИКА КАФЕДРИ

6.1 Загальні відомості

Застосування DepAdminApp розроблене у середовище швидкої розробки програм Visual Studio 2019. Програма написана мовою C# у формі додатку WindowFormApplication.

Застосування може працювати на комп’ютері, що знаходиться у локальній мережі, з’єднаний з сервером БД, на якому розташована база даних ІС «Особистий кабінет викладача».

Програма може працювати з операційною системою Windows версії 7 та вище.

6.2 Функціональне призначення

Застосування DepAdminApp використовується для управління даними інформаційної системи «Особистий кабінет викладача».

Застосування DepAdminApp дозволяє користувачеві, що є авторизованим як відповідальний користувач кафедри, переглядати та оновлювати у базі даних список викладачів кафедри.

.За допомогою застосування DepAdminApp з дозволу відповідального користувача кафедри викладачі кафедри авторизуються у системі, тобто вводять свій логін та пароль.

6.3 Керівництво програміста

Програма DepAdminApp.exe написана на об'єктно-орієнтованої мові C#.

В програмі крім класів автоматично згенерованого середовищем Visual Studio коду, є один клас Form1, у якому код створює програміст.

Клас Form1 є класом опису головної форми. Конструктор класу не має параметрів:

public Form1()

Методи класу Form1 є обробниками подій взаємодії користувача з інтерфейсом програми. Також є методи, призначені для ініціалізації збережених процедур бази даних.

Метод init_comm_DelTeachCom ініціалізує команду DelTeachCom, яка відповідає виклику збереженої процедури DelTeacher:

private void init_comm_DellTeachCom()

Метод init_comm_AddTeachCom ініціалізує команду AddTeachCom, яка відповідає виклику збереженої процедури AddTeacher:

private void init_comm_AddTeachCom()

Метод init_comm_EditTeachProc ініціалізує команду EditTeachProc, яка відповідає виклику збереженої процедури EditTeacher:

private void init_comm_EditTeachCom()

Метод init_comm_parTeachCom ініціалізує команду parTeachCom, яка відповідає виклику збереженої процедури MakeTeacherLogin:

private void init_comm_parTeachCom ()

Метод init_comm_parExecCom ініціалізує команду parExecCom, яка відповідає виклику збереженої процедури MakeExecLogin:

private void init_comm_parExecCom ()

Інші методи класу Form1 є обробниками подій.

Метод logOKbutt_Click є обробником події натискання на кнопку logOKbutt:

private void logOKbutt_Click (object sender, EventArgs e)

В методі logOKbutt_Click визначається, чи є в базі даних користувач, якому відповідають введені користувачем логін та пароль. Якщо такий користувач є, і він зареєстрований як відповідальний користувач кафедри, на формі Form1 становляться видимими елементи, що дозволяють користувачеві працювати з додатком.

Метод delTeachButton_Click є обробником події натискання на кнопку delTeachButton:

private void delTeachButton_Click(object sender, EventArgs e)

В методі delTeachButton_Click з поточного рядка табличного елемента форми DataGridView зчитуються значення дані про запис, котрий користувач бажає видалити.

6.3 Посібник користувача

При запуску застосування на екрані з’являється головна форма програми, на якій авторизований користувач повинен вибрати кафедру, на якій він працює (рис. 2).

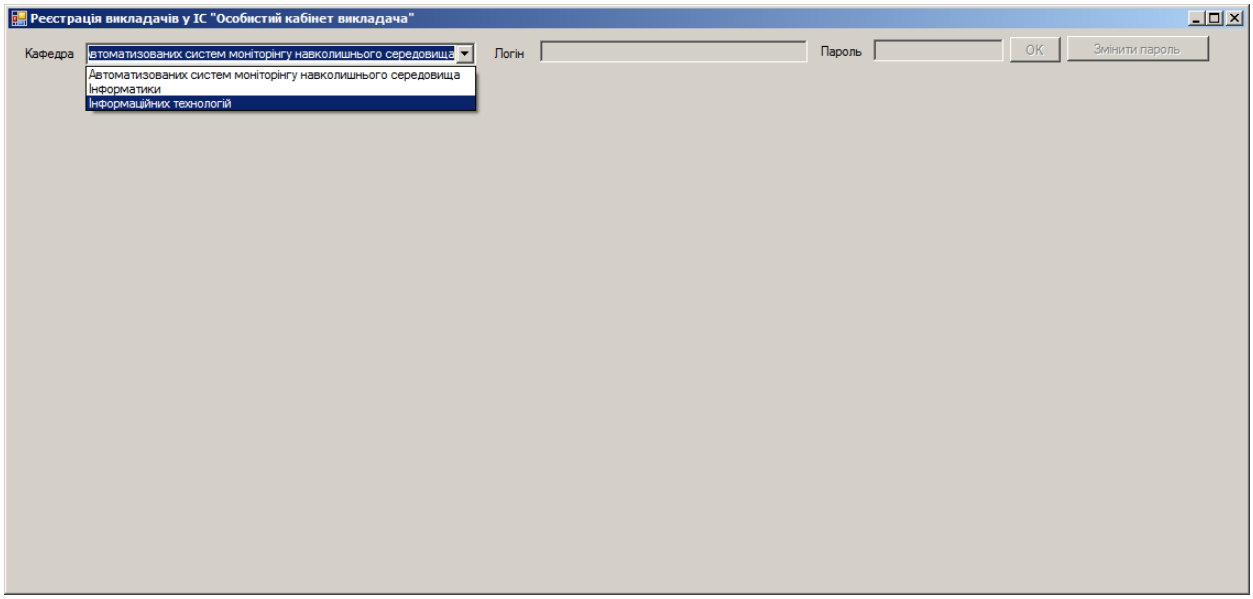

Рисунок 2 – Початок роботи з програмою: вибір кафедри користувача

Після вибору кафедри стають доступні елементи для введення логіну та паролю користувача (рис. 3).

Якщо користувач є авторизованим відповідальним користувачем від вибраної кафедри даних, після натискання ним кнопки «OK» у верхньому правому куті форми елементи для авторизації користувача зникають з форми. Випадний список кафедр університету стає недоступним.

На формі з’являється табличний компонент, на який з бази даних виводиться список викладачів вибраної кафедри; у правому верхньому куті стає доступною кнопка «Змінити пароль» (рис. 4).

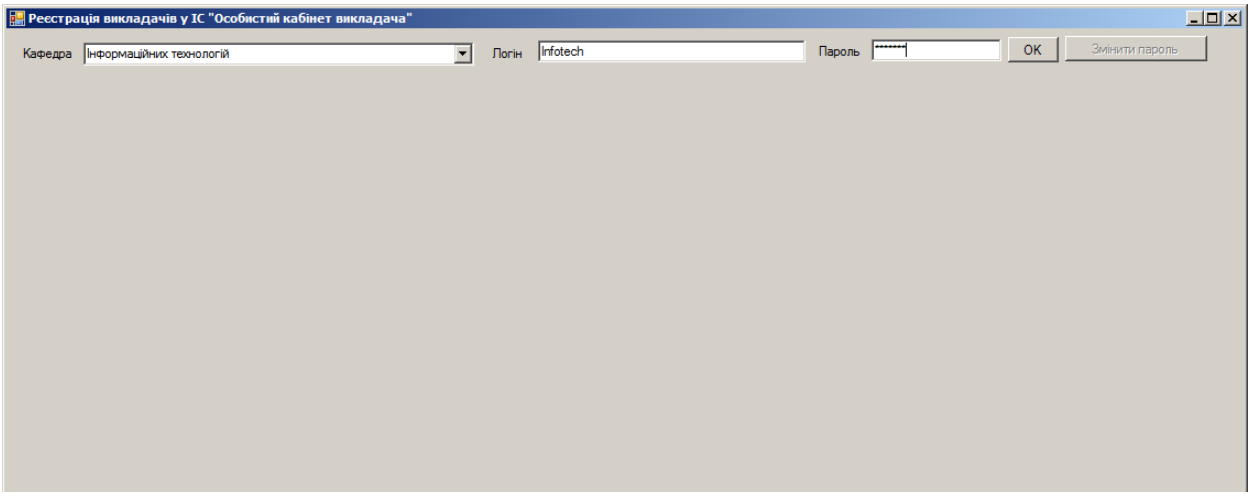

Рисунок 3 – Авторизація користувача

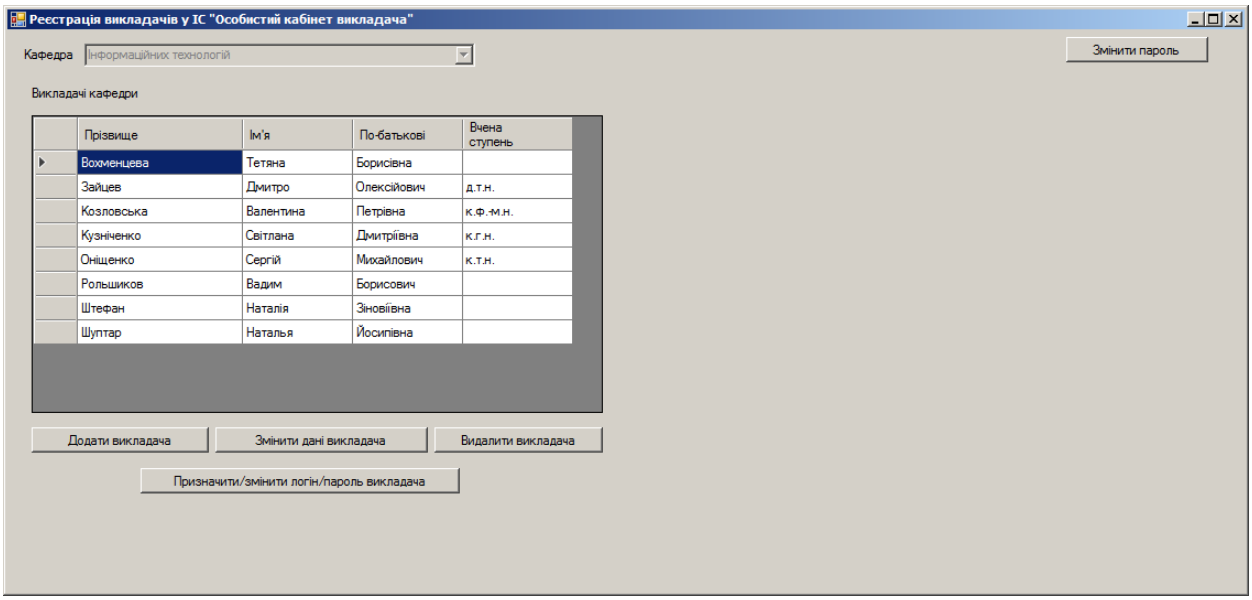

Рисунок 4 – Представлення на формі програми списку викладачів кафедри

Якщо користувач натискає кнопку «Змінити пароль», під кнопкою на формі з’являються елементи для зміни логіну та/або паролю (рис. 5). Користувачеві потрібно ввести логін та два рази ввести однаковий пароль.

Якщо логін є унікальним для БД, введені дані закріплюються за користувачем.

Під табличним компонентом розташовані кнопки маніпулювання даними у таблиці Teacher. При натисканні кнопки «Додати викладача» з’являються елементи для введення даних нового викладача кафедри (рис. 6).

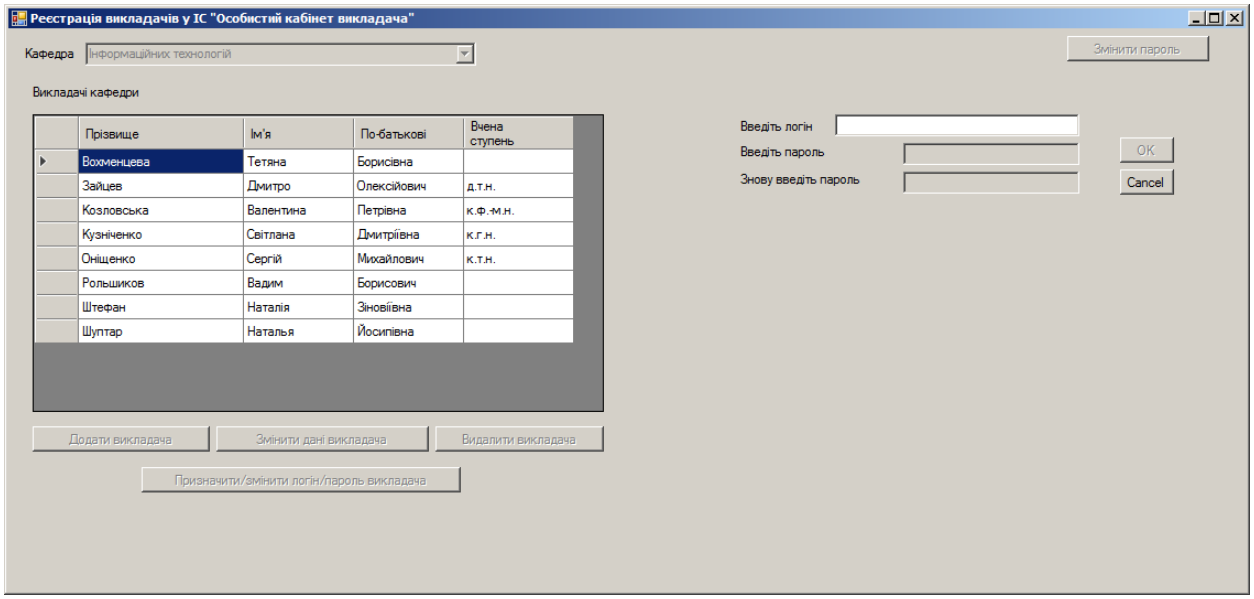

Рисунок 5 – Елементи форми для зміни користувачем своїх логіну та паролю

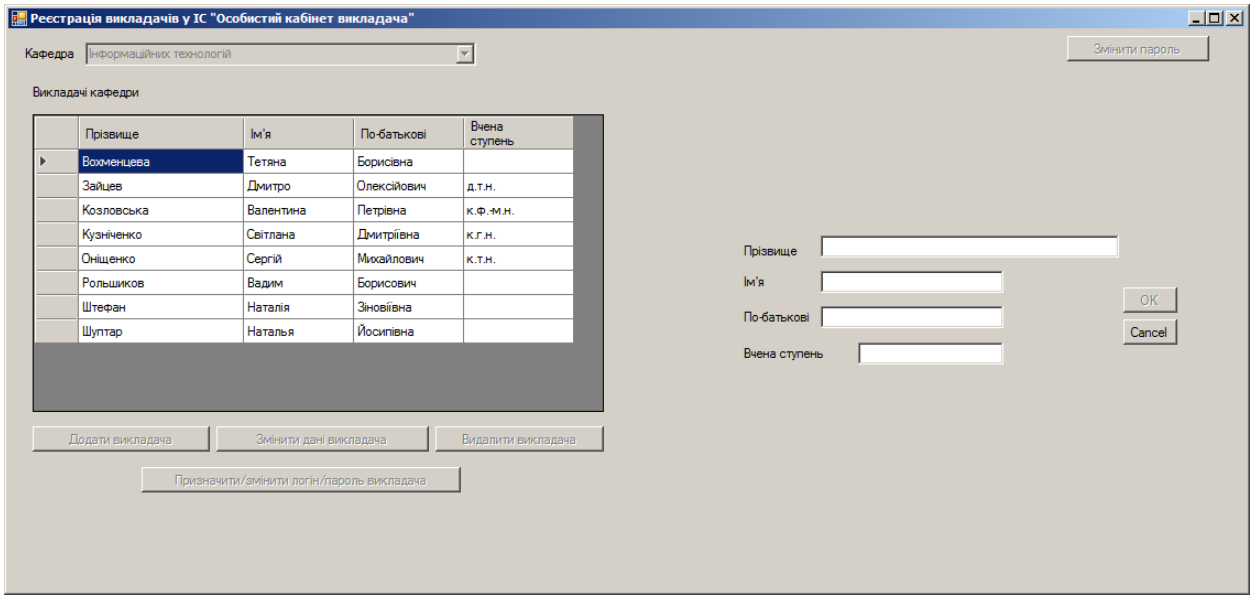

Рисунок 6 – Елементи форми для додавання нового викладача кафедри

При натисканні кнопки «Змінити дані викладача» з’являються ті самі елементи форми, що для введення даних нового викладача, але з виведеними на них даними поточного у таблиці викладача (рис. 7).

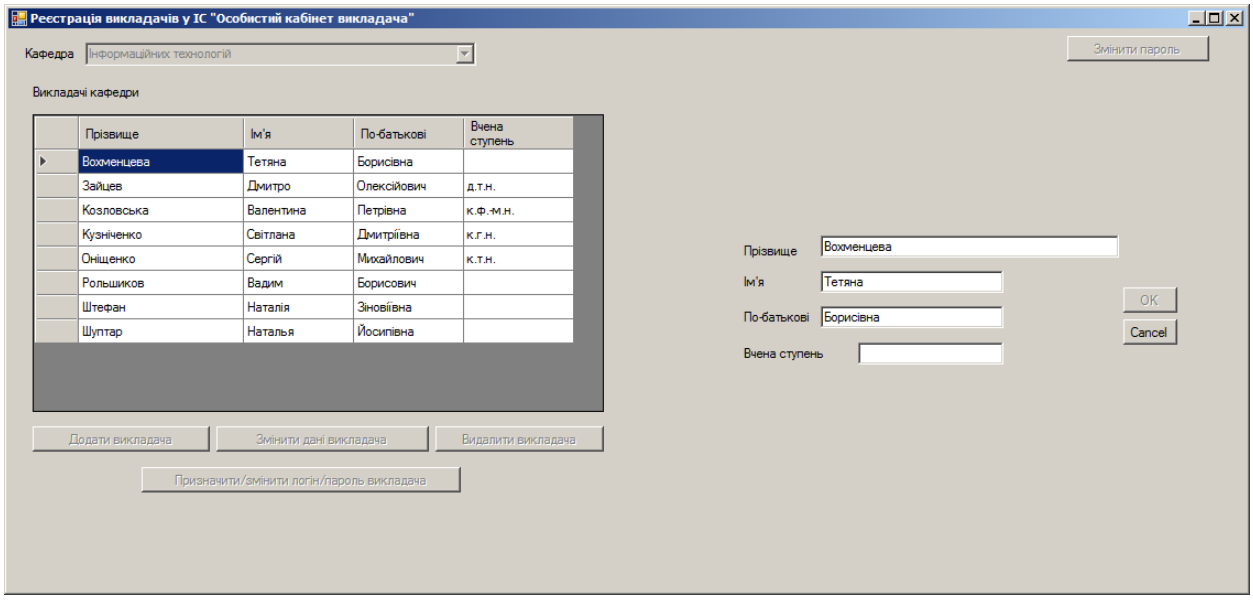

Рисунок 7 – Можливість зміни даних викладача

При натисканні кнопки «Видалити викладача» з’являється діалогове вікно з запитанням, чи дійсно потрібно видалити поточного викладача (рис. 8).

Якщо користувач натискає «ОК», запис про поточного викладача видаляється з таблиці БД і всіх підлеглих таблиць.

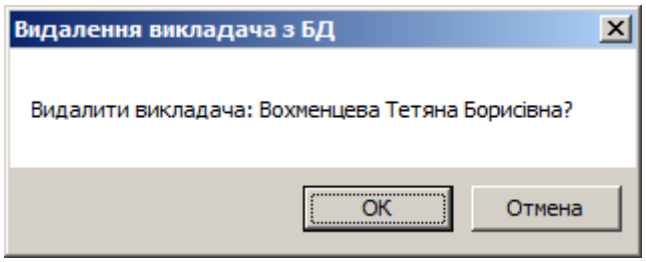

Рисунок 8 – Виведення запитання про необхідність видалення запису

Також викладачі можуть змінювати свої логін та пароль за допомогою даного застосування.

Користувач має натиснути кнопку «Призначити/змінити логін/пароль викладача» і викладач вводить свої нові логін та/або пароль (рис. 9).

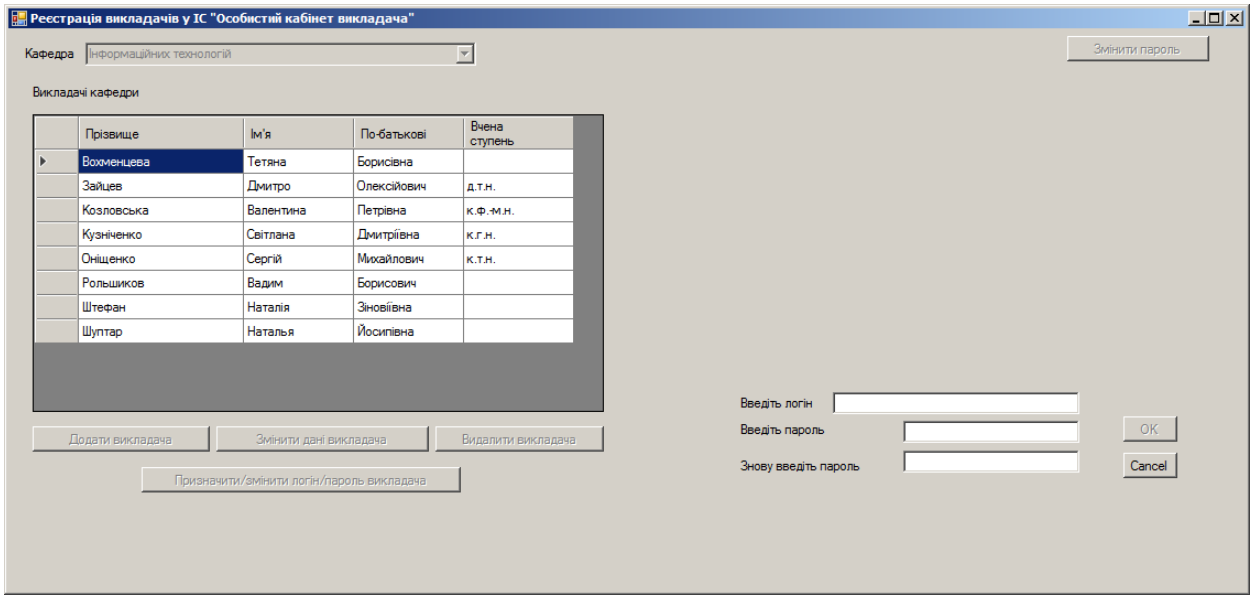

Рисунок 9 – Елементи форми для зміни логіну та паролю викладачем

Логін викладача також потрібен бути унікальним для інформаційної системи «Особистий кабінет викладача».

7 ОПИС ЗАСТОСУВАННЯ ПРАЦІВНИКА ВІДДІЛУ КАДРІВ

7.1 Загальні відомості

Застосування DepHumanApp розроблене у середовище швидкої розробки програм Visual Studio 2019. Програма написана мовою C# у формі додатку WindowFormApplication.

Застосування може працювати на комп’ютері, що знаходиться у локальній мережі, з’єднаний з сервером БД, на якому розташована база даних ІС «Особистий кабінет викладача».

Програма може працювати з операційною системою Windows версії 7 та вище.

7.2 Функціональне призначення

Застосування DepHumanApp використовується для управління даними інформаційної системи «Особистий кабінет викладача».

Застосування DepHumanApp дозволяє користувачеві, що є авторизованим як відповідальний користувач відділу кадрів, переглядати списки викладачів кафедр та оновлювати у базі даних відмітки про коректність або некоректність даних викладача, які мають бути підтверджені працівником відділу кадрів..

7.3 Керівництво програміста

Програма DepHumanApp.exe написана на об'єктно-орієнтованої мові C#.

В програмі крім класів автоматично середовищем Visual Studio згенерованого коду, є один клас Form1, у якому код створює програміст.

Клас Form1 є класом опису головної форми. Конструктор класу не має параметрів:

public Form1()

Методи класу Form1 є обробниками подій взаємодії користувача з інтерфейсом програми. Також є методи, призначені для ініціалізації збережених процедур бази даних.

Метод init_checkDegreeCom ініціалізує команду checkDegreeCom, яка відповідає виклику збереженої процедури CheckDegreeDate:

private void init_checkDegreeCom ()

Метод init_ChechWorkCom ініціалізує команду ChechWorkCom, яка відповідає виклику збереженої процедури CheckTeachWork:

private void init_ChechWorkCom ()

Інші методи класу Form1 є обробниками подій.

Метод EnterButton_Click є обробником події натискання на кнопку EnterButton:

private void EnterButton_Click(object sender, EventArgs e)

В методі EnterButton_Click визначається, чи є в базі даних користувач, якому відповідають введені користувачем логін та пароль. Якщо такий викладач є, і він зареєстрований як адміністратор даних, на формі Form1 становляться видимими елементи, що дозволяють користувачеві працювати з додатком.

Метод delDepButton_Click є обробником події натискання на кнопку delDepButton:

private void delDepButton_Click(object sender, EventArgs e)

В методі delDepButton_Click з поточного рядка табличного елемента форми DataGridView зчитуються значення дані про запис, котрий користувач бажає видалити. Текстові атрибути запису включаються у запитання до користувача, чи дійсно він хоче видалити запис з вказаними атрибутами.

Якщо користувач підтверджує бажання видалити поточний запис з таблиці, викликається відповідна збережена процедура, яка отримує ідентифікатор запису у якості параметру процедури. Процедура видаляє вказаний запис, та зв’язані з ним записи у підлеглих таблицях БД, і оновлюються дані у табличному елементі DataGridView.

7.3 Посібник користувача

При запуску застосування працівника відділу кадрів на екрані з’являється форма програми, на якій розташований компонент з двома вкладками (рис. 10).

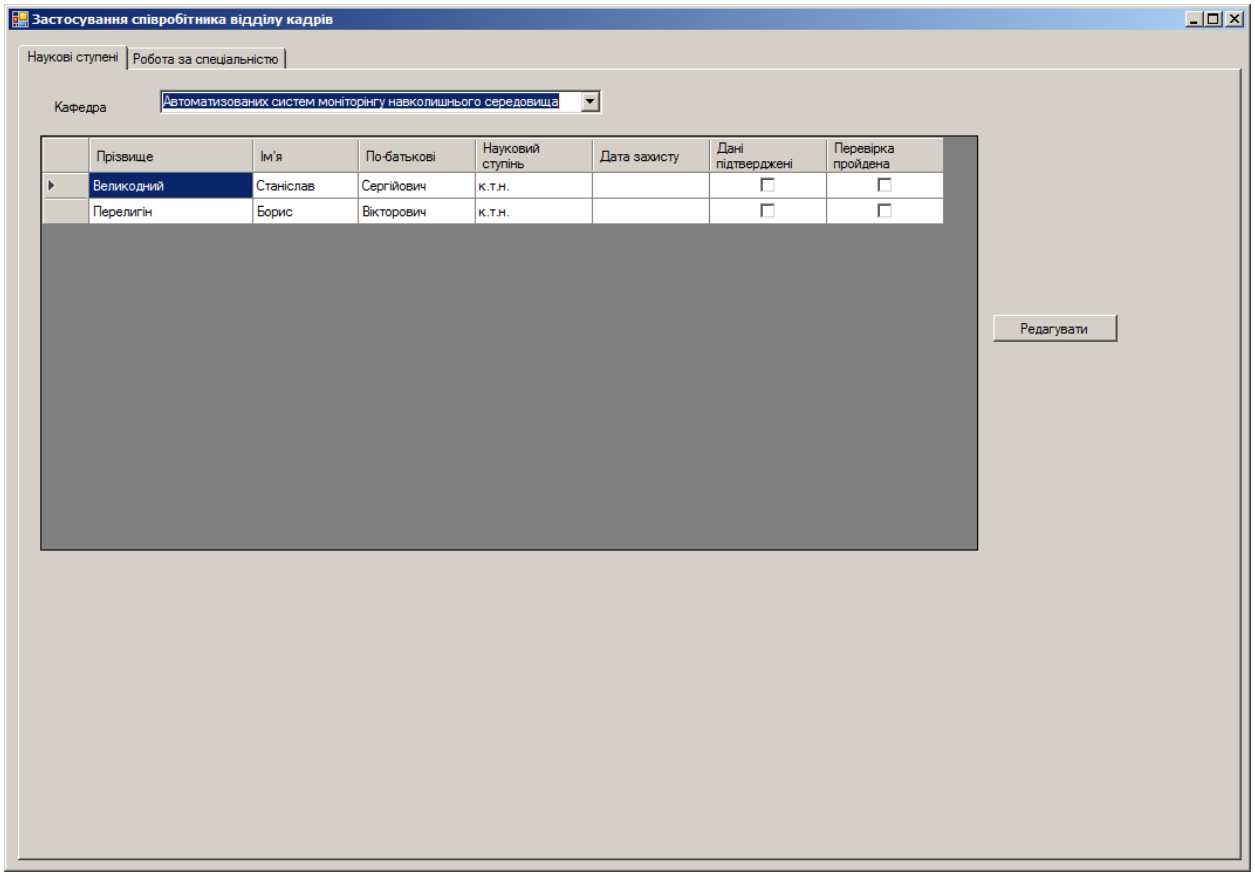

Рисунок 10 – Форма програми з двома вкладками

Перша вкладка має назву «Наукові ступені». Вона має випадний список, табличний компонент та кнопку. У табличний компонент виводиться список викладачів кафедри з зазначенням наукового ступеню, у кого він є. Дані виводяться для викладачів тієї кафедри, що вибрана у випадному списку наверху вкладки. Користувач може змінити кафедру у списку (рис. 11), і тоді на формі з’явиться список викладачів нової вибраної кафедри (рис. 12).

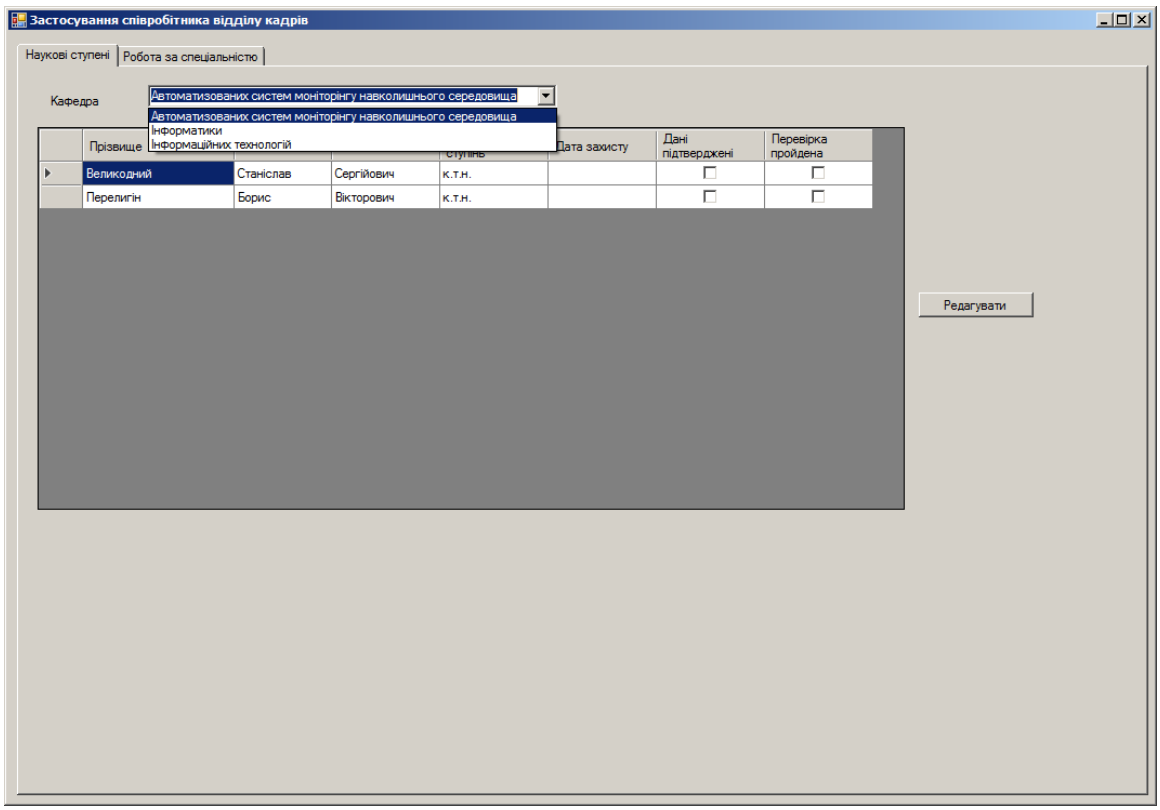

Рисунок 11 – Зміна кафедри

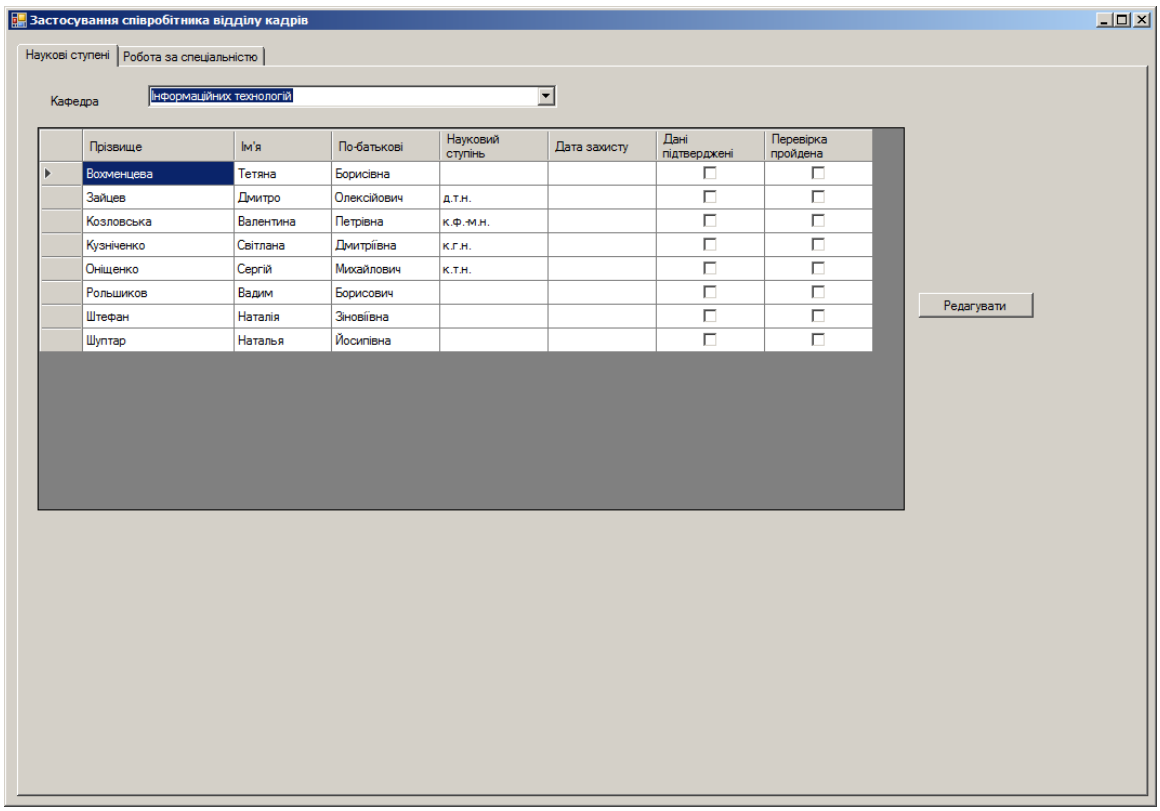

Рисунок 12 – Список викладачів вибраної кафедри

На вкладці є тільки одна кнопка для редагування даних у таблиці Teacher бази даних. Самі дані про викладачів користувач з відділу кадрів не вводить та не редагує. Ці дані вводять відповідальний користувач кафедри та сам викладач. Працівник відділу кадрів тільки перевіряє ті дані, що відносяться до його компетенції, та ставить відмітку, чи вони правильні. На першій вкладці потрібно перевіряти, чи захистив викладач у звітному році кандидатську або докторську дисертацію – ці дані потрібні для розрахунку рейтингу викладача.

Викладач повинен сам вносити у базу даних іС «Особистий кабінет викладача» дані прозахист дисертації та отримання наукового ступеня. Якщо цих даних для поточного викладача у списку немає, при натисканні кнопки редагувати програма видає повідомлення, що немає даних, що перевіряються (рис. 13).

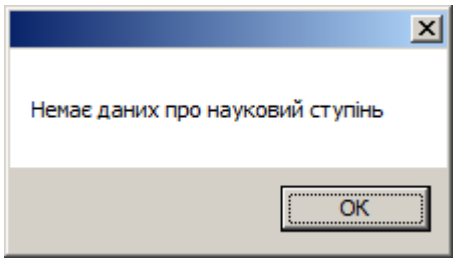

Рисунок 13 – Повідомлення про відсутність даних

Якщо у поточного викладача є науковий ступінь, на елементи понизу табличного компонента виводиться дата захисту дисертації, введена у БД самим викладачем. Поряд знаходиться компонент, у якому працівник відділу кадрів відмічає, що він виконав перевірку даних (рис. 14).

Якщо перевірки не було у колонці «Перевірка пройдена» табличного компоненту не буде стояти відмітки. Якщо перевірка була, у цій колонці у викладача буде стояти відмітка. Якщо дані користувача коректні, також з’явиться відмітка у колонці «Дані підтверджені» (рис. 15).

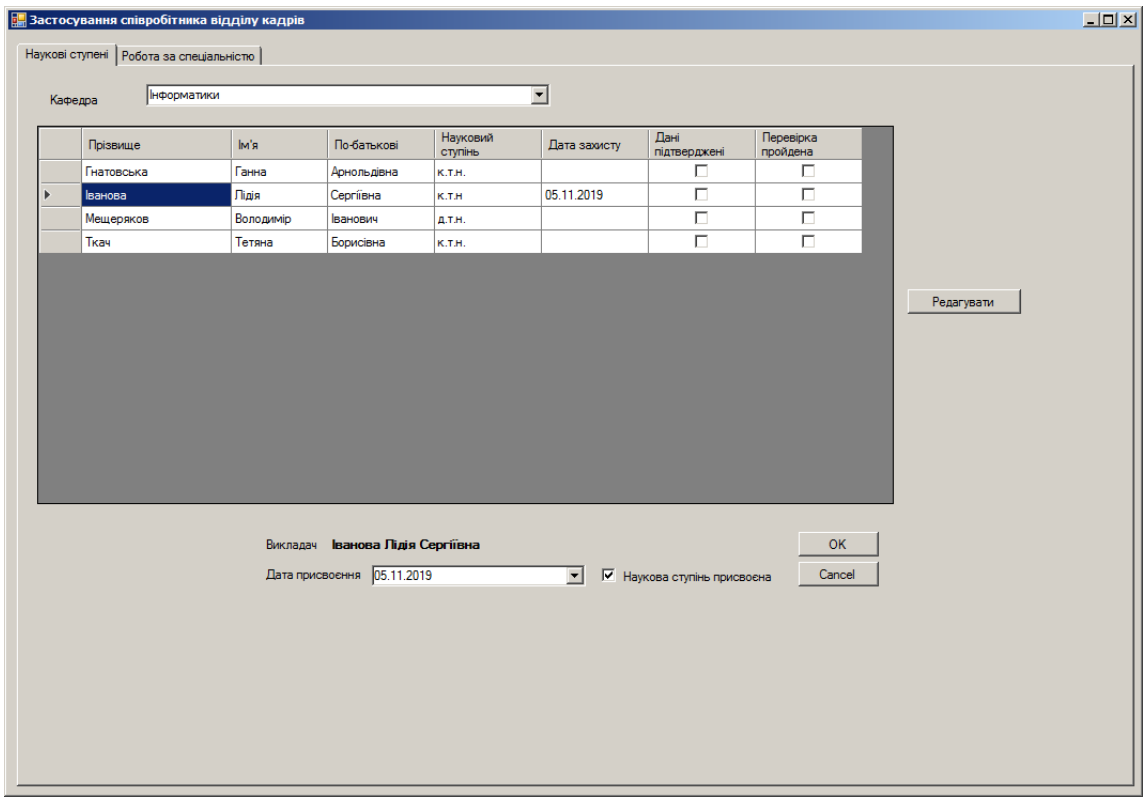

Рисунок 14 – Внесення користувачем результату перевірки даних

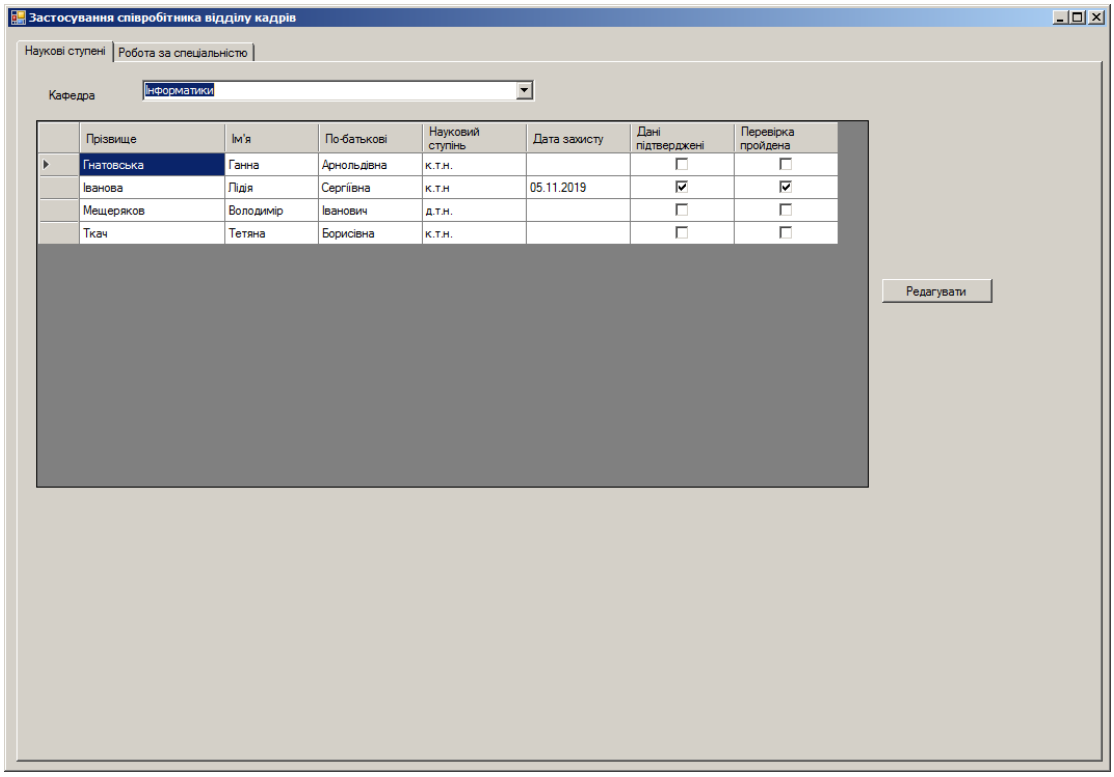

Рисунок 15 – Відмітка, що перевірка підтвердила дані користувача

Якщо дані некоректні, цієї відмітки не буде (рис. 16).

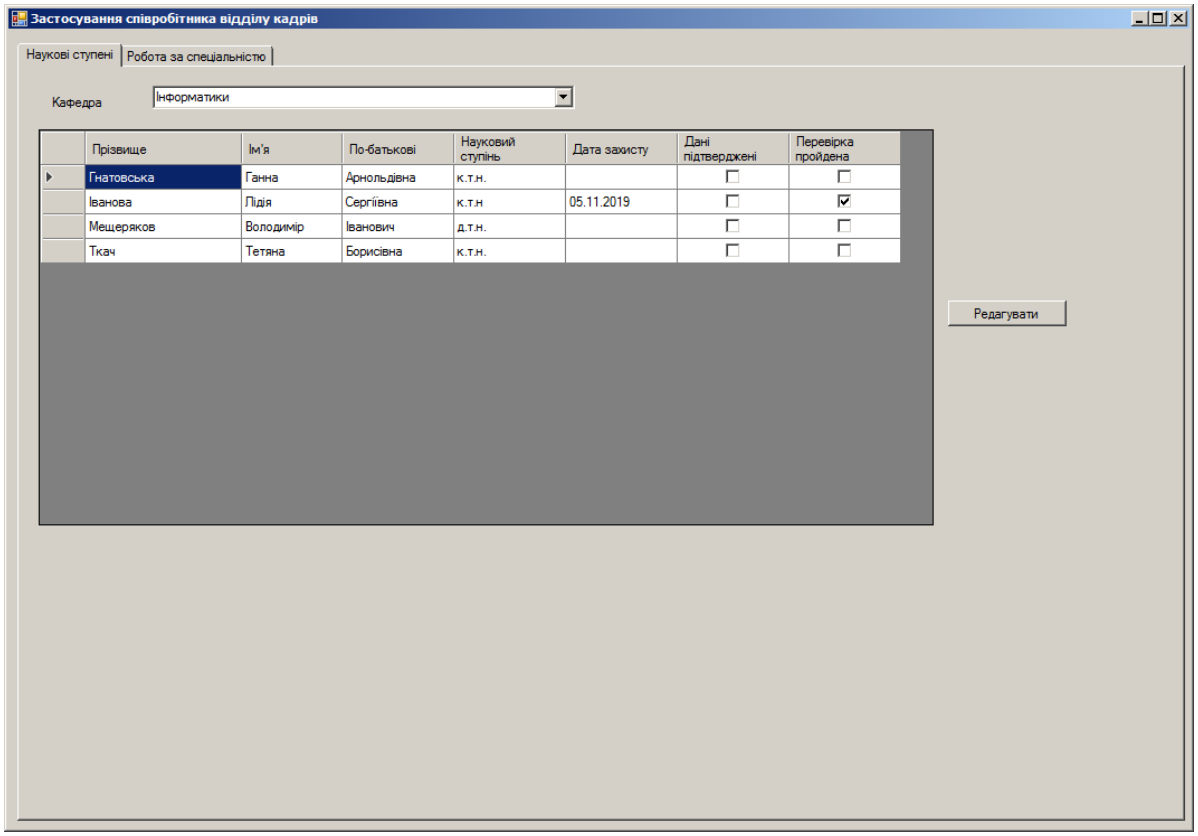

Рисунок 16 – Відмітка, що перевірка не підтвердила дані користувача

Друга вкладка «Робота за спеціальністю» призначена для перевірки даних викладачів щодо їх попередньої практичної роботи за фахом (рис. 17).

На вкладці розташовані два табличні компоненти. У верхньому компоненті виводиться список викладачів кафедри, що вибрана у випадному списку. У нижньому компоненті для поточного викладача виводяться дані щодо його попередніх практичних робіт за фахом; ці дані викладача повинен сам внести в БД.

Якщо поточний користувач не вносив ці дані, нижній табличний компонент буде порожнім (рис. 18). У цьому випадку кнопка «Редагувати» поруч з нижнім табличним компонентом буде недоступною.

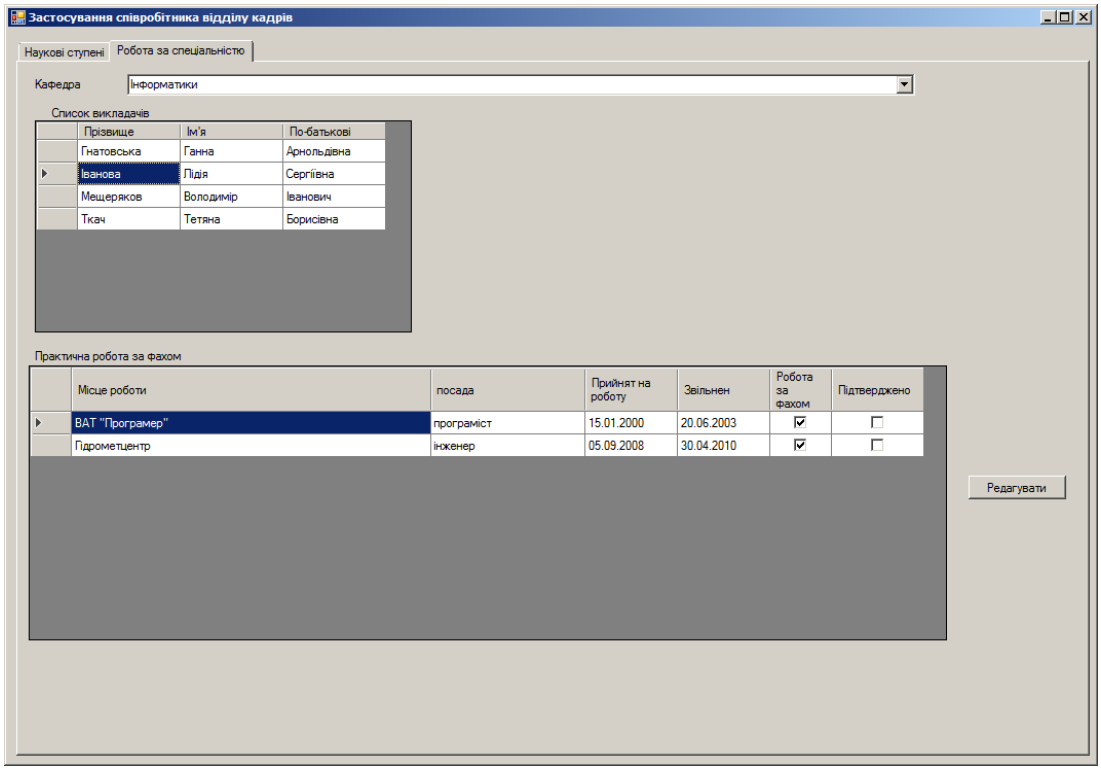

Рисунок 17 – Наявність даних про практичну роботу за фахом

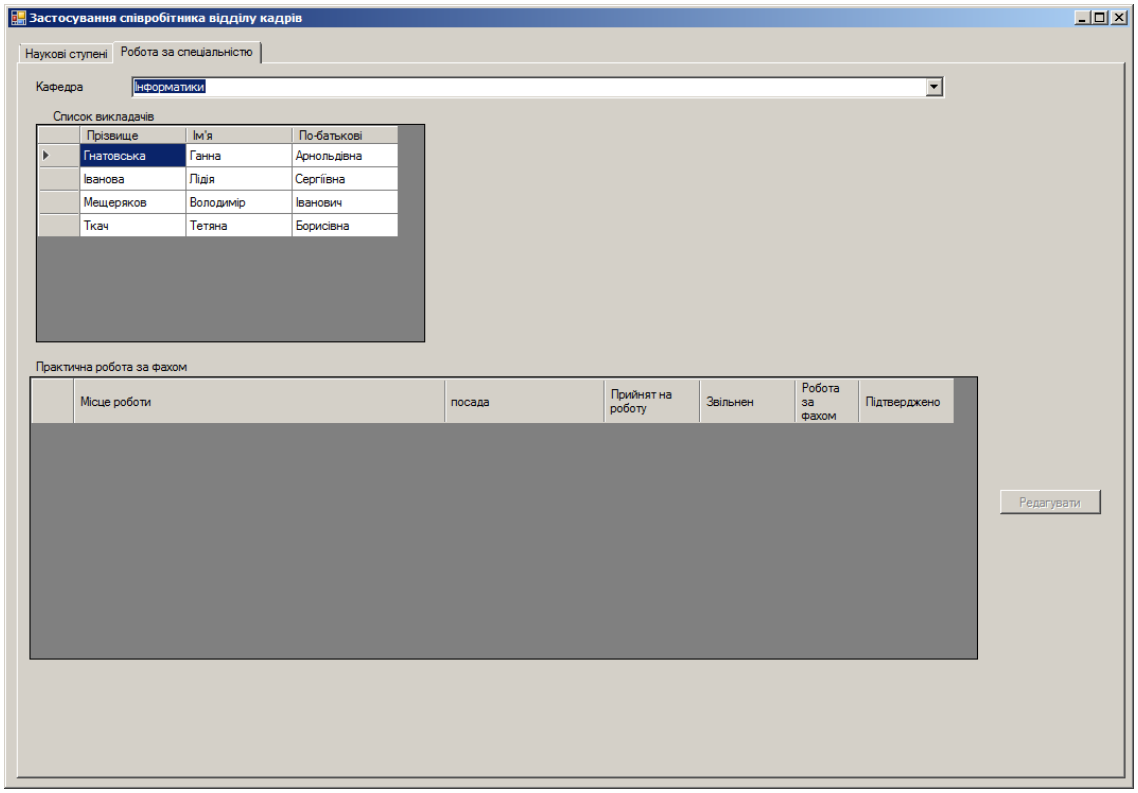

Рисунок 18 – Відсутність даних про практичну роботу за фахом

Якщо дані є, працівник відділу кадрів повинен їх перевірити, а потім поставити у БД відмітку, коректні ці дані, чи ні.

Для постановки відмітки потрібно натиснути кнопку «Редагувати» біля нижнього табличного компонента (рис. 19).

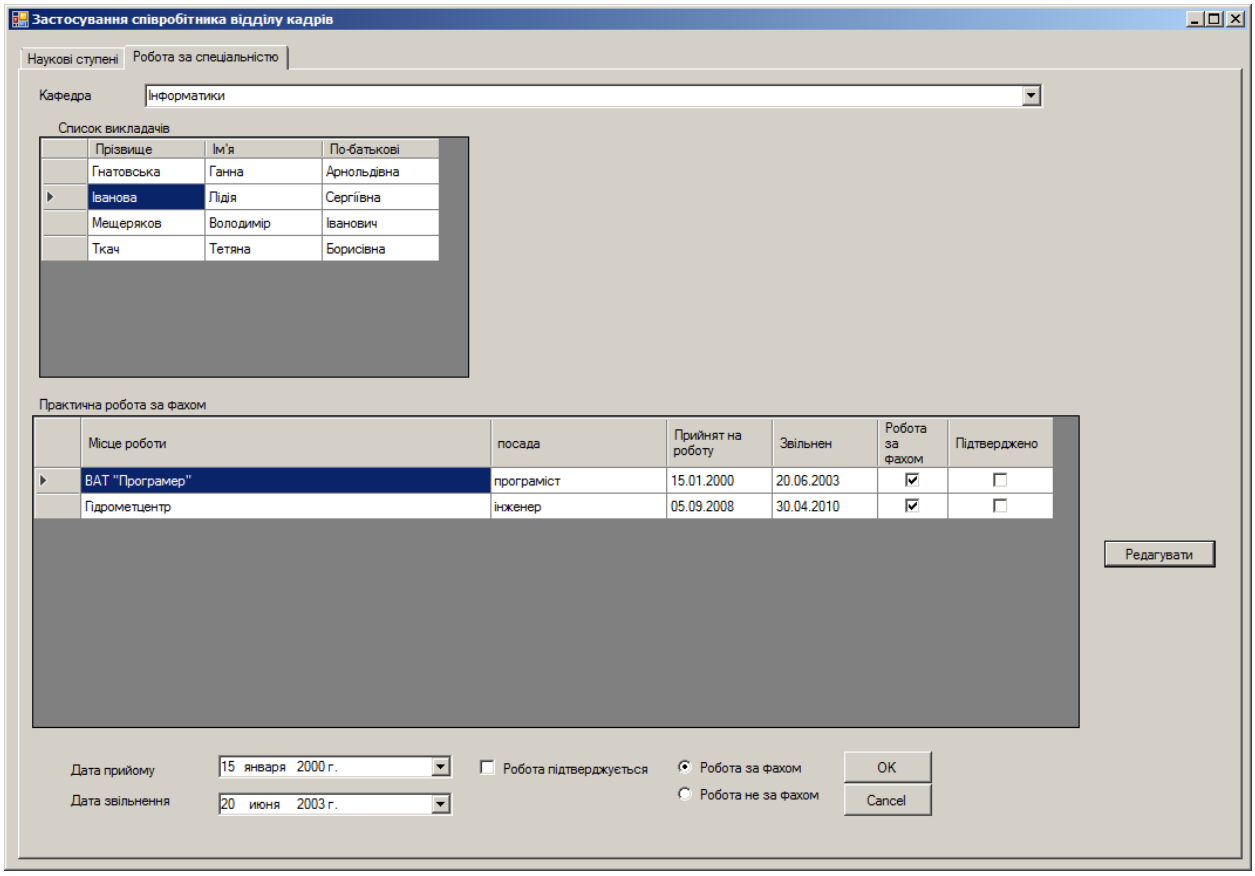

Рисунок 19 – Елементи для внесення відмітки про коректні дані щодо практичної роботи за фахом

При підтверджені даних користувач може змінити відмітку, що це робота за фахом. Може також змінити дати прийому та звільнення, але не може змінити місце роботи та посаду. Якщо дані про роботу коректні (можливо, після деякого корегування), користувач ставить відмітку «Робота підтверджується» та натискає кнопку «OK» (рис. 20).

Підтвердження даних відображається у нижньому табличному компоненті.

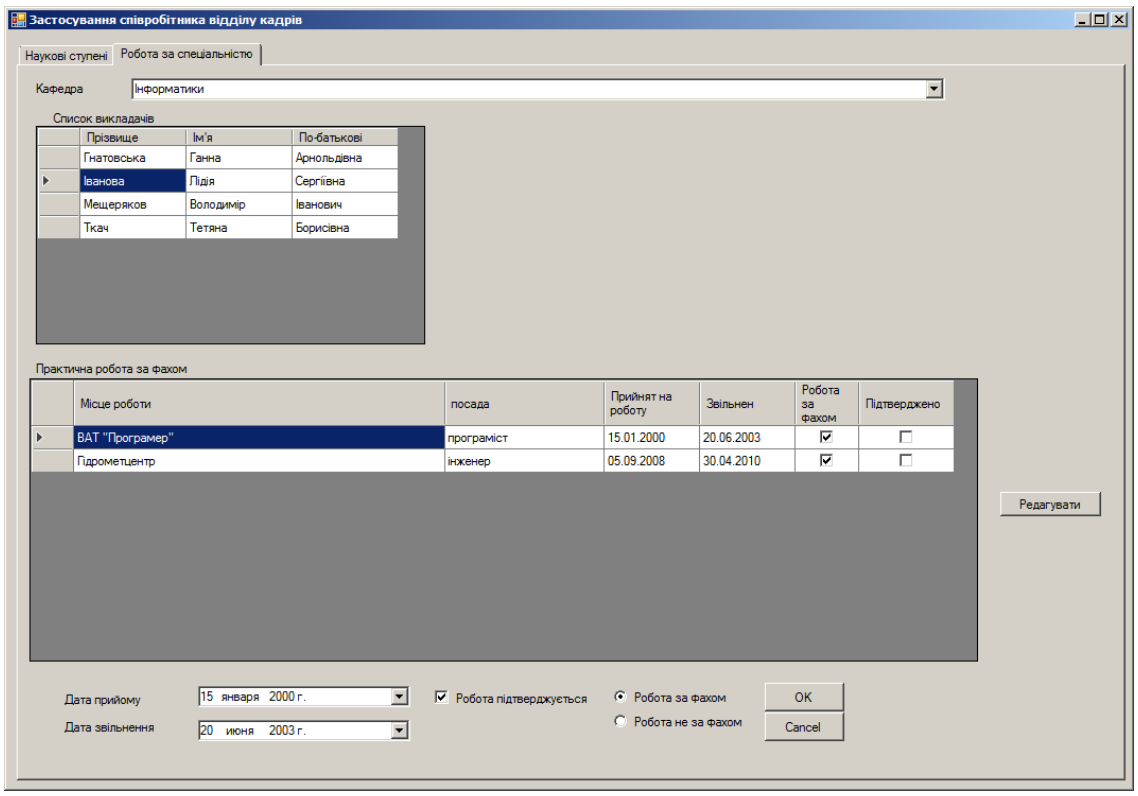

Рисунок 20 – Підтвердження даних викладача

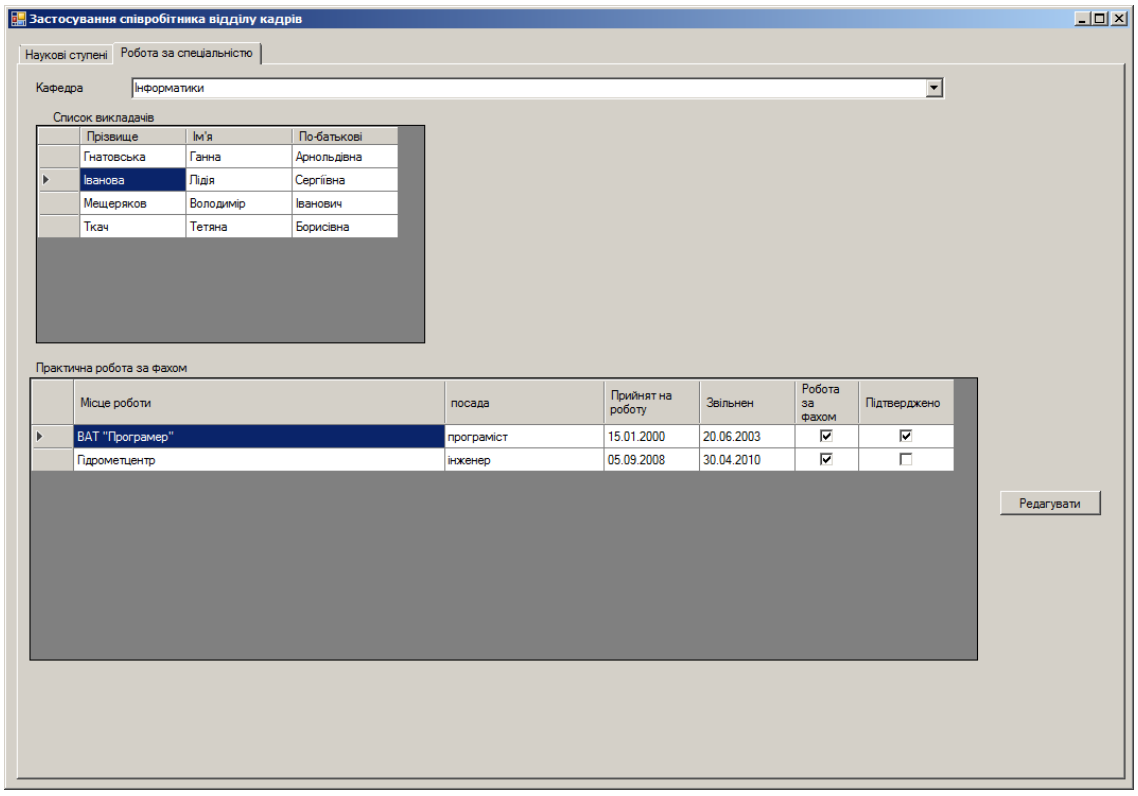

Рисунок 21 – Відображення підтверджених даних

ВИСНОВКИ

Оцінка професійного рівня викладача потребує часу та зусиль співробітників різних підрозділів університету, і ця оцінка необхідна, оскільки вона входить складовою до ліцензійних вимог при акредитації закладу вищої освіти.

Викладачеві потрібен час та увага для зведення його особистих даних до потрібного звітного виду. Тому бажано автоматизувати процес звітності викладача, перевірки вмісту його звіту та отримання результатів у вигляді таблиці рейтингу викладачів університету та таблиці відповідності викладачів кадровим вимогам (ліцензійним умовам).

Деякі дані, за якими викладач може отримати виконання пункту ліцензійних вимог або отримати додаткові бали при розрахунку його рейтингу, повинні перевірятись працівниками відділу кадрів.

Для можливості виконання цих дій було створене застосування працівника відділу кадрів.

Також було розроблене застосування відповідального користувача кафедри для внесення у базу даних списків викладачів кафедр. Це застосування дозволяє додавати, видаляти викладачів з кафедральних списків, а також змінювати деякі їх атрибути, які задає сам працівник кафедри. Застосування працівника кафедри також дозволяє виконувати авторизацію користувачів системи, які є викладачами кафедри.

ПЕРЕЛІК ПОСИЛАНЬ

1. Про затвердження Ліцензійних умов провадження освітньої діяльності закладів освіти. Постанова КМУ № 1187 від 30.12.15 року. URL: http://mdu.edu.ua/wp-content/uploads/postanova%E2%84%961187.pdf
2. Про затвердження Ліцензійних умов провадження освітньої діяльності. Документ 1187-2015-п, чинний, поточна редакція – Редакція від 23.05.2018, підстава – 347-2018-п. URL: https://zakon.rada.gov.ua/ laws/show/ 1187-2015-%D0%BF
3. Вендров А. М. CASE-технологии. Современные методы и средства проектирования информационных систем. М: «Финансы и статистика», 1998. – 98 с.
4. Куликов Г. Г., Никулина Н. О., Речкалов А. В. Управление проектами на основе системного моделирования: Учебное пособие. Уфа: УГАТУ, 2009. – 171 с.
5. Реляционные БД vs Объектно-ориентированные БД. URL: https:// habr.com/ru/post/93356 (Дата звернення 08.12.2019)
6. Объектно-ориентированные базы данных – основные концепции, организация и управление: краткий обзор URL: http://www.cyberguru.ru/ database/database-theory/object-oriented-db-overview.html
7. Хранимые процедуры. URL: https://metanit.com/sql/sqlserver/11.1.php (Дата звернення 10.05.2020)
8. CREATE PROCEDURE (Transact-SQL). Microsoft. Документация по SQL. URL: https://docs.microsoft.com/ru-ru/sql/t-sql/statements/create-procedure- transact-sql?view=sql-server-ver15 (Дата звернення 10.05.2020)
9. Полное руководство по языку программирования C# 8.0 и платформе .NET Core 3. URL: https://metanit.com/sharp/tutorial/(Дата звернення 08.12.2019)

10. Руководство по программированию на C#. URL: https://docs.microsoft.com/ ru-ru/dotnet/csharp/programming-guide/ (Дата звернення 10.05.2020)
11. Томас Конноли, Каролин Бегг. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 1440 с.
12. Дейт К., Дж. Введение в системы баз данных. 6-е издание: Пер. с англ. – К.; М.; СПб.: Издательский дом «Вильямс». 2000. – 848 с.

ДОДАТОК А

Кадрові вимоги

Кадрові вимоги щодо забезпечення провадження освітньої діяльності у сфері вищої та післядипломної освіти для осіб з вищою освітою

(Витяг з
Ліцензійних умов провадження освітньої діяльності закладів освіти,

затверджених постановою КМУ від 10.05.2018р. № 347)

28. Науково-педагогічні та наукові працівники, які здійснюють освітній процес, повинні мати стаж науково-педагогічної діяльності понад два роки та рівень наукової та професійної активності, який засвідчується виконанням **не менше чотирьох видів та результатів** з перелічених у пункті 30 цих Ліцензійних умов.

Для другого (магістерського) рівня вищої освіти кадровий склад закладу освіти повинен включати з розрахунку на кожні десять здобувачів освітнього ступеня магістра одного викладача, який має кваліфікацію відповідно до спеціальності, науковий ступінь або вчене звання.

Для третього (освітньо-наукового/освітньо-творчого) рівня (доктор філософії/доктор мистецтва) кадровий склад закладу освіти повинен включати з розрахунку на кожних два здобувачі освітнього ступеня доктора філософії/ доктора мистецтва одного викладача, який має науковий ступінь та/або вчене звання, а також кваліфікацію відповідно до спеціальності, підтверджену науковою, науково-педагогічною, педагогічною чи іншою професійною діяльністю за відповідною спеціальністю **за не менш як сімома видами чи результатами**, переліченими у пункті 30 цих Ліцензійних умов.

.....

30. Види і результати професійної діяльності особи за спеціальністю, яка застосовується до визнання кваліфікації, відповідної спеціальності:

1) наявність за останні п’ять років наукових публікацій у періодичних виданнях, які включені до наукометричних баз, рекомендованих МОН, зокрема Scopus або Web of Science Core Collection;

2) наявність не менше п’яти наукових публікацій у наукових виданнях, включених до переліку наукових фахових видань України;

3) наявність виданого підручника чи навчального посібника або монографії;

4) наукове керівництво (консультування) здобувача, який одержав документ про присудження наукового ступеня;

5) участь у міжнародних наукових проектах, залучення до міжнародної експертизи, наявність звання “суддя міжнародної категорії”;

6) проведення навчальних занять із спеціальних дисциплін іноземною мовою в обсязі не менше 50 аудиторних годин на навчальний рік;

7) робота у складі експертних рад з питань проведення експертизи дисертацій МОН або галузевих експертних рад Національного агентства із забезпечення якості вищої освіти, або Акредитаційної комісії, або їх експертних рад, або міжгалузевої експертної ради з вищої освіти Акредитаційної комісії, або трьох експертних комісій МОН/зазначеного

Агентства, або Науково-методичної ради/науково-методичних комісій (підкомісій) з вищої освіти МОН;

8) виконання функцій наукового керівника або відповідального виконавця наукової теми (проекту), або головного редактора/члена редакційної колегії наукового видання, включеного до переліку наукових фахових видань України, або іноземного рецензованого наукового видання;

9) керівництво школярем, який зайняв призове місце III—IV етапу

Всеукраїнських учнівських олімпіад з базових навчальних предметів, II— III етапу Всеукраїнських конкурсів-захистів науково-дослідницьких робіт учнів — членів Національного центру "Мала академія наук України"; участь у журі олімпіад чи конкурсів "Мала академія наук України";

10) організаційна робота у закладах освіти на посадах керівника (заступника керівника) закладу освіти/інституту/факультету/відділення (наукової установи)/філії/кафедри
або іншого відповідального за підготовку здобувачів вищої освіти підрозділу/відділу (наукової установи)/навчально-методичного управління (відділу)/лабораторії/іншого навчально-наукового (інноваційного) структурного підрозділу

вченого секретаря закладу освіти (факультету, інституту)/відповідального секретаря приймальної комісії та його заступника;

11) участь в атестації наукових працівників як офіційного опонента або члена постійної спеціалізованої вченої ради (не менше трьох разових спеціалізованих вчених рад);

12) наявність не менше п'яти авторських свідоцтв та/або патентів загальною кількістю два досягнення;

13) наявність виданих навчально-методичних посібників/посібників для самостійної роботи студентів та дистанційного навчання, конспектів лекцій/практикумів/методичних вказівок/рекомендацій загальною кількістю три найменування;

14) керівництво студентом, який зайняв призове місце на I етапі Всеукраїнської студентської олімпіади (Всеукраїнського конкурсу студентських наукових робіт), або робота у складі організаційного комітету/журі Всеукраїнської студентської олімпіади (Всеукраїнського конкурсу студентських наукових робіт), або керівництво постійно діючим студентським науковим гуртком/проблемною групою; керівництво студентом, який став призером або лауреатом Міжнародних мистецьких конкурсів, фестивалів та проектів, робота у складі організаційного комітету або у складі журі міжнародних мистецьких конкурсів, інших культурно-мистецьких проектів; керівництво студентом, який брав участь в Олімпійських, Параолімпійських іграх, Всесвітній та Всеукраїнській Універсіаді, чемпіонаті світу, Європи, Європейських іграх, етапах Кубка світу та Європи, чемпіонаті України; виконання обов'язків тренера, помічника тренера національної збірної команди України з видів спорту; виконання обов'язків головного секретаря,

головного судді, судді міжнародних та всеукраїнських змагань; керівництво спортивною делегацією; робота у складі організаційного комітету, суддівського корпусу;

15) наявність науково-популярних та/або консультаційних (дорадчих) та/або дискусійних публікацій з наукової або професійної тематики загальною кількістю не менше п’яти публікацій;

16) участь у професійних об’єднаннях за спеціальністю;

17) досвід практичної роботи за спеціальністю не менше п’яти років;

18) наукове консультування установ, підприємств, організацій протягом не менше двох років.

Під час визначення рівня наукової та професійної активності науково-педагогічного (наукового) працівника можуть зараховуватися здобутки за попередніми місцями роботи, п’ятирічний строк може продовжуватися на час перерви в роботі з об’єктивних причин (соціальна відпустка, призов/мобілізація на військову службу чи військова служба за контрактом, тривала хвороба тощо).

ДОДАТОК Б

Лист самоатестації викладача

№	Назва показника	Показник	Ваговий коефіцієнт	Результат
	І.1 Педагогічна діяльність			
1.1	Кількість навчальних дисциплін, які викладались викладачем державною мовою		1	
1.2	Кількість студентів, успішно захистивших дипломні проекти (роботи) у поточному календарному році під керівництвом викладача		0,15	
1.3	Кількість навчальних дисциплін у звітному календ. році, за якими успішність навчання на кінець літньої та зимової сесій була нижче 85%		1	
	Індивідуальний показник роботи:			
	І.2 Методична робота			
1.4	Кількість методичних вказівок, підготовлених та затверджених у поточному календарному році		1	
1.5	Кількість навчальних посібників (у тому числі конспектів лекцій), підготовлених та затверджених у поточному календарному році		2	
1.6	Кількість підручників, підготовлених та затверджених у поточному календарному році		3	
1.7	Участь у роботі проектних груп по розробці освітніх програм та навчальних планів		0,5	
1.8	Кількість рецензій на рукописи підручників, навчальних посібників (окрім конспектів лекцій)		0,25	
1.9	Кількість рецензій на рукописи конспектів лекцій		0,1	
1.10	Кількість розроблених робочих програм навчальних дисциплін, які вперше вводяться у поточному році		0,25	
1.11	Кількість робочих програм навчальних дисциплін, які були перероблені		0,1	
1.12	Кількість дисциплін, по яких розроблено та затверджено тестові завдання для проведення ректорського контролю		1	
1.13	Участь у створенні навчально-методичного комплексу англійською мовою		3	
1.14	Участь у створенні електронного інтерактивного курсу дисципліни для дистанційного навчання		3	
1.15	Методичне керівництво роботою студентів з підготовки складових вмісту офіційного сайту університету		0,5	
1.16	Підвищення кваліфікації: стажування в межах України		0,5	
	Підвищення кваліфікації: навчання на курсах		1	
	Підвищення кваліфікації: навчання, стажування або		3	

	праця в країні, яка входить до ОЕСР та/або ЄС		
1.17	Керівництво стажуванням викладачів		0,1
1.18	Викладання на курсах підвищення кваліфікації		1
1.19	Кількість підготованих у складі авторських колективів запитів для отримання грантів на участь у міжнародних освітніх проектах		1
1.20	Кількість отриманих авторами запиту грантів на виконання на базі університету грантів за міжнародними освітніми програмами		3
1.21	Кількість навчальних дисциплін, за якими успішність по ректорському контролю була нижче 85%		
	Індивідуальний показник роботи:		
	II. Підготовка науково-педагогічних кадрів		
2.1	Захист у звітному календ. році докторської дисерт.		3
2.2	Захист у звітному календ. році кандидат. дисертац.		1,5
2.3	Наукове консультування успішно захищеною у звітному кален. році докторською дисератцією		1,5
2.4	Наукове керівництво успішно захищеною у звітн. календар. році кандидатської дисертації		1
2.5	Кіл-сть магістрів, захістивших магістерські роботи під керівництвом викладача у звітному календ.році		0,25
	Індивідуальний показник роботи:		
	III. Видавнича та патентна діяльність		
3.1	Кількість виданих у звітному календ.році особисто підготовлених викладачем монографій, виданих:		
	а) в Україні		3
	б) за кордоном		4
	в) у міжнародних видавництвах системи Scopus		6
3.2	Кількість підготовлених викладачем розділів в колективній науковій монографії, виданої в Україні		1
3.3	Кількість надрукованих у поточному календарному році статей (розділів) у міжнародних журналах, збірниках, колективних монографіях		
	а) що входять в систему Scopus		2
	б) що не входять в систему Scopus		1
3.4	Кіл-сть надрукованих у календарному році статей у фахових (ВАК) журналах, збірниках		0,5
3.5	Кіл-сть надрукованих у календарному році статей у нефахових журналах, збірниках		0,2
3.6	Кількість надрукованих тез доповідей на конф-ціях:		
	а) за кордоном		0,3
	б) в Україні		0,2
3.7	Отримані у звітному році авторські свідоцтва на твір		1
3.8	Отримані у звітному році патенти на корисну модель		1
3.9	Отримані у звітному році патенти на винахід		2

	Індивідуальний показник роботи:		
	IV. Організаційна наукова робота по виконанню НДР		
4.1	Участь у підготовці та проведенні наукових, науково-практичних та науково-методичних конференцій:		
	а) університетського рівня		0,25
	б) всеукраїнських		0,5
	в) міжнародних		0,75
4.2	Участь у підготовці та проведенні 2-х турів Всеукраїнських олімпіад, конкурсів студентських робіт		0,75
4.3	Участь у виконанні науково-дослідних робіт:		
	а) кафедральних		0,2
	б) держбюджетної або госпдоговірної		0,5
	в) міжнародного проекту		0,75
4.4	Кіл-сть отриманих у звітному році наукових грантів:		
	а) на виконання регіональних, відомчих НДР		1
	б) індивідуал. грантів участі у міжнародних проектах		1
	в) виконання НДР за міжнародними програмами		3
	Індивідуальний показник роботи:		
	V. Організація науково-дослідної роботи студентів		
5.1	Підготовка студентів-переможців перших турів Всеукраїнських олімпіад та конкурсу студентських робіт		0,25
5.2	Підготовка студентів-переможців других турів Всеукраїнських олімпіад та конкурсу студентських робіт		2
5.3	Кількість наукових доповідей, з якими виступили студенти/магістри на наукових конференціях:		
	а) університетського рівня		0,25
	б) всеукраїнських		0,5
	в) міжнародних		0,75
5.4	Кіл-сть надрукованих у звітному році наукових статей у співавторстві зі студентами		1
5.5	Кіл-сть надрукованих у звітному році студентами особисто наукових статей під керівництвом викладача		1,5
	Індивідуальний показник роботи:		
	VI. Організаційна робота (викладачем заповнюється лише графа "Показник", інші графи заповнюються експертами)		
6.1	Участь у роботі як представник університету:		
	а) в експертних радах ВАК		
	б) в експертних радах ДАК		
	в) в НМК МОНмолодьспорту України		
6.2	Участь у роботі університет. спеціаліз. Вчених рад		
6.3	Виконання обов'язків вченого секретаря унів.Спецрад та Вчених рад університету, інституту/факультету, головування у Раді НДРС, Ради молодих вчених		
6.4	Посади, обов'язки з яких виконуються викладачем на громадських засадах на університетському рівні		

6.5	Встановлені випадки порушення викладачем трудової дисципліни, діючих положень по організації навчально-виховного процесу	
	Індивідуальний показник роботи:	

Викладач _____ (підпис)

Зав. кафедрою_____ (підпис)

	VII. Виховна, соціальна та профорієнтаційна робота (заповнюється експертами та студентськими органами)	
7.1	Показник результативності та якості виконання завдань під час чергування академ. групи з куратором	
7.2	Показник ефективності роботи викладача з студент. активом та органами студентського самоврядування	
7.3	Показник ефективності та результативності роботи викладача при організації та проведені виховних, культурно-масових та спортивних заходів всіх рівнів	
7.4	Показник ефективності та результативності роботи викладача в студентських гуртожитках	
7.5	Показник участі або надання організаційної допомоги в проведені різноманітних університетських громадських заходів	
7.6	Показник ефективності та результативності роботи викладача при проведені профорієнтаційних заходів	
7.7	Оцінка діяльності викладача за результатами щорічного анкетування “Викладач очами студентів”	
	Індивідуальний показник роботи:	

ДОДАТОК В

Логічна схема бази даних ІС

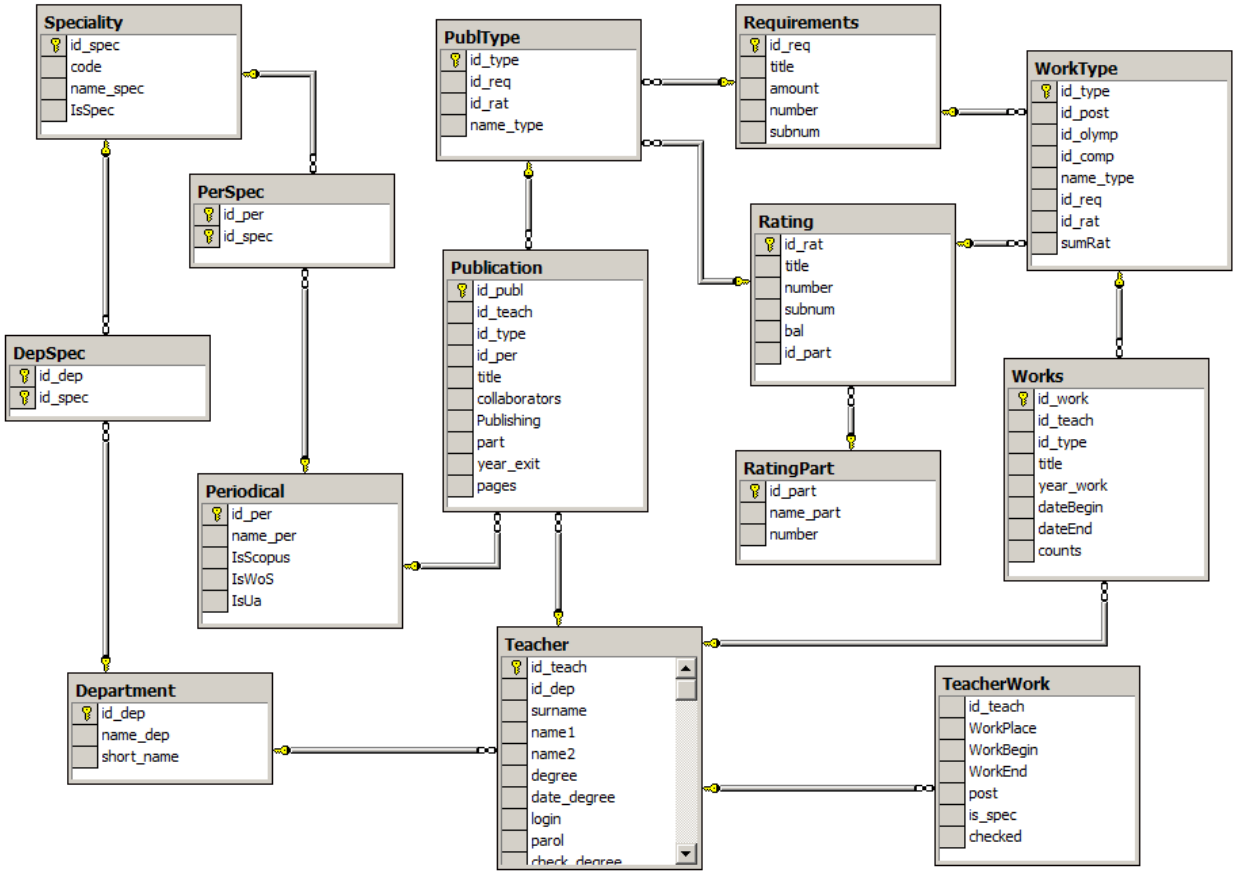

Рисунок В.1 – Логічна схема бази даних ІС «Особистий кабінет викладача»

ДОДАТОК Г

Вихідний код збережених процедур БД

```
CREATE PROC EditTeacher
@id_dep smallint,
@id_teach smallint ,
@surname varchar(30),
@name1 varchar (25),
@name2 varchar (25),
@degree varchar (12),
@k tinyint output,
@msg varchar(100) output
AS
begin
	set @k=(select k=count(*) from Teacher where id_dep=@id_dep and
surname=@surname AND
		name1=@name1 AND name2=@name2 and id_teach!= @id_teach)
	if @k>0
	set @msg='Этот преподаватель уже есть в БД'
	else
	begin
		update Teacher set
id_dep=@id_dep,surname=@surname,name1=@name1,name2=@name2,degree=@degree
		where id_teach = @id_teach
		set @msg='OK'
	end
end

go



CREATE PROC AddTeacher
@id_dep smallint,
@id_teach smallint output,
@surname varchar(30),
@name1 varchar (25),
@name2 varchar (25),
@degree varchar (12),
@k tinyint output,
@msg varchar(100) output
AS
begin
	set @k=(select k=count(*) from Teacher where id_dep=@id_dep and surname=@surname AND
		name1=@name1 AND name2=@name2)
	if @k>0
	set @msg='Этот преподаватель уже есть в БД'
	else
	begin
		insert Teacher(id_dep,surname,name1,name2,degree,login,parol)
		values(@id_dep,@surname,@name1,@name2,@degree,@surname,@name1)
		set @id_teach=(select id_teach from Teacher where id_dep=@id_dep and
surname=@surname AND
		name1=@name1 AND name2=@name2)
		set @msg='OK'
	end
end

go
```

```
CREATE proc CheckDegreeDate
@id_teach smallint,
@degree varchar(12),
@date_degree smalldatetime,
@check bit
as
begin
	if @check=0
	update Teacher set check_degree=0 where id_teach=@id_teach
	else
	update Teacher set degree= @degree, date_degree=@date_degree,
check_degree=@check
	where id_teach=@id_teach
end

go


create proc CheckTeachWork
@id_teach smallint,
@WorkPlace varchar (100),
@WorkBegin smalldatetime,
@WorkEnd smalldatetime,
@is_spec bit,
@checked bit
as
begin
	if @checked=0
	update TeacherWork set checked=0 where id_teach=@id_teach and
WorkPlace=@WorkPlace and WorkBegin=@WorkBegin
	else
	if @checked=1
	update TeacherWork set checked=1, WorkEnd=@WorkEnd,is_spec=@is_spec
	where id_teach=@id_teach and WorkPlace=@WorkPlace and WorkBegin=@WorkBegin
end

go
```

ДОДАТОК L

Вихідний код клієнтського додатку

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

namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        int x;
        System.Data.OleDb.OleDbCommand addTeachCom;
        System.Data.OleDb.OleDbCommand editTeachCom;
        System.Data.OleDb.OleDbCommand delTeachCom;
        System.Data.OleDb.OleDbCommand parTeachCom;
        System.Data.OleDb.OleDbCommand parExecCom;

        short id_dep = 1;
        bool add = true;
        int k=0;
        public Form1()
        {
            InitializeComponent();
            init_addTeachCom();
            init_editTeachCom();
            init_delTeachCom();
            init_parTeachCom();
            init_parExecCom();
        }

        private void init_parTeachCom()
        {
            parTeachCom = new System.Data.OleDb.OleDbCommand();
            parTeachCom.Connection = this.teacherTableAdapter.Connection;
            parTeachCom.CommandText = "MakeTeacherLogin";
            parTeachCom.CommandType = System.Data.CommandType.StoredProcedure;
            parTeachCom.Parameters.Add("@id_teach",
System.Data.OleDb.OleDbType.SmallInt);
            parTeachCom.Parameters.Add("@login", System.Data.OleDb.OleDbType.VarChar,
50);
            parTeachCom.Parameters.Add("@parol", System.Data.OleDb.OleDbType.VarChar,
30);
            parTeachCom.Parameters.Add("@k",
System.Data.OleDb.OleDbType.UnsignedTinyInt).Direction =
System.Data.ParameterDirection.InputOutput;
            parTeachCom.Parameters.Add("@msg", System.Data.OleDb.OleDbType.VarChar,
100).Direction = System.Data.ParameterDirection.InputOutput;
        }

        private void init_parExecCom()
        {
            parExecCom = new System.Data.OleDb.OleDbCommand();
            parExecCom.Connection = this.departmentTableAdapter.Connection;
            parExecCom.CommandText = "MakeExecLogin";
            parExecCom.CommandType = System.Data.CommandType.StoredProcedure;
            parExecCom.Parameters.Add("@id_dep", System.Data.OleDb.OleDbType.SmallInt);
            parExecCom.Parameters.Add("@login", System.Data.OleDb.OleDbType.VarChar, 50);
```

```
            parExecCom.Parameters.Add("@parol", System.Data.OleDb.OleDbType.VarChar, 30);
            parExecCom.Parameters.Add("@k",
System.Data.OleDb.OleDbType.UnsignedTinyInt).Direction =
System.Data.ParameterDirection.InputOutput;
            parExecCom.Parameters.Add("@msg", System.Data.OleDb.OleDbType.VarChar,
100).Direction = System.Data.ParameterDirection.InputOutput;
        }

        private void init_addTeachCom()
        {
            addTeachCom = new System.Data.OleDb.OleDbCommand();
            addTeachCom.Connection = this.teacherTableAdapter.Connection;
            addTeachCom.CommandText = "AddTeacher";
            addTeachCom.CommandType = System.Data.CommandType.StoredProcedure;
            addTeachCom.Parameters.Add("@id_dep", System.Data.OleDb.OleDbType.SmallInt);
            addTeachCom.Parameters.Add("@id_teach",
System.Data.OleDb.OleDbType.SmallInt).Direction =
System.Data.ParameterDirection.InputOutput;
            addTeachCom.Parameters.Add("@surname",
System.Data.OleDb.OleDbType.VarChar,30);
            addTeachCom.Parameters.Add("@name1", System.Data.OleDb.OleDbType.VarChar,25);
            addTeachCom.Parameters.Add("@name2", System.Data.OleDb.OleDbType.VarChar,25);
            addTeachCom.Parameters.Add("@degree",
System.Data.OleDb.OleDbType.VarChar,12);

            addTeachCom.Parameters.Add("@k",
System.Data.OleDb.OleDbType.UnsignedTinyInt).Direction =
System.Data.ParameterDirection.InputOutput;
            addTeachCom.Parameters.Add("@msg", System.Data.OleDb.OleDbType.VarChar,
100).Direction = System.Data.ParameterDirection.InputOutput;
        }

        private void init_editTeachCom()
        {
            editTeachCom = new System.Data.OleDb.OleDbCommand();
            editTeachCom.Connection = this.teacherTableAdapter.Connection;
            editTeachCom.CommandText = "EditTeacher";
            editTeachCom.CommandType = System.Data.CommandType.StoredProcedure;
            editTeachCom.Parameters.Add("@id_dep", System.Data.OleDb.OleDbType.SmallInt);
            editTeachCom.Parameters.Add("@id_teach",
System.Data.OleDb.OleDbType.SmallInt);
            editTeachCom.Parameters.Add("@surname", System.Data.OleDb.OleDbType.VarChar,
30);
            editTeachCom.Parameters.Add("@name1", System.Data.OleDb.OleDbType.VarChar,
25);
            editTeachCom.Parameters.Add("@name2", System.Data.OleDb.OleDbType.VarChar,
25);
            editTeachCom.Parameters.Add("@degree", System.Data.OleDb.OleDbType.VarChar,
12);

            addTeachCom.Parameters.Add("@k",
System.Data.OleDb.OleDbType.UnsignedTinyInt).Direction =
System.Data.ParameterDirection.InputOutput;
            addTeachCom.Parameters.Add("@msg", System.Data.OleDb.OleDbType.VarChar,
100).Direction = System.Data.ParameterDirection.InputOutput;
        }

        private void init_delTeachCom()
        {
            delTeachCom = new System.Data.OleDb.OleDbCommand();
            delTeachCom.Connection = this.teacherTableAdapter.Connection;
            delTeachCom.CommandText = "DelTeacher";
            delTeachCom.CommandType = System.Data.CommandType.StoredProcedure;
            delTeachCom.Parameters.Add("@id_teach",
System.Data.OleDb.OleDbType.SmallInt);
```

```
            delTeachCom.Parameters.Add("@m",
System.Data.OleDb.OleDbType.UnsignedTinyInt);
            delTeachCom.Parameters.Add("@k",
System.Data.OleDb.OleDbType.UnsignedTinyInt).Direction =
System.Data.ParameterDirection.InputOutput;
            delTeachCom.Parameters.Add("@msg", System.Data.OleDb.OleDbType.VarChar,
100).Direction = System.Data.ParameterDirection.InputOutput;
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'dataSet1.executor' table. You
can move, or remove it, as needed.
            this.executorTableAdapter.Fill(this.dataSet1.executor,id_dep);
            // TODO: This line of code loads data into the 'dataSet1.Department' table.
You can move, or remove it, as needed.
            this.departmentTableAdapter.Fill(this.dataSet1.Department);
            this.teacherTableAdapter.Fill(this.dataSet1.Teacher, id_dep);

        }

        private void depComboBox_SelectedIndexChanged(object sender, EventArgs e)
        {
            logTextBox.Enabled = true;
            x = depComboBox.SelectedIndex;
            if (x >= 0)
            {
                id_dep = (short)depComboBox.SelectedValue;
            }
        }

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

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

        private void logOKbutt_Click(object sender, EventArgs e)
        {
            string a = parTextBox.Text, d=logTextBox.Text;
            int n = a.Length;
            char[] b  = new char[n];
            for (int i = 0; i < n; i++)
                {
                    ushort x = (ushort)(a[i]);
                    b[i] = (char)(x+10);
                }
            string c=new string(b);

            this.executorTableAdapter.Fill(this.dataSet1.executor, id_dep);

            if ((d.Equals(dataGridView2.CurrentRow.Cells[1].Value)) &&
(b.SequenceEqual((string)(dataGridView2.CurrentRow.Cells[2].Value))))
                {
                this.teacherTableAdapter.Fill(this.dataSet1.Teacher, id_dep);
                teachGridView.Visible = true;
             teachGridLabel.Visible = true;
                parTextBox.Visible = false;
             parLabel.Visible = false;
                logLabel.Visible = false;
```

```
      logTextBox.Visible = false;
        logOKbutt.Visible = false;
        depComboBox.SelectedIndex = x;
      depComboBox.Enabled = false;
        addTeachButt.Visible = true;
      editTeachButt.Visible = true;
        delTeachButt.Visible = true;
      changeTeachButt.Visible = true;
    }
    changebutt.Enabled = true;

}

private void textBox6_TextChanged(object sender, EventArgs e)
{
    textBox10.Enabled = true;
}

private void textBox10_TextChanged(object sender, EventArgs e)
{
    textBox11.Enabled = true;
}

private void changebutt_Click(object sender, EventArgs e)
{
    logPanel.Visible = true;
      allElem(false);
}

private void allElem(bool a)
{
    changebutt.Enabled = a;
    changeTeachButt.Enabled = a;
    delTeachButt.Enabled = a;
    editTeachButt.Enabled = a;
    addTeachButt.Enabled = a;
}

private void execCancelButt_Click(object sender, EventArgs e)
{
    logPanel.Visible = false ; allElem(true);
}

private void addTeachButt_Click(object sender, EventArgs e)
{
    add = true;
    surnameTextBox.Text = "";
      name1TextBox.Text = "";
    name2TextBox.Text = "";
      degreeTextBox.Text = "";
    teachPanel.Visible = true;allElem(false);
}

private void teachCancelButt_Click(object sender, EventArgs e)
{
    teachPanel.Visible =false ;
      allElem(true);
}

private void changeTeachButt_Click(object sender, EventArgs e)
{
    teachLogPanel.Visible = true;
      allElem(false);
}
```

```
        private void teachLogCancelButt_Click(object sender, EventArgs e)
        {
            teachLogPanel.Visible =false ;
              allElem(true);
        }

        private void editTeachButt_Click(object sender, EventArgs e)
        {
            add = false;
            surnameTextBox.Text = (string)(teachGridView.CurrentRow.Cells[0].Value);
            name1TextBox.Text = (string)(teachGridView.CurrentRow.Cells[1].Value);
            name2TextBox.Text = (string)(teachGridView.CurrentRow.Cells[2].Value);
            degreeTextBox.Text = (string)(teachGridView.CurrentRow.Cells[3].Value);
            teachPanel.Visible = true;
              allElem(false);
        }

    }
}
```