

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

_____ Олександр КОВАЛЬ

«___» _____ 2021 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Геометричне моделювання в
інформаційних системах»**

спеціальності 122 «Комп'ютерні науки»

**на тему: «Система аналізу результатів проведення атестації студентів на
кафедрі на основі Microsoft Dynamics 365 Business Central»**

Виконав:

студент IV курсу, групи ТР-71

Чорненький Микола Володимирович

Керівник:

Старший викладач,

Гурін Артем Леонідович

Консультант:

Рецензент:

Інженер з розробки програмного забезпечення, к.т.н.,

Дідик Марія Сергіївна

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

Студент (-ка) _____

Київ — 2021 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки: 122 Комп'ютерні науки

Спеціалізація: Геометричне моделювання в інформаційних системах

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр КОВАЛЬ
(підпис)

” ” _____ 2021р.

ЗАВДАННЯ

на дипломну роботу студенту

Чорненькому Миколі Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи Система аналізу результатів проведення атестації студентів на кафедрі на основі Microsoft Dynamics 365 Business Central

керівник роботи _____ Гурін Артем Леонідович, к.е.н.

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від "24" травня 2021р. № **1267-с**

2. Строк подання студентом роботи 12.06.2021р.

3. Вихідні дані до роботи мова програмування AL, платформа Microsoft Dynamics 365 Business Central, середовище розробки Visual Studio Code

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) розробити систему для ведення відомостей успішності студентів та аналізу атестацій на кафедрі ; проаналізувати існуючі системи для ведення відомостей та аналізу атестацій; розробити функціонал для друкування звітів.

5. Перелік ілюстративного матеріалу

Актуальність, Постановка задачі, Існуючі системи, Платформа Microsoft Dynamics Business Central, Архітектура системи, Алгоритм виявлення та ідентифікації маневру, Діаграма класів, Діаграма розгортання, Модель бази даних, Діаграма прецедентів, Сторінки реєстрації та авторизації, Інтерфейс головної сторінки, Заповнення відомостей успішності студентів, Сторінка аналізу атестацій, Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання ” 9 ” березня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	09.03.2021	
2.	Вивчення та аналіз задачі	10.03.2021-20.12.2021	
3.	Розробка архітектури та загальної структури системи	20.03.2021-28.03.2021	
4.	Розробка структур окремих підсистем	28.03.2021-10.04.2021	
5.	Програмна реалізація системи	10.04.2021-6.05.2021	
6.	Оформлення пояснювальної записки	06.05.2021-10.05.2021	
7.	Захист програмного продукту	11.05.2021	
8.	Передзахист	25.05.2021	
9.	Захист	14.06.2021	

Студент _____ Чорненко М.В.
(підпис) (прізвище та ініціали,)

Керівник роботи _____ Гурін А.Л.
(підпис) (прізвище та ініціали,)

АНОТАЦІЯ

У роботі розглянуті теоретичні та практичні аспекти розробки програмного продукту для аналізу результатів атестації студентів на кафедрі.

Метою роботи є створення кроссплатформенного застосунку на основі Microsoft Dynamics Business Central для аналізу результатів атестації студентів на кафедрі.

Для досягнення мети розроблено додаток для Microsoft Dynamics Business Central який включає в себе базу даних, алгоритми для заповнення оцінок студентів та їх атестацій та генерацію звітів успішності студентів.

Пояснювальна записка складається зі вступу, шести розділів, висновку, списку використаних джерел; містить 50 сторінок, 32 рисунки, 1 таблицю та 1 додаток. Список використаних джерел включає 9 бібліографічних найменувань.

Ключові слова: кроссплатформенний застосунок, аналіз успішності студентів, атестації студентів, Microsoft Dynamics Business Central.

ABSTRACT

Theoretical and practical aspects of software product development for the analysis of students' certification results at the department are considered in the work.

The aim of the work is to create a cross-platform application based on Microsoft Dynamics Business Central for analyzing the results of student certification at the department.

To achieve this goal, an application was developed for Microsoft Dynamics Business Central, which includes a database, algorithms for filling out student grades and their certifications, and generating student performance reports.

The explanatory note consists of an introduction, six sections, a conclusion, a list of sources used; contains 50 pages, 37 figures, 1 table and 1 appendix. The list of used sources includes 9 bibliographic names.

Keywords: cross-platform application, student performance analysis, student certification, Microsoft Dynamics Business Central.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП.....	8
1. Задача розробки системи аналізу атестацій	10
2. Методи розробки системи атестацій	11
3. Засоби розробки.....	16
3.1. Середовище розробки Visual Studio Code	16
3.2. Система керування базою даних MSSQL	17
3.3. Технологія розмітки RDLC	18
3.4. Мова програмування AL.....	19
3.5. ERP Microsoft Dynamics Business Central	21
4. Опис програмної реалізації	25
4.1. Створення моделі бази даних.....	25
4.2. Створення логіки системи	28
4.3. Створення відображення	29
4.4. Створення звітів.....	32
5. Інтеграція системи атестацій	34
6. Робота користувача з системою	37
6.1. Налаштування ролі користувача.....	37
6.2. Сутності системи та робота з ними	38
Висновки	49
ДОДАТОК А.....	52
ДОДАТОК Б.....	54
ДОДАТОК В.....	62

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API — прикладний програмний інтерфейс

БД — база даних

REST — Representational State Transfer, підхід до архітектури мережеских протоколів

СКБД — система керування базами даних

URL — Uniform Resource Locator — єдиний вказівник на ресурс

CRM система — Customer Relationship Management, система управління взаємовідносинами з клієнтами

ERP система — Enterprise Resource Planning System, система планування та управління ресурсами

CRUD — Create, Read, Update, Delete, підхід до роботи з даними що передбачає можливість їх створення, зчитування, видалення та оновлення

ВСТУП

У вік технологій дуже багато процесів зазнають кардинальних змін. Ці зміни включають в себе реорганізацію, автоматизацію, аналіз для покращення результатів, зміну підходу до виконання поставленого завдання, тощо. Ці зміни здебільшого спрямовані на:

- пришвидшення та спрощення процесу виконання поставлених завдань і як наслідок їх здешевлення;
- підвищення якості виконання завдання шляхом виключення людського фактору та додавання перевірок на правильність та актуальність введених даних;
- зменшення кількості складної та нецікавої роботи та збільшення можливостей для аналізу робочих даних.

Як один з прикладів вдалих та розповсюджених автоматизацій можна розглянути CRM ERP системи. Ці системи широко використовуються для управління малим, середнім та великими бізнесами та автоматизують наступні завдання:

- ведення контролю закупівель і продажів;
- контроль товарів на складі;
- розрахунок податків;
- розрахунок логістичних витрат;
- контроль людських ресурсів;
- аналіз та передбачення:
 - заробітку;
 - збитків;
 - навантаження;
 - використання ресурсів.

Окрім цих основних цілей більшість лідерів індустрії дають можливість

розширювати та змінювати функціонал CRM ERP систем для збільшення поля їх можливостей. Це обумовлено надзвичайно великою кількістю варіантів організації установи та областей застосування цих продуктів.

Було запропоновано створити систему аналізу результатів проведення атестації студентів на кафедрі. Для реалізації цього завдання була обрана платформа одного з безсумнівних лідерів IT індустрії загалом та індустрії CRM ERP зокрема, а саме рішення компанії Microsoft з продуктом Dynamics 365 Business Central.

В роботі проведено аналіз проблеми автоматизації аналізу результатів проведення атестації (розділ 2). Описано розробку системи заповнення та аналізу атестаційних даних, де визначені важливі структури, форми, дії, які безпосередньо пов'язані з процесом автоматизації (розділ 3), розроблено архітектуру програмної системи та обґрунтовано методи реалізації для кожного модуля системи (розділ 4).

Результатом роботи є працююча система, яка є гнучкою та здатна легко розширюватись для додавання нового функціоналу. Вона представлена у вигляді прикладного програмного інтерфейсу (API), яка є частиною клієнт-серверної архітектури.

1. Задача розробки системи аналізу атестацій

Метою розробки є створення системи для аналізу та ведення атестації для кафедри. Система є роздробленою на декілька частин, що полегшує внесення в неї змін та подальшу підтримку. Система має наступні основні модулі:

- Студент;
- Викладач;
- Група;
- Предмет;
- Оцінки;
- Атестації.

Завданням системи є надати можливість ведення відомостей, на основі яких можуть бути виставлені атестації. На основі атестацій студентів може бути проведено аналіз і внесено корективи в навчальний процес.

Для коректної роботи системи необхідно заповнити дані в усіх вищезазначених модулях, що дасть можливість аналізувати:

- успішність студента;
- успішність групи;
- компетентність викладача;
- рівень навчальної програми.

Система має бути достатньо гнучкою для можливості її розширення, в тому числі для роботи з іншими зовнішніми системами, зокрема з використовуваною на даний момент системою КПП Кампус.

2. Методи розробки системи атестацій

Виконання аналізу атестацій, безумовно, є дуже важливою частиною навчального процесу, адже це допомагає корегувати та підвищувати рівень студентів та викладачів, оновлювати та додавати до навчальної програми актуальні теми та розділи, аналізувати зацікавленість студентів в тих чи інших предметах та викладачах. Також даний систематичний підхід до аналізу цієї проблеми підвищить швидкість виставлення атестацій та заповнення навчальних відомостей.

Першочергово для вирішення цієї задачі необхідно провести аналіз існуючих рішень. Найбільш близькими з рішень є «1С Підприємство» та Campus KPI.

«1С Підприємство» — це фреймворк для побудови бізнес-додатків для різних галузей і доменів. 1С Підприємство значно прискорює розробку бізнес-додатків, оскільки багато функцій, пов'язані з додатками, вже реалізовані і готові до використання. Розробляється російським виробником 1С з 1991 року.

Оскільки і платформа розробки, і додатки призначалися для ринку Співдружності Незалежних Держав (СНД), використовувалася мова програмування, запозичена з російської. Тільки з орієнтацією на міжнародний ринок «1С: Підприємство» було розширено діалектом, заснованим на англійській мові і, відповідно, на латинському наборі символів. Для підтримки проектів за межами СНД в 2005 році була заснована дочірня компанія «1С Німеччина» з головним офісом в Німеччині. Приклад інтерфейсу «1С Підприємство» зображено на рисунку 2.1.

1С:Предприятие 8. Бухгалтерия сельскохозяйственно... (1С:Предприятие)

Поступление товаров и услуг 0000-000001 от 05.05.2019 12:00:00

Провести и закрыть Записать Провести АТ КТ Создать на основании Еще ?

Скрыть основные реквизиты

Вид операции: Покупка, комиссия

Номер: 0000-000001 от: 05.05.2019 12:00:00

Организация: Труд

Склад: Ферма

Подразделение: Ферма №1

Вид движения животных: Куплено, получено в обмен Поступление животных

Контрагент: АгроКомплект

Договор: Договор №1

Зачет аванса: Автоматически

Цена не включает НДС. Цена за массу

МОЛ: Иванов Игорь Витальевич

Товары (1) Услуги Агентские услуги Возвратная тара Счета расчетов Дополнительно

Добавить Подбор

N	Номенклатура	Количество	Цена без НДС	Счет учета	Ставка НДС
1	Нетели	Голов: 10,000	0,35	11.1	20%
	0,35	Масса: 2 500,000			

Всего: 1 050,00 BYN НДС (в т.ч.): 175,00 BYN

Рисунок 2.1. – Интерфейс «1С Підприємство»

Серед плюсів «1С Підприємство» можна виділити його ціну, адже на етапі ліцензування він має найнижчу вартість та готовий функціонал для ведення фінансового обліку.

З мінусів можна назвати його невідповідність актуальним на сьогодні технологічним стандартам, неможливість інтеграції зі сторонніми системами, перевантажений користувацький інтерфейс та повільну швидкість роботи.

Система Camrus KPI, що використовується на даний момент для ведення відомостей з успішності студентів, була створена і приведена до сьогоднішнього вигляду в 2015 році. Система створена потужностями Київського політехнічного інституту й активно використовується дотепер.

Приклад інтерфейсу Camrus KPI зображено на рисунку 2.2.

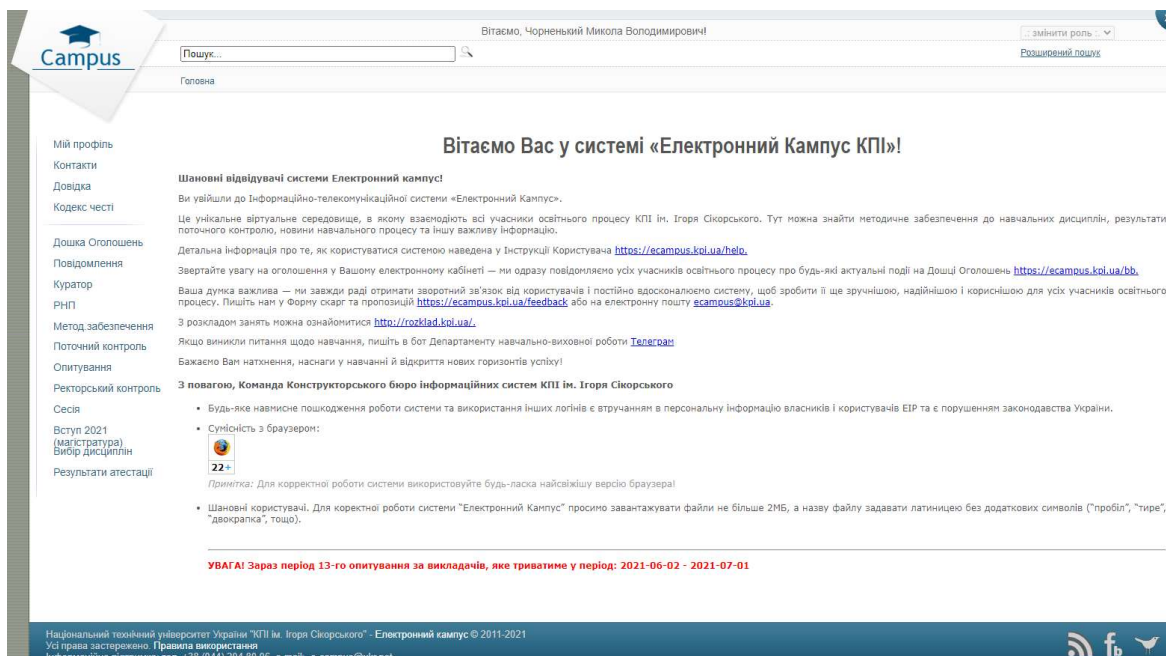


Рисунок 2.2. – Інтерфейс Campus KPI

Серед плюсів Campus KPI можна виділити готову та заповнену базу даних, високий попит користувачів та можливість ведення відомостей студентів.

З мінусів можна назвати його швидкість роботи та неможливість проводити аналіз атестацій автоматизованим шляхом.

На основі проведеного порівняння можна зробити висновок що ні одна з порівнюваних систем не відповідає умовам для вирішення завдання аналізу атестацій. Саме через це при виконанні цього завдання була обрана основа в вигляді Microsoft Dynamics Business Central.

Microsoft Dynamics Business Central використовує встановлений набір засобів розробки, котрий описано в наступному розділі.

Базуючись на вимогах до системи, необхідно розробити предметну область, що задовольнить користувацькі вимоги. За результатами проведеного аналізу була створена діаграма прецедентів, що описує потрібний користувачу функціонал (рисунок 2.3).

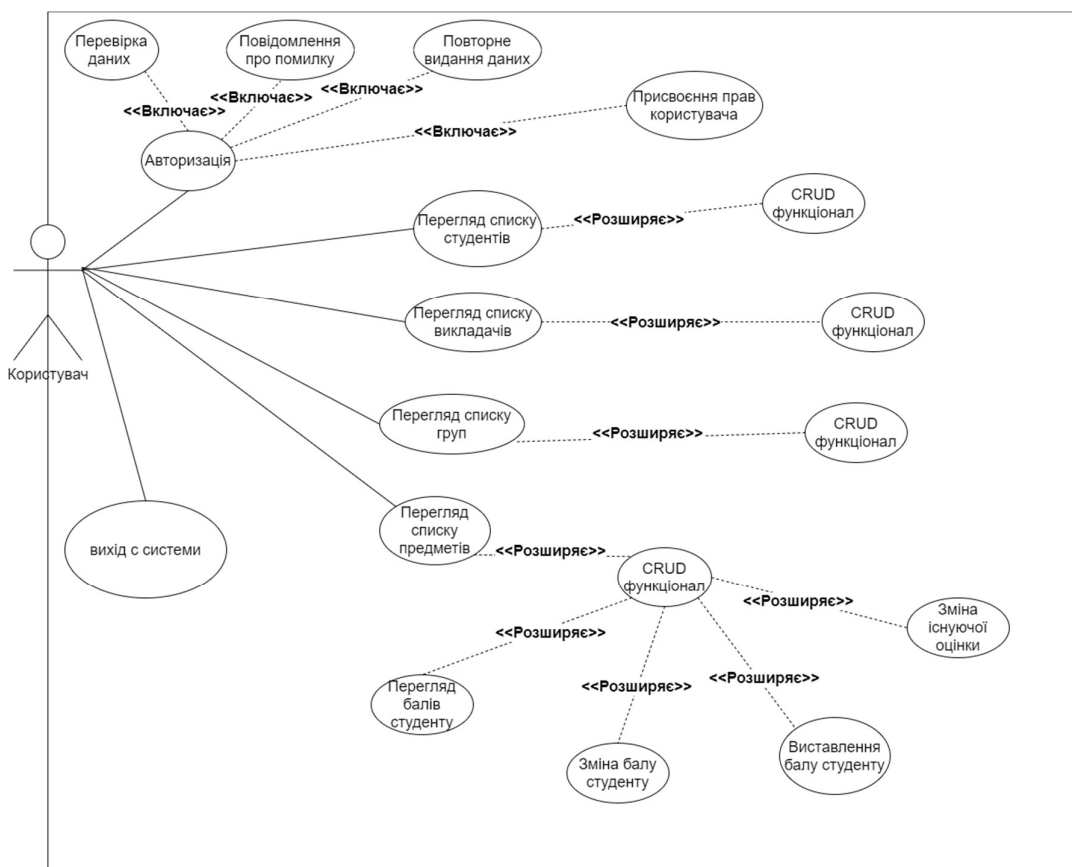


Рисунок 2.3. – Діаграма прецедентів

Діаграма класів є однією з найчастіше використовуваних діаграм UML і призначена для представлення класів в об'єктно-орієнтованому аналізі, проектуванні і програмуванні в ясній і зрозумілій формі.

Однією з областей, де діаграми класів використовуються найбільш ефективно, є об'єктно-орієнтоване проектування: Класи, їхні властивості та поведінка, яку вони можуть висловлювати, визначаються на реальних прикладах і малюються у вигляді діаграм класів. Між класами можуть існувати різні відносини. Наприклад, клас може використовувати інший клас, в цьому випадку зв'язок між двома класами позначається пунктирною лінією. Або клас може складатися з іншого класу (композиція), в цьому випадку до лінії зв'язку між двома класами додається порожній ромб. Розроблену діаграму класів зображено на рисунку 2.4.

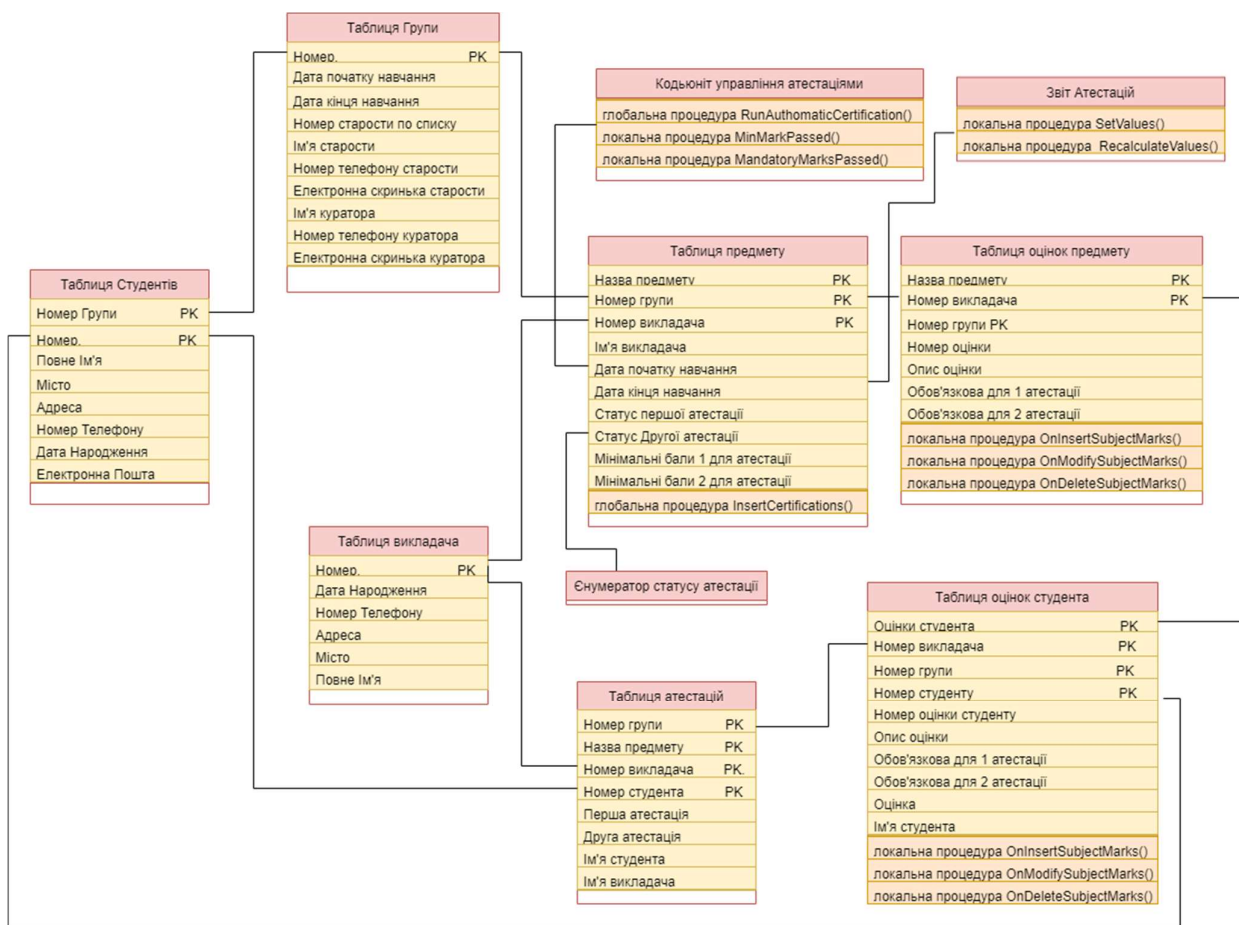


Рисунок 2.4. – Діаграма класів

3. Засоби розробки

Успіх програмних продуктів безпосередньо залежить від правильного вибору засобів розробки. Інструменти розробки повинні відповідати всім вимогам і не можуть їх диктувати.

Впровадження цього програмного продукту базується на Microsoft Dynamics Business Central. Мова програмування AL охоплює вимоги до створення користувальницьких інтерфейсів, операцій з базами даних та алгоритмів програмування. Операційна система Windows 10 — це гібридна операційна система на базі ядра від Microsoft.

Середовищем для розробки лейауту був обраний Report Builder — програма від Microsoft, що забезпечує можливість виведення агрегованих даних в мові програмування AL та змінювати оформлення цих даних.

Як Середовище розробки обрано Visual Studio Code — продукт Microsoft, який має вбудовані плагіни підтримки для використання з Microsoft Dynamics Business Central, мовою програмування AL та мовою розмітки RDLC.

3.1. Середовище розробки Visual Studio Code

Visual Studio Code — це редактор вихідного коду, розроблений компанією Microsoft для Windows, Linux і macOS. Він включає підтримку налагодження, інтегрований контроль Git, підсвічування синтаксису, інтелектуальне завершення коду, фрагменти і рефакторинг коду. Він також налаштовується, тому користувачі можуть змінити тему редактора, поєднання клавіш і переваги. Він є безкоштовним і з відкритим вихідним кодом, хоча офіційно завантаження здійснюється в рамках пропрієтарного програмного забезпечення і включає функції, налаштовані Microsoft.³

Visual Studio Code заснований на Electron, фреймворку, що використовується для розгортання Chromium і Node.js як настільних додатків, який працює на движку

проектування Blink. Хоча в програмі використовується фреймворк Electron, вона не використовує Atom, а замість нього застосовується той же компонент редактора (Monaco), який використовується в Visual Studio Team Services (раніше називалася Visual Studio Online) .

Visual Studio Code підтримує кілька мов програмування і набір функцій, які можуть бути доступні або недоступні для даної мови, як показано в таблиці нижче. Багато можливості Visual Studio Code не розкриваються через меню або призначений для користувача інтерфейс. Швидше, доступ до них здійснюється через командну палітру або через файли .json (наприклад, переваги користувача). Палітра команд — це інтерфейс командного рядка. Однак він зникає, якщо користувач клацає за його межами або натискає комбінацію клавіш на клавіатурі, щоб взаємодіяти з чимось за його межами. Це також відноситься до команд, віднімають час. Коли це відбувається, виконувана команда скасовується.

У ролі редактора вихідного коду Visual Studio Code дозволяє змінити кодову сторінку, на якій зберігається активний документ, символ, що визначає новий рядок (вибір між LF і CRLF), і мова програмування активного документа.

Окрім функцій, які Visual Studio Code включає в себе з коробки він дозволяє встановлювати розширення. Розширення VS Code дозволяють додавати мови, відладчики й інструменти для підтримки робочого процесу розробки. Багата модель розширюваності VS Code дозволяє авторам розширень підключатися безпосередньо до призначеного для користувача інтерфейсу VS Code і надавати функціональність через ті ж API, які використовуються в VS Code.

3.2. Система керування базою даних MSSQL

Microsoft SQL Server — це система керування базами даних, розроблена корпорацією Microsoft. Як сервер даних він виконує основну функцію зберігання та надання даних у відповідь на запити інших програм, які можуть працювати на тому самому сервері та в мереж.

Для запитів використовується мова Transact-SQL, яку спільно створили Microsoft та Sybase. Transact-SQL — це реалізація стандарту ANSI / ISO із розширеною структурованою мовою запитів SQL. Використовується для малих та середніх баз даних та великих баз даних на рівні підприємств. Протягом багатьох років він успішно конкурує з іншими системами управління базами даних.

SQL Server 2005 має вбудовану підтримку .NET Framework. Таким чином, ви можете використовувати повний набір бібліотек для .NET Framework для написання процедур збережених баз даних на будь-якій мові платформи .NET. На відміну від інших процесів .NET Framework виділяє додаткову пам'ять і створює елементи керування SQL Server без використання вбудованих інструментів Windows. Порівняно із загальним алгоритмом Windows це покращує продуктивність, оскільки алгоритм розподілу ресурсів спеціально налаштований для використання в архітектурі SQL Server.

3.3. Технологія розмітки RDLC

RDLC звіти — це клієнтські звіти, які ніде не розміщуються. Зайвий с в назві означає «Client». Як правило, це розширення мови RDL, призначене для використання тільки в клієнтських додатках Visual Studio. Він існує в Visual Studio, коли ви додаєте елемент reporting .

Переваги звітів RDLC:

- Ви можете підключити сервіс wcf набагато простіше до набору даних.
- Ви маєте більше контролю над набором даних і можете використовувати класи POCO, заповнені об'єктами Entity framework або ADO.NET безпосередньо, а також самі таблиці. Ви можете мавпувати з даними для їх оптимізації, перш ніж прив'язувати їх до звіту.
- Ви можете налаштувати зовнішній вигляд більше за допомогою надбудов безпосередньо в коді позаду.

Мінуси звітів RDLC:

- Вам потрібно обробляти параметри самостійно, і хоча ви можете реалізувати методи обгортки, щоб допомогти роботі з низом звіту, це трохи більше, ніж очікувалося.

3.4. Мова програмування AL

AL — це мова програмування, що використовується для керування даними в базі даних Dynamics 365 Business Central (наприклад, для видалення, вставки та модифікації записів), а також для керування виконанням різних об'єктів програми (наприклад, сторінок, звітів або кодових одиниць).

За допомогою AL ви можете створювати бізнес-правила, щоб переконатись, що дані, котрі зберігаються в базі даних, є значущими та сумісними із способом ведення бізнесу клієнтами.

Мова програмування дозволяє контролювати реалізацію об'єктів та взаємодіяти з користувачами. Крім того, мова програмування містить набір заздалегідь визначених функцій, які ви можете використовувати під час написання коду, але ви також можете розробити власні функції. В AL ви можете мати тригери подій або функціональні тригери.

Кожен тип об'єкта в AL (таблиця, сторінка, CodeUnit, звіт тощо) має активатори подій, і їх імена починаються з On, такі як OnInsert, OnModify або OnDelete. Ці тригери спрацьовують, коли відбувається подія. Доступні тригери залежать від типу об'єкта. Ви також можете написати власні функції в об'єкті, і ці спеціальні функції будуть доступні в об'єкті за допомогою тригерів функцій. Ці користувацькі функції називаються процедурами.

Як і інші мови програмування, AL дозволяє створювати змінні, які можна використовувати у функціях для обчислення або зберігання тимчасових даних.

Мова програмування AL є об'єктно-базованою, термін «об'єктно-базована мова» може бути використаний у технічному сенсі для опису будь-якої мови програмування, що використовує ідею інкапсуляції та маніпулювання станом

всередині об'єкта. Об'єктно-орієнтовані мови не вимагають підтримки успадкування або підтипу, але їх також називають об'єктно-базованими. Об'єктно-базовані мови, які не підтримують успадкування або підтипівання, як правило, не вважаються справжніми об'єктно-орієнтованими мовами. Приклади об'єктно-орієнтованих мов у хронологічному порядку включають Simula, Smalltalk, C ++ (об'єктна модель базується на Simula), Objective-C (об'єктна модель базується на Smalltalk), Eiffel, Hojo (раніше REALbasic), Python, Ruby, Java, Visual Basic .NET, C # та Fortran 2003. Приклади об'єктно-орієнтованих, але не об'єктно-орієнтованих мов, включають попередні версії Ada, Visual Basic (VB), JavaScript та Fortran 90.

На відміну від них, терміни об'єктно-орієнтовані та об'єктно-базовані часто використовуються як взаємовиключні заміни, а не як категорії, що перекриваються. Іноді термін об'єктно-орієнтований застосовується до мов програмування на основі прототипів. Справжнім об'єктно-орієнтованим мовам не вистачає класів, але об'єкти безпосередньо успадковують свій код та дані від інших об'єктів шаблону. Прикладом поширеної мови сценаріїв на основі прототипів є JavaScript.

Іноді термін об'єктно-орієнтований застосовується до мов програмування на основі прототипів, справжніх об'єктно-орієнтованих мов, яким бракує класів, але в яких об'єкти натомість успадковують свій код та дані безпосередньо від інших об'єктів шаблону. Прикладом часто використовуваної мови сценаріїв на основі прототипів є JavaScript.

Як об'єктно-орієнтовані мови, так і об'єктно-базовані (як на основі класів, так і на основі прототипів) можуть статично перевірятися за типом. Статична перевірка мов на основі прототипів може бути складною, оскільки ці мови часто дозволяють динамічно розширювати об'єкти з новою поведінкою та навіть змінювати їх батьківський об'єкт (від якого вони успадковуються) під час виконання.

3.5. ERP Microsoft Dynamics Business Central

Система планування ресурсів підприємства (ERP) — це інформаційна система підприємства (СНД), призначена для автоматизації обліку та управління. Як правило, ERP-системи побудовані на модульній основі і певною мірою охоплюють усі ключові процеси компанії. Класична система ERP забезпечує управління завданнями:

- Фінансовий менеджмент;
- Управління виробництвом;
- Управління запасами та управління розподілом;
- Управління продажами та маркетингом;
- Управління утриманням клієнтів;
- Управління поставками; управління проектами;
- Управління послугами;
- Управління процедурами;
- Контроль якості продукції.

Microsoft Business Central має всі вищеперечислені блоки системи та надає можливості повністю змінювати їх та додавати нові. Залагальноприйнятою практикою є використання ERP систем, функціонал яких хоча б на 70% співпадає з функціоналом планованого результату. В цьому плані Microsoft Business Central має абсолютно вигравну позицію в порівнянні з конкурентами через його обширний базовий функціонал та можливість змінювати його. На рисунку 3.1 зображено інтерфейс Microsoft Dynamics Business Central.

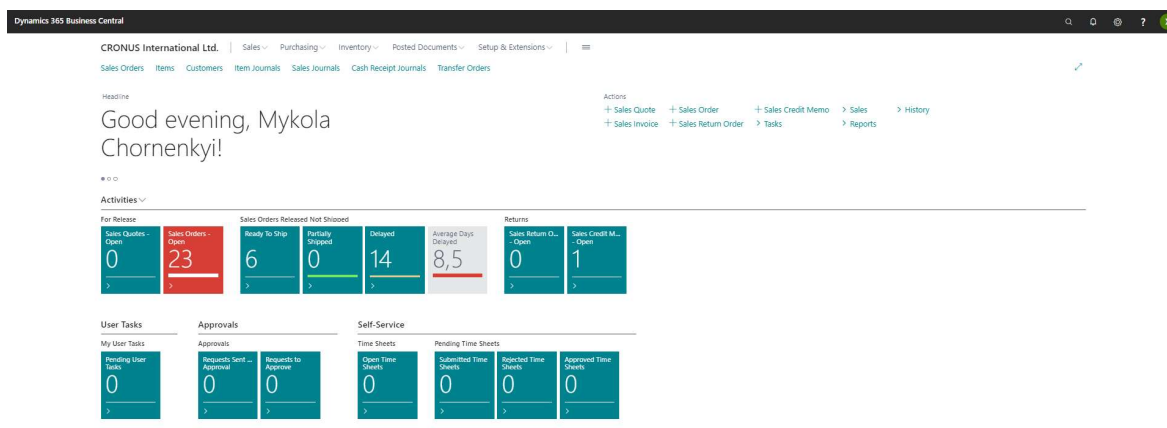


Рисунок 3.1. – Інтерфейс Microsoft Dynamics Business Central

Одним з великих плюсів вищеописаної ERP системи є її кросплатформеність. Оскільки Dynamics 365 Business Central може бути інтегрований в інші додатки і через інші платформи і пристрої, його функції працюють в повному обсязі. Іншими словами існують наступні можливості для використання Microsoft Dynamics Business Central:

- Пропонує інтеграцію з Office 365, починаючи з Outlook і Excel. Office 365 — це онлайн-версія Microsoft Office, яка складається з пакету програм для компаній, спрямованих на спільну роботу і одночасно.
- Пропонує інтеграцію з такими браузерами, як Edge, Internet Explorer, Chrome, Firefox або Safari.
- Існує версія програми для Android, яка працює на смартфоні і планшеті.
- Існує у версії і для iPhone і iPad.
- Працює як настільний додаток для Windows (комп'ютер і планшет).

На рисунку 3.2 зображена діаграма кросплатформенних можливостей Microsoft Dynamics Business Central.

Business Central

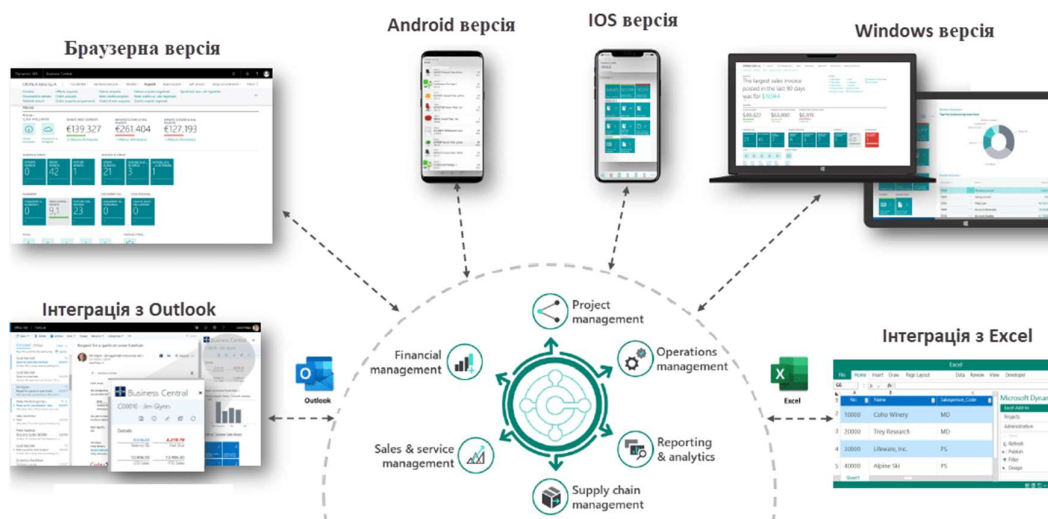


Рисунок 3.2. – Діаграма можливостей Microsoft Dynamics Business Central

Проте вищеназвані можливості інтеграції між Microsoft Outlook та Microsoft Excel не єдині. Business Central підтримує два типи веб-сервісів: SOAP і OData. Веб-служби - це легкий, стандартний для галузі спосіб зробити функціональність програми доступною для різних зовнішніх систем і користувачів. Розробники можуть створювати та публікувати функціональність у вигляді веб-сервісів, де вони розкривають сторінки, блоки коду або запити, і навіть розширювати веб-сервіс сторінки за допомогою блоку коду розширення. Коли об'єкти Business Central публікуються як веб-сервіси, вони відразу ж стають доступними в мережі.

Веб-сервіси Business Central не мають статичних даних і не зберігають значення глобальних змінних або кодових одиниць одного примірника між викликами.

Веб-сервіси SOAP забезпечують повну гнучкість при створенні сервісів, орієнтованих на роботу. Вони забезпечують сумісність з промисловими стандартами. Windows Communication Framework (WCF) підтримує сервіси SOAP з моменту свого початкового випуску в .NET Framework 3.0, а більш пізні випуски .NET Framework додали додаткову підтримку і прив'язки за замовчуванням, щоб

полегшити створення сервісів SOAP за допомогою WCF.

Стандарт OData добре підходить для додатків веб-сервісів, яким потрібна єдиний, гнучкий, універсальний інтерфейс для надання клієнтам операцій створення, зчитування, оновлення, видалення (CRUD) в табличній моделі даних. OData менше підходить для додатків, які в основному орієнтовані на RPC або в яких операції з даними обмежені певними запропонованими шаблонами. OData підтримує сервіси даних на основі Representational State Transfer (REST), які дозволяють публікувати і редагувати ресурси, ідентифіковані за допомогою уніфікованих ідентифікаторів ресурсів (URI) і певні в абстрактної моделі даних (EDM), веб-клієнтами в корпоративних мережах і через Інтернет за допомогою простих повідомлень Hypertext Transfer Protocol (HTTP). Сервіси OData є легковажними, їх функціональність часто вказується безпосередньо в URI. В таблиці 1 наведено порівняння можливостей SOAP та ODATA веб сервісів.

Тип Об'єкту	SOAP	ODATA
Page	створення, зчитування, оновлення, видалення	створення, зчитування, оновлення, видалення
Codeunit	Підтримується	Підтримується без передачі інформації
Queries	Не підтримується	Підтримується тільки для зчитування

Таблиця 1. – Порівняння можливостей SOAP та ODATA веб сервісів

В розгляді можливої інтеграції з системою Campus KPI найкращою звязкою буде SOAP вебсервіс з типом об'єкту Codeunit, адже через різні цілі систем вони мають різні структури баз даних. Це робить неактуальним використання ODATA сервісів, а використання іншого об'єкта буде викликати створення буферних прокладок для перестворення даних.

4. Опис програмної реалізації

Для створення системи було використано середовища: Visual Studio Code, Microsoft Report Builder та плагіни Visual Studio Code. Всі частини системи загреговані мовою AL на платформі .NET Core у середовищі Visual Studio Code з використаннями плагінів “AL code snippets by Waldo”, “AL code beautifier” та “AL Language”. Це надає наступні переваги:

- підвищена швидкість взаємодії між підсистемами;
- швидший процес встановлення програми;

Фінальним результатом розробки системи є файл типу app, який завантажуються в Microsoft Dynamics 365 Business Central і одразу готовий до роботи.

4.1. Створення моделі бази даних

Мова програмування AL надає можливість для створення бази даних її засобами. Це дозволяє розширювати існуючі бази даних додатками сторонніх розробників, підвищувати швидкість роботи системи та облегшувати розробку програмного забезпечення.

На рисунку 4.1 зображено всі створені таблиці за допомогою мови програмування AL.

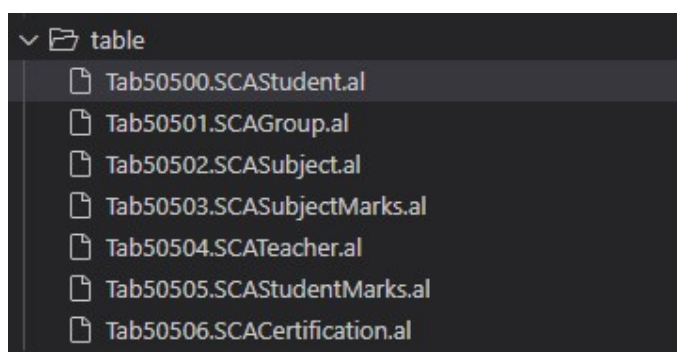


Рисунок 4.1. – Перелік всіх таблиць

Після завантаження структури таблиць на демо базу Microsoft Business Central 365 CRONUS International LTD. на MSSQL сервер в цій базі даних з'являються відповідні створені сутності, що відображені на рисунку 8.

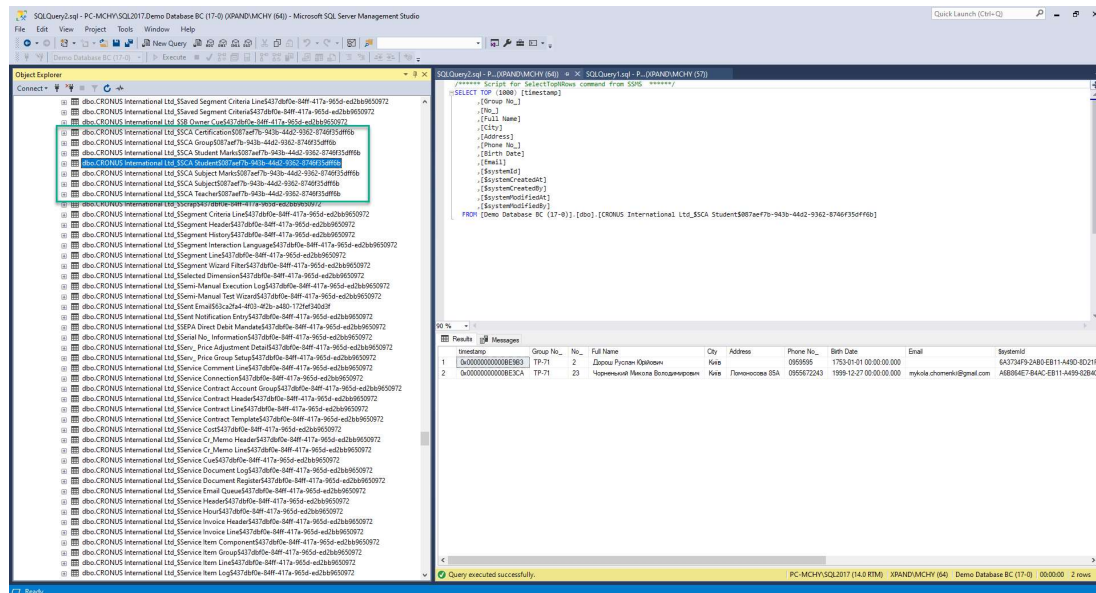


Рисунок 4.2. – Перелік всіх таблиць в MSSQL

В результаті створення було отримано базу даних, що має відношення між основними її сутностями за допомогою функціоналу TableRelations, доступного в Microsoft Business Central. При проектуванні реляційних баз даних прийнято відмінювати наступні типи відносин між таблицями:

- відношення один до багатьох;
- відношення багато до багатьох;
- відношення один до одного.

Відносини один-до багатьох є найбільш поширеними. Якщо ваша модель проектування бази даних показує, що вам необхідно встановити відносини «багато-до-багатьох», то, швидше за все, ваш проект неефективний. Зазвичай стосунки «багато-до-багатьох» можна розбити на два відносини «один-до-багатьох». Відносини «один до одного» зазвичай не є оптимальними, і їх часто можна уникнути, об'єднавши дві таблиці. Модель бази даних зображено на рисунку 9.

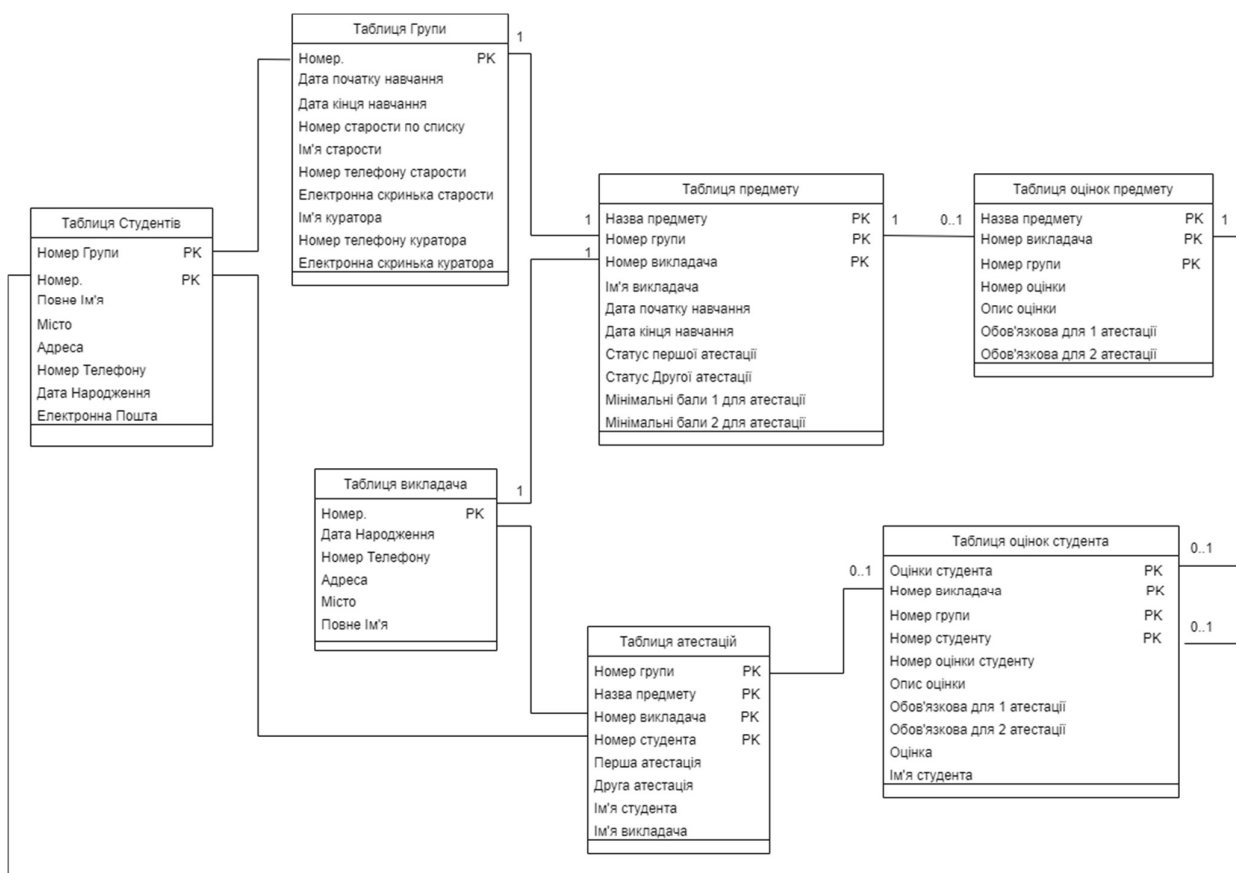


Рисунок 4.3. – Перелік всіх таблиць в MSSQL

Таблиці — це фундаментальні об'єкти будь-якої бази даних. Це об'єкти, в яких ви зберігаєте дані і маніпулюєте ними. Це вірно незалежно від того, якими даними вам потрібно управляти. Коли ви створюєте нову базу даних, ви починаєте з створення таблиць. Пізніше ви створюєте сторінки і звіти для доступу і перегляду даних в таблицях.

Таблицю можна представити у вигляді двовимірної матриці, що складається з стовпців і рядків. На наступному малюнку показана таблиця, де кожен рядок — це запис, а кожен стовпець — поле.

Таблиця складається з двох частин: дані таблиці і опис таблиці. Дані таблиці — це та частина, яку користувачі часто вважають частиною бази даних, оскільки вона містить фактичні записи з їх полями даних. Однак розташування і властивості

цих полів визначаються описом таблиці. Опис таблиці не видно безпосередньо користувачеві.

4.2. Створення логіки системи

У мові програмування AL можливість створення програмістської логіки надана в об'єктах всіх типів, проте загально прийнято використовувати наступний підхід розміщення логіки відносно об'єктів:

- Codeunit — громіздка та ресурсозатратна логіка що може відноситись одразу до декількох окремих підсистем;
- Table — невелика і нересурсозатратна логіка, що має пряме відношення до сутності;
- Page — логіка яка не може бути винесена чи використана за межами цього об'єкту;
- Report — вся логіка пов'язана з цією сутністю окрім коду що може бути використано повторно.

В створеній системі код використовується саме за таким принципом. Розглянемо блок коду що відповідає за автоматичне атестування студентів. Його було винесено в окремий об'єкти типу Codeunit, адже він мстить громіздку та ресурсозатратну логіка що відноситься одразу до декількох окремих підсистем. Процедура RunAutomaticCertification відфільтровує списки атестацій та за результатами дочірніх процедур MinMarkPassed та MandatoryMarksPassed виставляє чи ні атестацію студентам. Блок-схему роботи цього блоку програми зображено на рисунку 4.4.

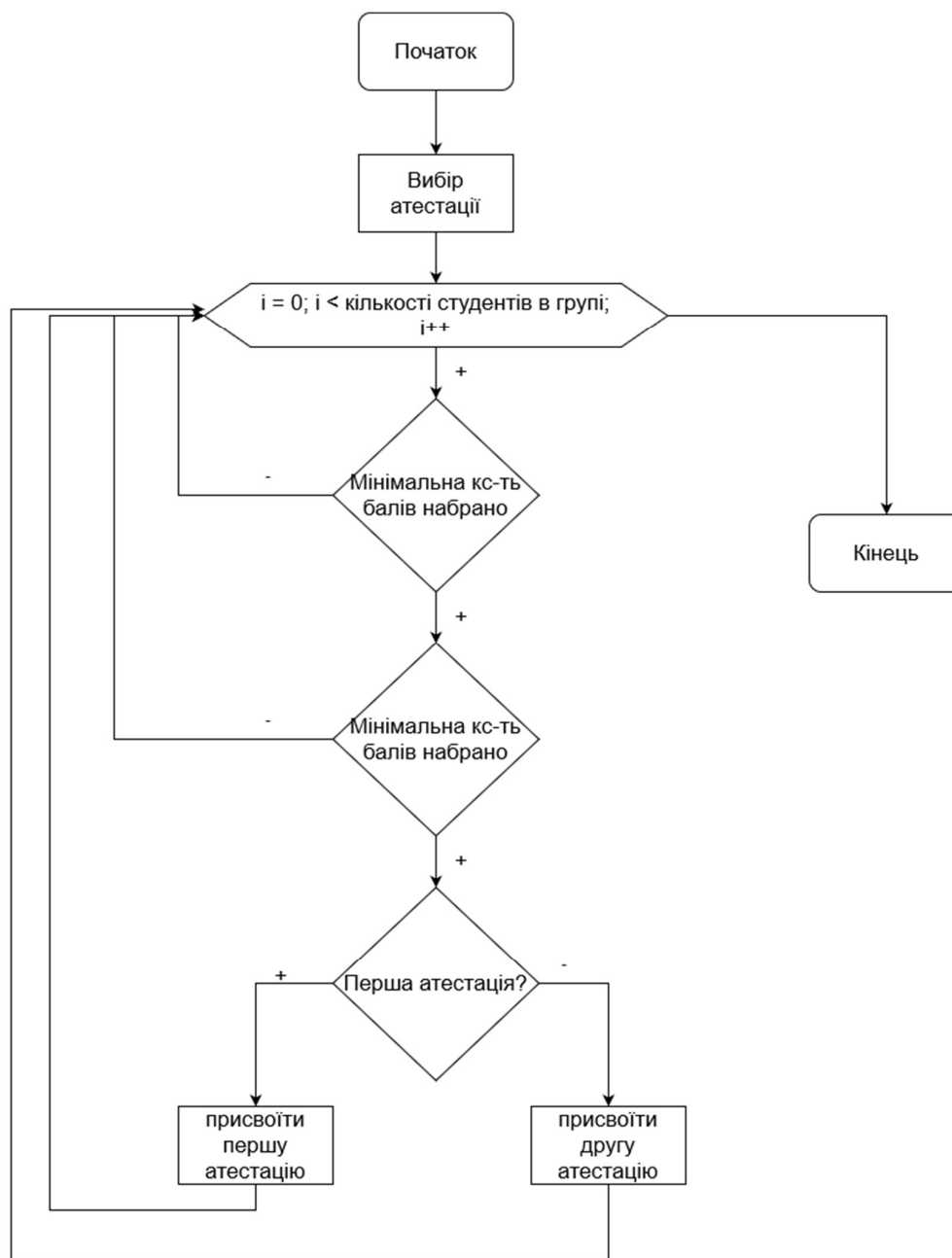


Рисунок 4.4. – Блок схем роботи автоматичних атестацій

4.3. Створення відображення

В мові програмування AL сутності для користувацьких інтерфесів створюються за допомогою об'єктів типу page. Об'єкти типу page мають ряд підтипів, які були використані в системі:

- Card;
- List;
- Document;
- Worksheet.

Об'єкти типу List використовувались для швидкої навігації та вибору потрібного запису з бази даних. Яскравий приклад такого об'єкту це список студентів. На рисунку 4.5 зображено його вигляд в користувацькому інтерфейсі.

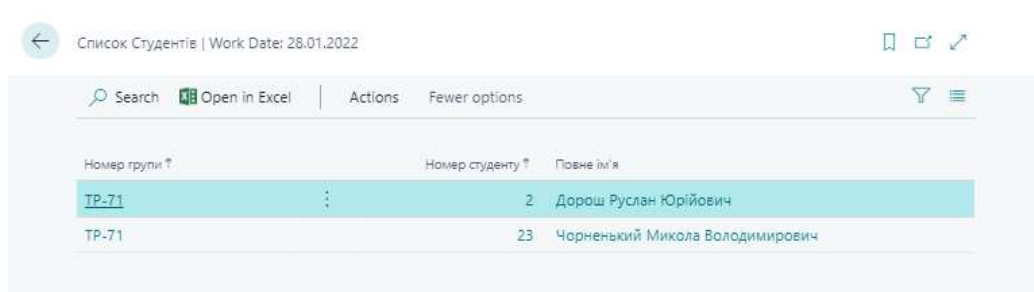


Рисунок 4.5. – Вигляд списку студентів в користувацькому інтерфейсі

Об'єкти типу Card використовувались для розгляду детальної інформації сутності. Яскравий приклад такого об'єкту це картка студенту. На рисунку 4.6 зображено його вигляд в користувацькому інтерфейсі.

TP-71 · 2

Actions

GroupName

Номер групи	TP-71	Адреса	
Номер Студенту	2	Номер телефону	0959595
Повне ім'я	Дорош Руслан Юрійович	Дата Народження	
Місто	Київ	Електронна Пошта	

Рисунок 4.6. – Вигляд картки студента в користувацькому інтерфейсі

Об'єкти типу Document використовувались для розгляду детальної інформації сутності та паралельної роботи з додатковими сутностями. Яскравий приклад такого об'єкту це документ предмету. На рисунку 4.7 зображено його вигляд в користувацькому інтерфейсі.

Edit - SCA Subject Document - Програмування 2 · 1 · TP-71

Manage
Page
Actions
Fewer options

GroupName

Назва Дисципліни
Програмування 2

Дата Початку Виклад...

Номер Викладача
1

Дата Кінця Викладання

Ім'я Викладача
Варава Іван Андрійович

Статус Першої Атестації
Не Почато

Номер Групи
TP-71

Статус Другої Атестації
Не Почато

Список Оцінок
Manage

Опис Оцінки	Обов'язково Для 1 Атестації	Обов'язково Для 2 Атестації
→ Лабораторна робота 1	☒	☐

Close

Рисунок 4.7. – Вигляд документа предмету в користувацькому інтерфейсі

Об'єкти типу Worksheet використовувались для швидкої взаємодії з фільтрами та виборупотрібного запису з бази даних. Яскравий приклад такого об'єкту це сторінка оцінок. На рисунку 4.8 зображено його вигляд в користувацькому інтерфейсі.

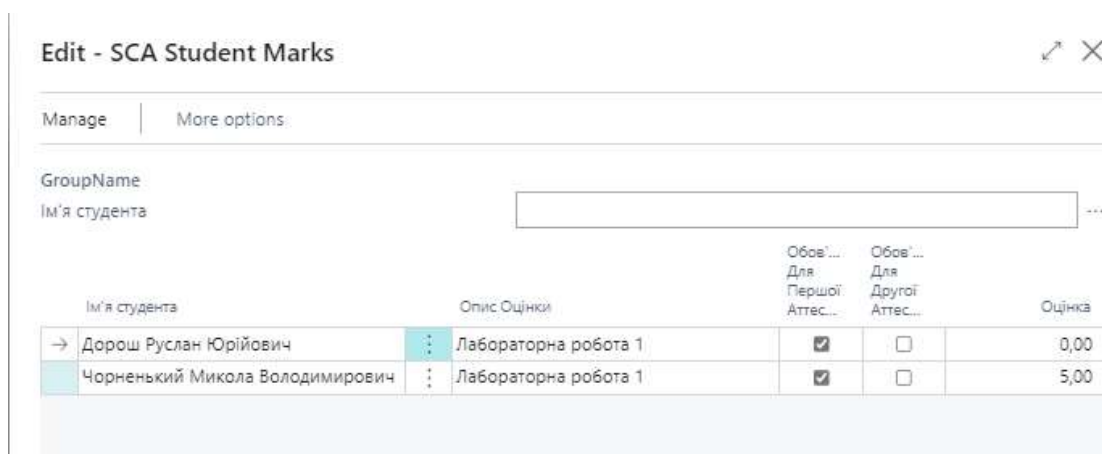


Рисунок 4.8. – Вигляд сторінки оцінок в користувацькому інтерфейсі

4.4. Створення звітів

Мова програмування AL дозволяє інтегрувати в рішення користувацькі звіти. Ця технологія потребує використання об'єкту типу report та інтегрованого в неї RDLC файлу розмітки.

За логікою програми в цьому об'єкті збирається потрібна інформація про атестації студентів з SQL таблиці та формується в RDLC звіт, по макету, шлях до якого вказано в налаштуваннях об'єкту. На рисунку 4.9 зображено RDLC макет.

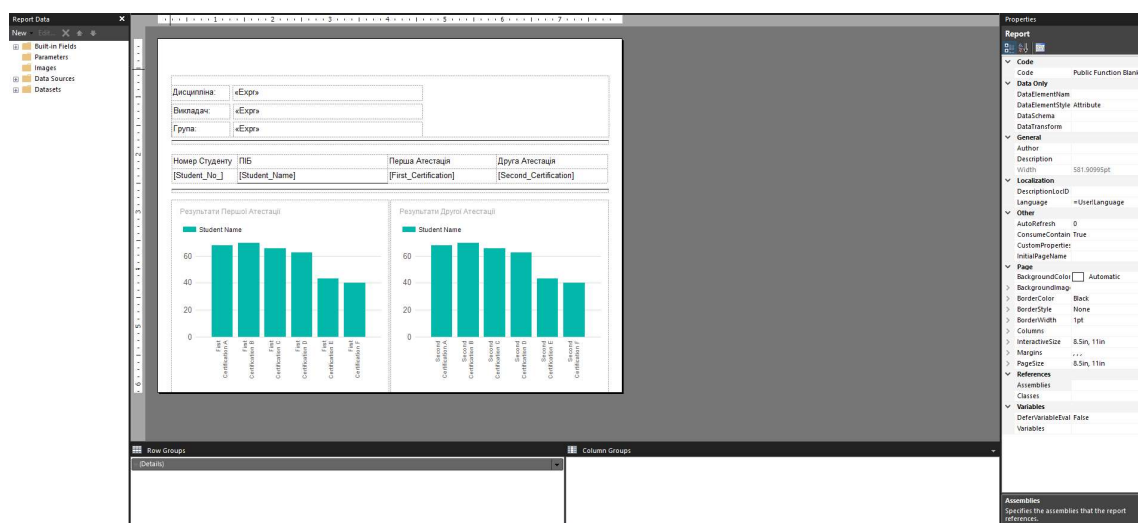


Рисунок 4.9. – RDLC макет для звіту атестацій

Ширина макета була обрана фіксованою по ширині формату А4 для можливості правильного друкування на принтері. В цьому макеті отримані дані формуються в цілісну картину, додаючи ідентифікуючу інформацію про предмет, список студентів та їх атестаційний статус і діаграму зі статистику атестованості.

5. Інтеграція системи атестацій

Систему аналізу атестацій створено на основі Microsoft Dynamics Business Central, через це для використання системи в першу чергу потрібно встановити Microsoft Dynamics Business Central, після чого встановити систему аналізу атестацій завантаживши відповідний app файл. Після встановлення всіх компонентів системи користувачу буде доступний весь функціонал системи.

Перш ніж встановлювати систему аналізу сертифікації, потрібно встановити її платформу, а саме Microsoft Dynamics Business Central. Для цього необхідно завантажити файли інсталяції. Їх можна зайти на офіційному сайті Microsoft Download Center, що зображений на рисунку 5.1.

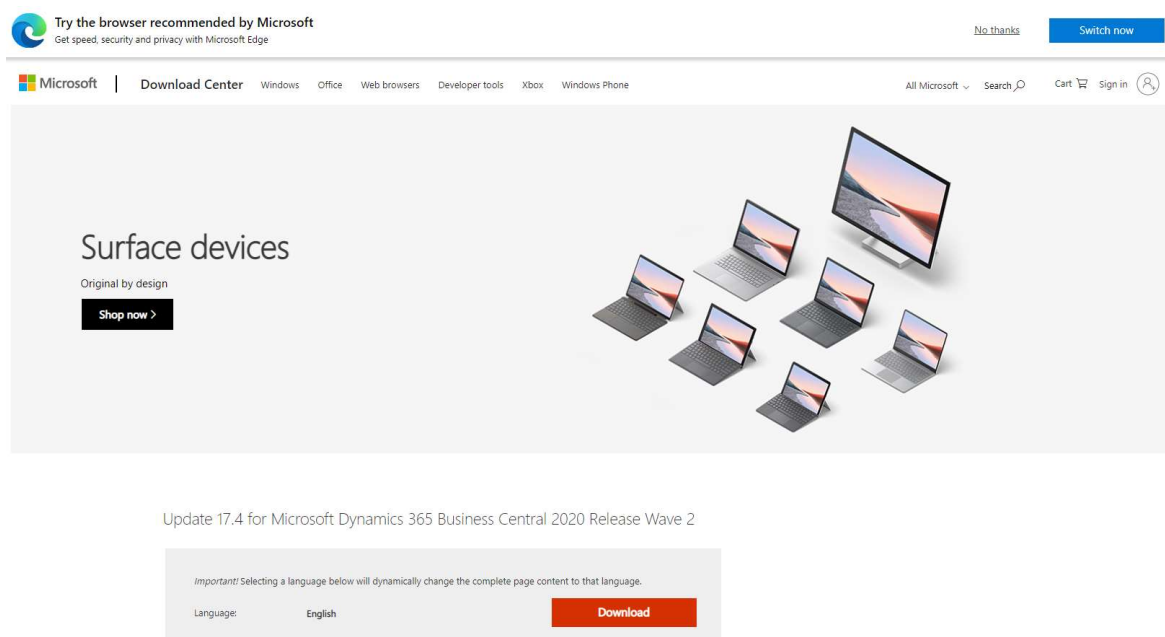


Рисунок 5.1. – Microsoft Download Center

Після завантаження файлів встановлення необхідно запустити файл Setup, інтерфейс якого зображено на рисунку 5.2.

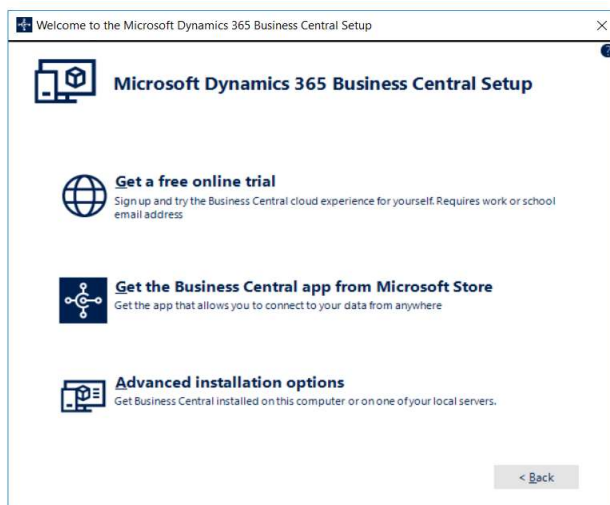


Рисунок 5.2. – Вікно інсталяції Business Central

Важливо розуміти що встановлення Business Central необхідно проводити одразу на сервері. Після коректного встановлення додатку автоматично буде доступний створений веб сайт. Для цілей безпеки може бути використана як вбудована система автентифікації Microsoft Business Central так і система автентифікації самого сервера.

Для використання системи користувачем достатньо встановленого браузеру та використання відповідного посилання для системи.

Для встановлення системи атестацій необібно використати встановлену в процесі інасталяції Microsoft Business Central програму Windows PowerShell. Для цього необхідно:

- Запустити її від ім'я адміністратора системи.
- Імпортувати модуль Microsoft Dynamics Nav Apps Management, використавши команду Import-Module та вказавши шлях до відповідного файлу psd1. Цей файл знаходиться за дерикторією вказаною при встановленні Microsoft Business Central всередині папки Service.
- Завантажити app файл. Для цього необхідно використати команду

Publish-NAVApp, використавши параметри:

- -Path вказавши повну директорію до app файлу
- -ServerInstance вказавши назву інстанса вказаного при інсталяції.
- -SkipVerification без використання параметрів.

Процес завантаження додатку для Business Central зображено на рисунку 5.3.

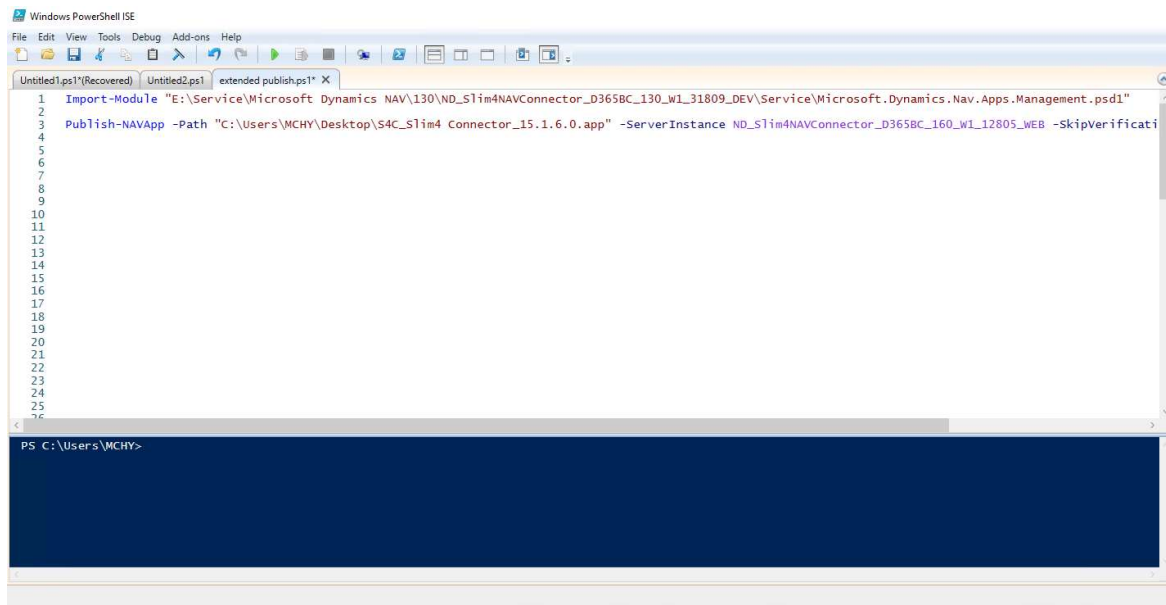


Рисунок 5.3. – Встановлення додатку для Business Central

Після завантаження додатку необхідно його встановити. Для цього потрібно зайти в Business Central, зайти на сторінку

6. Робота користувача із системою

Для коректної роботи системи аналізу атестацій необхідно дотримуватись вимог при його використанні. Так як систему розроблено на основі Microsoft Dynamics Business Central, однією з важливих вимог до користувача є його правильна конфігурація та налаштування.

6.1. Налаштування ролі користувача

Microsoft Dynamics Business Central це доволі велика ERP система, що без додаткової розробки має в собі достатній функціонал для управління основними напрямками бухгалтерського обліку. На рівні з цим він має багато системних налаштувань які дають достатні можливості для системи аналізу атестацій без додаткового втручання в них. Однією з таких системних функцій є налаштування користувачів. Список користувачів зображено на рисунку 6.1

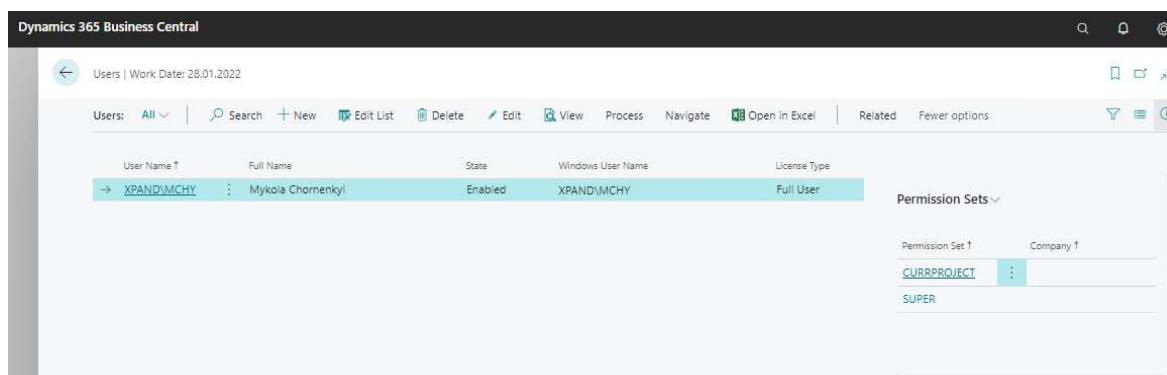


Рисунок 6.1. – Список користувачів

За допомогою цього стандартного функціоналу можна налаштувати авторизацію для системи, та можливості окремих користувачів і груп. Кожен користувач має унікальний ключ GUID та унікальне доменне ім'я, що дає змогу використовувати їх для авторизації в системі. Для кожного користувача можна створити набір дозволених можливостей, присвоїти існуючий чи присвоїти користувача до групи. За допомогою цих можливостей можна регламентувати змогу змінювати існуючі записи в таблицях, редагувати їх, видаляти та додавати

нові записи. Для присвоєння можливостей користувача потрібно створити список можливостей Permissions Set. Таким чином можливості користувацької групи, що зображені на рисунку 6.2., включають в себе тільки перегляд інформації про їх успішність та інформацію про викладачів.

Permissions | Work Date: 28.01.2022

Search + New Edit List Delete Select Permission Set Start Stop SmartList Permissions Open in Excel More options

General

Permission Set: СТУДЕНТ Add Read Permission to Related Tables: ☐

Show: Only In Permission Set Copied from System Permission Set: ☐

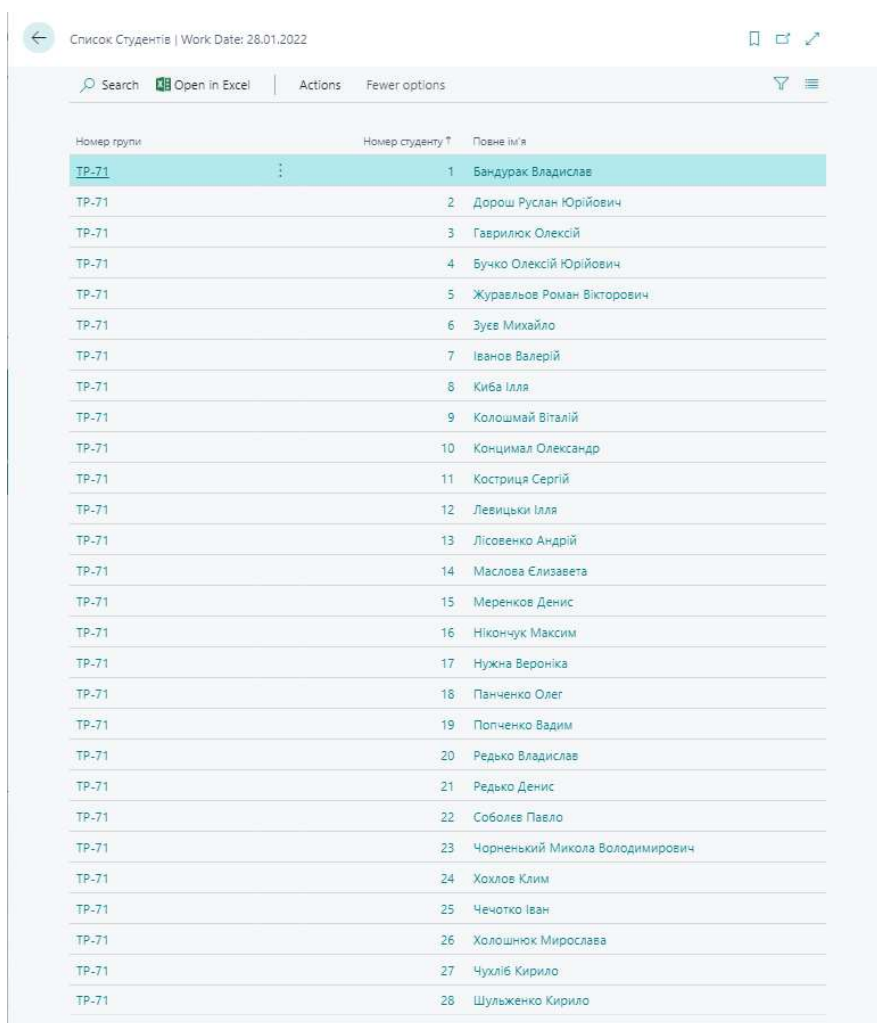
Object Type 1	Object ID 1	Object Name	Read Permission	Insert Permission	Modify Permission	Delete Permission
Table Data	50504	SCA Teacher	Yes	Indirect	Indirect	Indirect
Table Data	50506	SCA Certification	Yes	Indirect	Indirect	Indirect
Table Data	50505	SCA Student Marks	Yes	Indirect	Indirect	Indirect
Table Data	50503	SCA Subject Marks	Yes	Indirect	Indirect	Indirect
→ Table Data	50502	SCA Subject	Yes	Indirect	Indirect	Indirect

Рисунок 6.2. Список можливостей користувацької групи

6.2. Сутності системи та робота з ними

Для коректної роботи системи аналізу атестацій необхідне правильне заповнення таблиць для ведення відомостей успішностей студентів. Першими сутностями рекомендованими для заповнення є сутності студента та викладача.

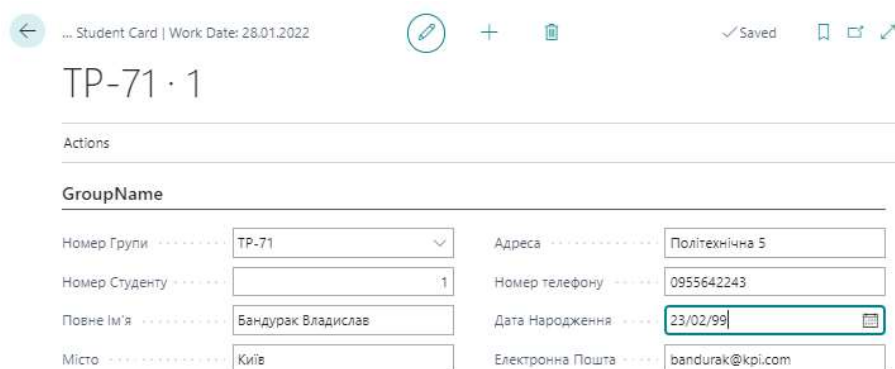
Для заповнення сутності студента необхідно в пошуку Business Central ввести “Список студентів”. Після цього відкриється список зі всіма створеним студентами в системі. Для перегляду, або редагування існуючого студенту потрібно натиснути на запис з його інформацією. Для створення нового необхідно натиснути кнопку new. Список студентів зображено на рисунку 6.3.



Номер групи	Номер студенту	Повне ім'я
TP-71	1	Бандурак Владислав
TP-71	2	Дорош Руслан Юрійович
TP-71	3	Гаврилюк Олексій
TP-71	4	Бучко Олексій Юрійович
TP-71	5	Журавльов Роман Вікторович
TP-71	6	Зуєв Михайло
TP-71	7	Іванов Валерій
TP-71	8	Киба Ілля
TP-71	9	Колошмай Віталій
TP-71	10	Концимал Олександр
TP-71	11	Костиця Сергій
TP-71	12	Левицьки Ілля
TP-71	13	Лісовенко Андрій
TP-71	14	Маслова Єлизавета
TP-71	15	Меренков Денис
TP-71	16	Нікончук Максим
TP-71	17	Нужна Вероніка
TP-71	18	Панченко Олег
TP-71	19	Попченко Вадим
TP-71	20	Редько Владислав
TP-71	21	Редько Денис
TP-71	22	Соболев Павло
TP-71	23	Чорненький Микола Володимирович
TP-71	24	Хохлов Клим
TP-71	25	Чечотко Іван
TP-71	26	Холошнюк Мирослава
TP-71	27	Чухліб Кирило
TP-71	28	Шульженко Кирило

Рисунок 6.3. – Список студентів

Після натискання на запис студента буде відкрито картку студента, що зображена на рисунку 6.4.



... Student Card | Work Date: 28.01.2022

TP-71 · 1

Actions

GroupName

Номер Групи: TP-71

Номер Студенту: 1

Повне Ім'я: Бандурак Владислав

Місто: Київ

Адреса: Політехнічна 5

Номер телефону: 0955642243

Дата Народження: 23/02/99

Електронна Пошта: bandurak@kpi.com

Рисунок 6.4. – Картка студента

На списку студентів винесені основні поля за якими можна ідентифікувати студента. На списку студентів поля не редаговані, проте переходячи на картку студентів вони стають редагованими. Картка студента містить в собі наступні поля, що можуть бути використані як контактна інформація так і для аналізу атестацій та успішності студента:

- номер групи;
- номер студента;
- повне ім'я;
- місто;
- адреса;
- номер телефону;
- електронна адреса.

Наступною необхідною для заповнення сутністю є викладач. Для заповнення сутності викладач необхідно в пошуку Business Central ввести “Список викладачів”. Після цього відкриється список зі всіма створеним викладачами в системі, що зображений на рисунку 6.5.

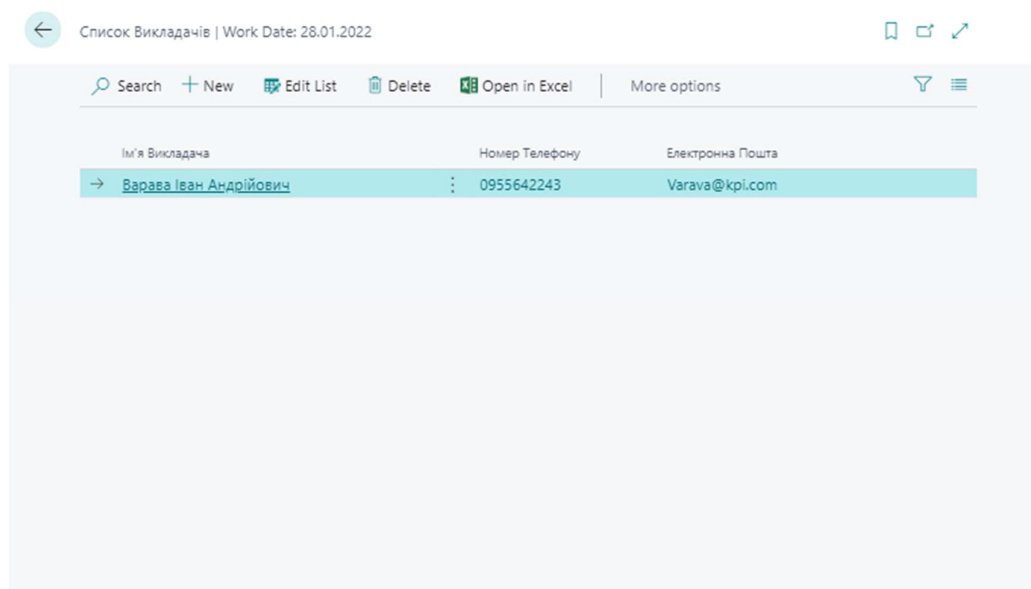


Рисунок 6.5. – Список викладачів

За аналогією з сутністю студентів для перегляду, або редагування існуючого студенту потрібно натиснути на запис з його інформацією. Для створення нового необхідно натиснути кнопку new. Картку викладачів зображено на рисунку 6.6.

Викладач | Work Date: 28.01.2022

1

Actions

GroupName

Номер Викладача	1	Номер Телефону	0955642243
Ім'я Викладача	Варава Іван Андрійович	День Народження	01.01.1990
Місто	Київ	Електронна Пошта	Varava@kpi.com
Адреса	Ломоносова 85а		

Рисунок 6.6. – Картка викладача

Оскільки ціль для сутностей викладачів і студентів дуже схожа, а саме збереження контактної та потрібної для ведення відомостей успішності, — вони мають ідентичний функціонал, та ідентичні поля для заповнення.

Для групування студентів та спрощення керування навчальним процесом наступна необхідна для заповнення сутність це сутність групи. Для її заповнення необхідно перейти в “Список Груп”, що зображена на рисунку 6.7. Для цього необхідно використати стандартний пошук Business Central використавши запит “Список Груп”.

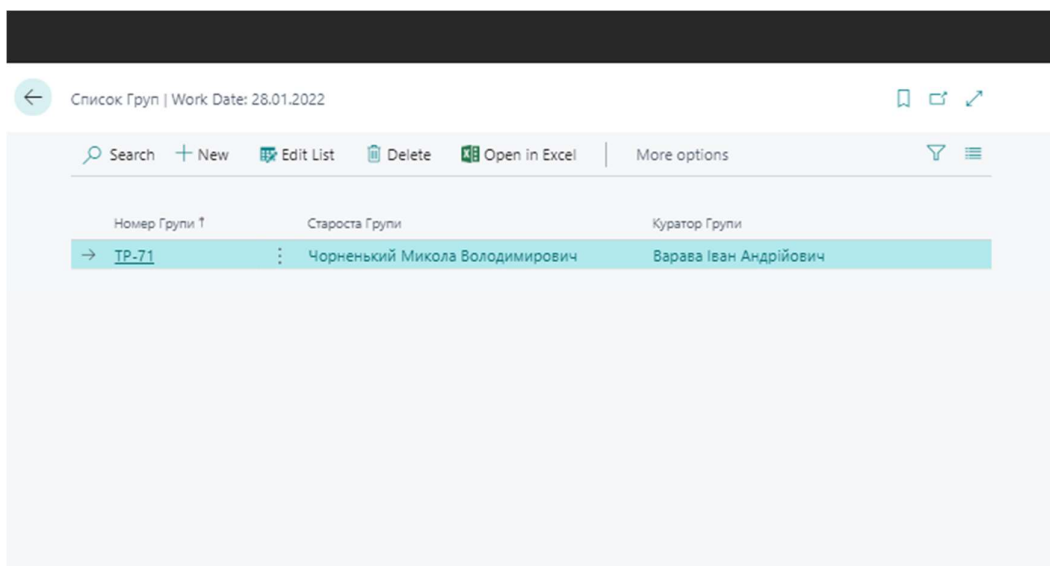


Рисунок 6.7. – Список груп

Для зручності ідентифікації групи на сторінку списку груп виведено номер групи, ім'я старости та куратора. Поля на списку не є радеговними і для створення, перегляду даних групи та зміни існуючої групи необхідно перейти на картку студентів. Картку студентів зображено на рисунку 6.8.

Група | Work Date: 28.01.2022

TP-71

Actions

Основне

Номер Группы: TP-71 Дата Кінця Навчання: [Calendar Icon]

Дата Початку Навча...: [Calendar Icon]

Староста

Номер Старости По ...: 23 Номер Телефону Ста...: 0955672243

Ім'я Старости: Чорненький Микола Володим... Електронна Пошта С...: mykola.chornenki@gmail.com

Куратор

Номер Куратора: 1 Номер Телефону Кур...: 0955642243

Ім'я Куратора: Варава Іван Андрійович Електронна Пошта К...: [Field]

Рисунок 6.8. – Картка групи

Картка групи містить в собі три логічних розділи:

- Основна інформація. В цій групі відображається основна інформація про групу, така як:
 - номер групи;
 - дата початку навчання;
 - дата кінця навчання.
- Староста. В цій групі відображається основна інформація про старосту, така як:
 - номер старости;
 - ім'я старости;
 - номер телефону старости;
 - електронна пошта старости.
- Куратор. В цій групі відображається основна інформація про куратора, така як:
 - номер куратора;
 - ім'я куратора;
 - номер телефону куратора;
 - електронна пошта куратора.

Після заповнення цих сутностей необхідно оновити список студентів, приствоївши їм номер створеної групи. Після оновлення система містить в собі всю необхідну інформацію для ведення відомостей успішності студентів.

Для ведення відомостей студентів необхідно ввести в пошук Business Central запит “Список Предметів”, після чого буде відкритий список предметів, що зображений на рисунку 6.9.

Електронна Назва Предмету ↑	Ім'я Викладача	Номер Групи ↑
Геометрія	Варава Іван Андрійович	TP-71
Програмування	Варава Іван Андрійович	—
Програмування	Варава Іван Андрійович	TP-71
Програмування 2	Варава Іван Андрійович	TP-71

Рисунок 6.9. – Список предметів

На сутності списку предметів виведено назву предмета, ім'я викладача та номер групи. Для перегляду інформації з предмету, її зміни чи створення нового необхідно перейти на вкладку предмета, що зображена на рисунку 6.10.

Edit - SCA Subject Document - Програмування 2 · 1 · TP-71

Manage Page Actions Fewer options

GroupName

Назва Дисципліни: Програмування 2

Номер Викладача: 1

Ім'я Викладача: Варава Іван Андрійович

Номер Групи: TP-71

Дата Початку Виклад...:

Дата Кінця Викладання:

Статус Першої Атестації: Завершено

Статус Другої Атестації: Триває

Список Оцінок Manage

Опис Оцінок	Обов'язково Для 1 Атестації	Обов'язково Для 2 Атестації
→ Лабораторна робота 1	☒	☐
Лабораторна робота 2	☒	☐
Лабораторна робота 3	☐	☒
Лабораторна робота 4	☐	☒

Close

Рисунок 6.10. – Список предметів

Картка предмету містить в собі два логічних розділи:

- Основна інформація. В цій групі відображається основна інформація про предмет, така як:
 - назва дисципліни;
 - номер викладача;
 - ім'я викладача;
 - номер групи;
 - дата початку викладання;
 - дата кінця викладання;
 - статус першої атестації;
 - статус другої атестації;
 - мінімальна кількість балів для першої атестації;
 - мінімальна кількість балів для другої атестації.
- Списки оцінок. В цій групі відображено список всіх можливих оцінок з цього предмету, вона містить поля:
 - опис оцінок;
 - обов'язково для першої атестації;
 - обов'язково для другої атестації.

Використавши функцію групи списку оцінок, можна перейти на список оцінок студентів, натиснувши на кнопку “Список оцінок”, після чого буде відкрита сторінка оцінок студентів, що зображена на рисунку 6.11. При створенні чи зміні оцінки всі зміни автоматично синхронізуються. Використовуючи фільтрацію за іменем студента, можна виставляти бали та вести відомості успішності.

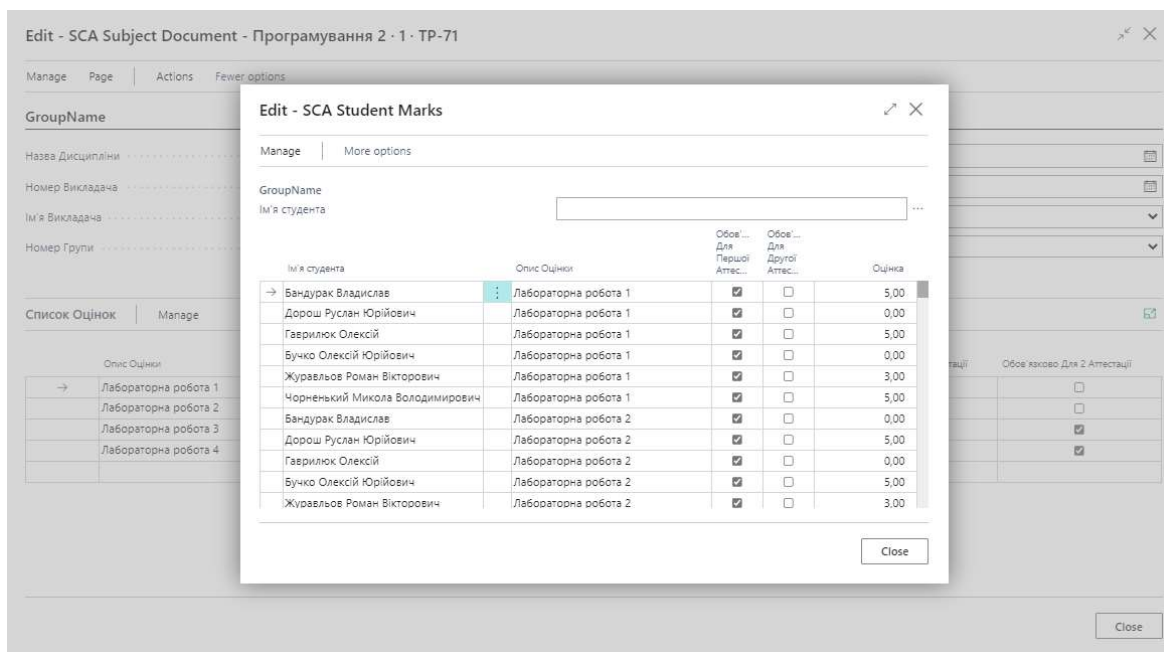


Рисунок 6.11. – Список оцінок студентів

Після заповнення оцінок студентів на головній сторінці предмету можна вставити лист атестації, проставити автоматичні атестації, базовані на отриманих студентами оцінках та відкрити лист отриманих атестацій. Для цього необхідно використати зображені на рисунку 30 кнопки.

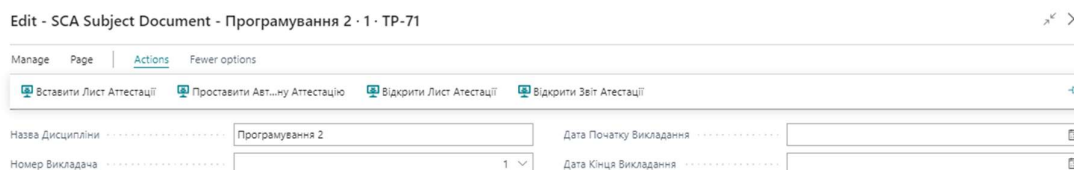


Рисунок 30. – Функції картки предмету

Відкривши список атестацій за допомогою описаного вище функціоналу відкриється список атестацій студентів. Цей список редагований на випадок можливих додаткових змін в ньому (студент виконав необхідні завдання для атестації, проте зробив це невчасно). Описаний список зображено на рисунку 6.12.

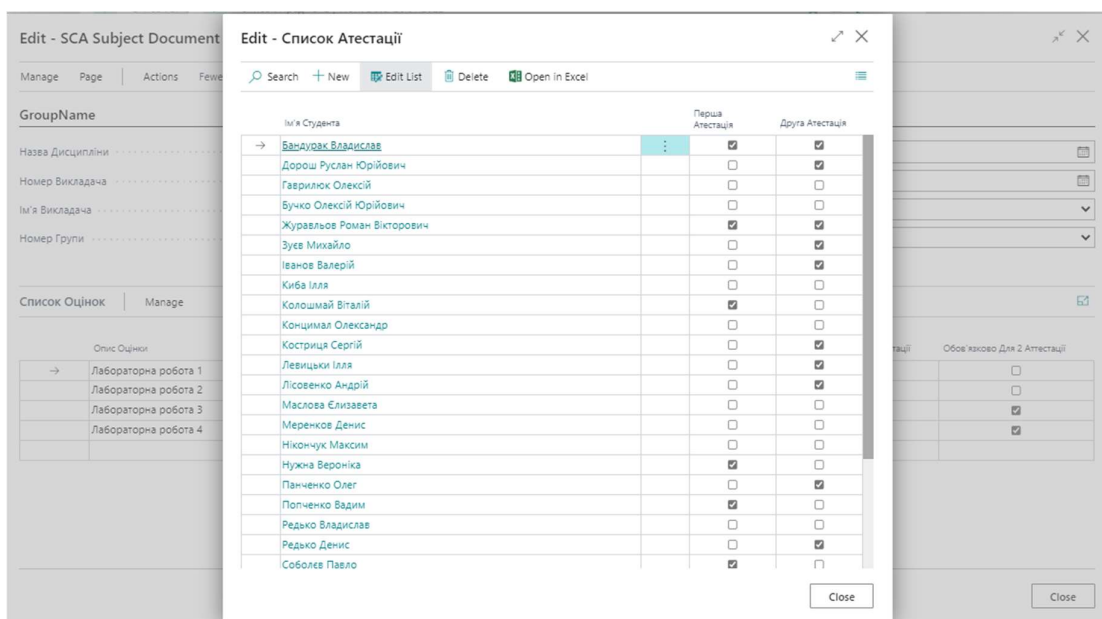


Рисунок 6.12. – Список атестації студентів

Базуючись на цій сутності, буде створений звіт з атестацій студентів. Звіт атестацій створюється після використання функції “Створити звіт атестацій”. Після цього буде згенерований pdf-звіт, та за налаштування браузера буде завантажений чи автоматично відкритий і надрукований.

Звіт містить в собі повну інформацію про статус атестації та може бути умовно розділений на три частини:

- Загальна інформація про предмет, де відображено:
 - назву дисципліни;
 - ім'я викладача;
 - номер групи.
- Інформація про атестації, де відображено:
 - номер студента;
 - ім'я студента;
 - статус першої атестації;
 - статус другої атестації.

- Діаграми успішності атестацій, де зображена кількість атестованих та неатестованих студентів.

Звіт атестацій зображено на рисунку 6.13.

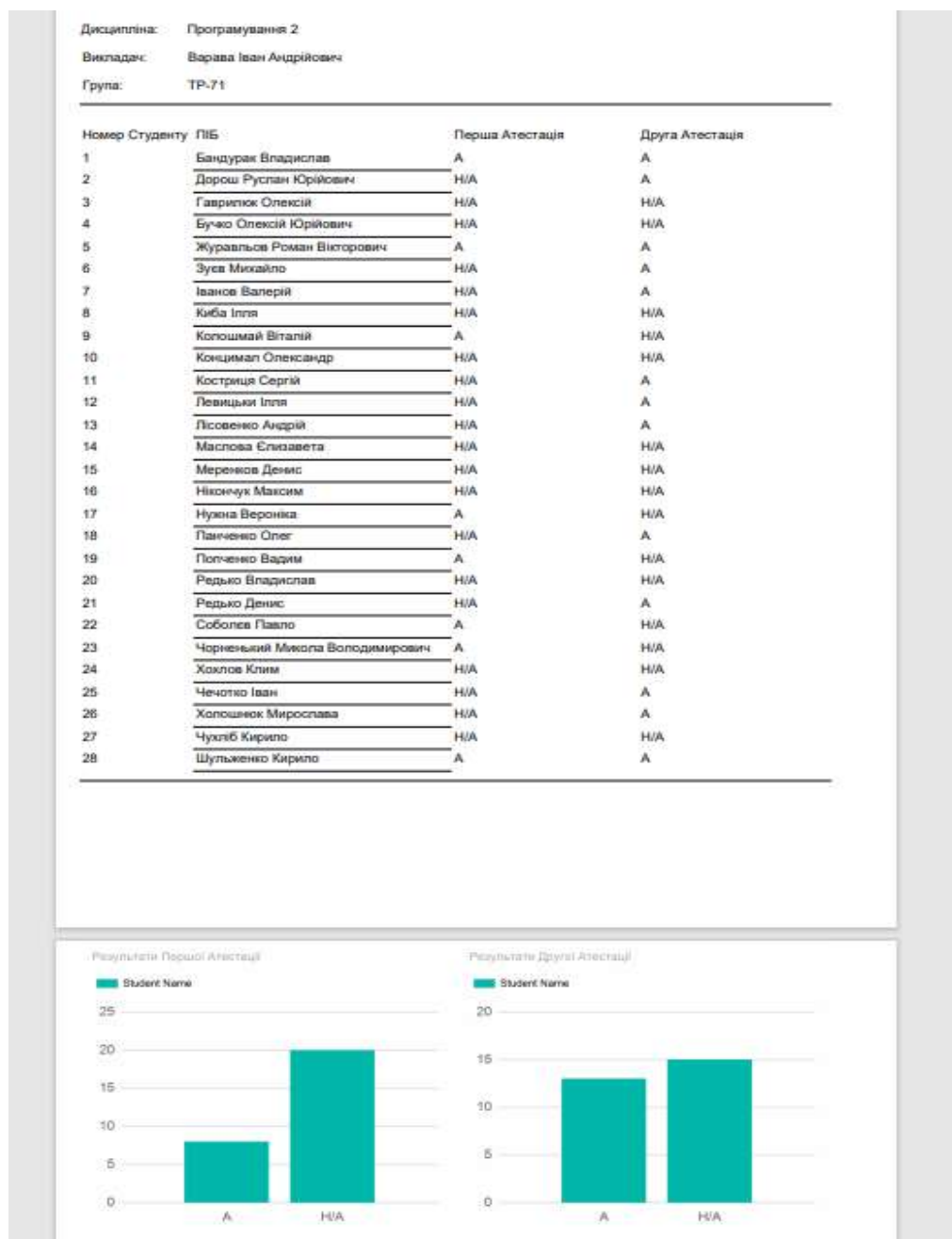


Рисунок 6.13. – Звіт про атестацію студентів

ВИСНОВКИ

В результаті даної роботи була створена система аналізу атестацій студентів, яка спрощує процес упорядкування інформації про успішність студентів та аналізу їх успішності. Це вирішує проблеми поганої обізнаності студентів та інших викладачів з навчальним процесом та атестацією студентів, неправильним заповненням інформації, надмірним витрачанням дорогих паперових носіїв та неможливістю швидкого аналізу.

Для створення системи спочатку було проаналізовано організацію існуючого процесу управління даними. Визначено, які етапи можна систематизувати та автоматизувати.

Модель предметного домену розроблена з урахуванням природи та взаємозв'язку між ними. При проектуванні використовуються різні діаграми, котрі створюють ці схеми, які заохочують знання інформації та допомагають структуровано вирішувати проблеми.

Прикладами таких діаграм є попередні діаграми, які допомагають визначити ролі користувачів у системі, та діаграми класів, які допомагають подальшому проектуванню моделі бази даних.

Після аналізу предметної області були сформовані вимоги для майбутньої системи. Відповідно до цих потреб було обрано засоби розробки програмного забезпечення, такі як мови програмування, бази даних тощо. Для кожного інструменту були враховані переваги та недоліки.

Здійснено програмну реалізацію системи, описав її архітектуру та розглянуто різні модулі системи. Була підтверджена архітектура системи та розглянуті її можливі альтернативи та можливості вдосконалення.

Нарешті, описується, як користувачі застосовують програму. З цією метою детально розглядаються основні сценарії використання системи з акцентом на пришвидшення та вдосконалення ведення та аналізу інформації студентів.

Результатом роботи є працююча система, яка є гнучкою та здатна легко

розширюватись для додавання нового функціоналу. Вона представлена у вигляді прикладного програмного інтерфейсу (API), який є частиною клієнт-серверної архітектури.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Rabindra Sah «Mastering Microsoft Dynamics NAV 2016», «Packt Publishing Ltd.», 2017
2. Глушаков С.В., Ломотько Д.В. Базы данных. Учебный курс // Харьков: Фолио; Ростов н/Д: Феникс; Киев: Абрис, 2000. - 504 с.
3. Корнеев В.В. и др. Базы данных. Интеллектуальная обработка информации М.:Нолидж, 2000. - 352 с
4. Интернет-ресурс : <https://docs.microsoft.com/ru-ru/dynamics365/business-central/>
5. "Світ InterBase. Архітектура, адміністрування та розробка додатків баз даних в InterBase/FireBird/Yaffil» А. Ковязін, С. Востріков (Вид. «Кудиц-Образ») 2002 р.
6. А. В. Потапкін. Основи Visual Basic для пакета Microsoft Office, «Еком», 1995, 256 с.
7. Система управління базами даних Access. Навчальний посібник з курсу «Організація баз даних і баз знань»/ Укладач В.О. Нелюбов. – Ужгород: Редакційно-видавничий відділ ЗакДУ, 2011. – 62 с.
8. Гайдаржи В. І. Основи проектування та використання баз даних : Навч. посібник – 2-ге вид., виправл. і допов./ В. І. Гайдаржи, О. А. Дацюк – К.:ІВЦ “Видавництво “Політехніка”, ТОВ “Фірма “Періодика”, 2004. - 256с.
9. Системи баз даних. Комп’ютерний практикум [Електронний ресурс]: навч. посіб. для здобувачів ступеня бакалавра за освітньою програмою «Комп’ютерний моніторинг та геометричне моделювання процесів і систем» / КПІ ім. Ігоря Сікорського ; уклад.: І.В. Сегеда, О.А. Дацюк. – Електронні текстові дані (1 файл: 987 Кбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 43 с. ; Url: ela.kpi.ua/handle/123456789/27252

ДОДАТОК А

Система аналізу результатів проведення атестації студентів на кафедрі
на основі Microsoft Dynamics 365 Business Central

Специфікація

УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТР-71241_21Б

Аркушів 2

Київ — 2021

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ ТР-71241_21Б	Записка.docx	Текстова частина дипломної роботи
Компоненти		
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ ТР-71241_21Б 12-1	Student_Certification_ Analyzer_1.0.app	Компонент, що містить систему аналізу атестацій

ДОДАТОК Б

Система аналізу результатів проведення атестації студентів на кафедрі
на основі Microsoft Dynamics 365 Business Central

Текст програми

УКР.НТУУ"КПІ" _ТЕФ_АПЕПС_ТР-71241_21Б
12-1

Аркушів 7

Київ — 2021

Student_Certification_Analyzer_1.0.app

```
codeunit 50500 "SCA Certification MGT"
```

```
{
```

```
    trigger OnRun()
```

```
    begin
```

```
end;
```

```
procedure RunAutomaticCertification(Subject: Record "SCA Subject")
```

```
var
```

```
    Selected: Integer;
```

```
    Text000: Label 'Перша Атестація, Друга Атестація';
```

```
    Text002: Label 'Виберіть потрібну атестацію:';
```

```
    StudentMarks: Record "SCA Student Marks";
```

```
    Certification: Record "SCA Certification";
```

```
begin
```

```
    Selected := Dialog.StrMenu(Text000, 3, Text002);
```

```
    Certification.Reset();
```

```
    Certification.SetRange("Subject Name", Subject."Subject Name");
```

```
    Certification.SetRange("Teacher No", Subject."Teacher No");
```

```
    Certification.SetRange("Group No.", Subject."Group No");
```

```
    if Certification.FindSet() then
```

```
        repeat
```

```
            if MinMarkPassed(Subject, Selected, Certification) and MandatoryMarksPassed(Selected, C
ertification) then
```

```
                if Selected = 1 then begin
```

```
                    Certification."First Certification" := true;
```

```
                    Certification.Modify();
```

```
                end else begin
```

```
                    Certification."Second Certification" := true;
```

```
                    Certification.Modify();
```

```
                end;
```

```

    until Certification.Next() = 0;
end;

```

```

procedure MinMarkPassed(Subject: Record "SCA Subject"; Selected: Integer; var Certification: Record "SCA Certification"): Boolean

```

```

var
    StudentMarks: Record "SCA Student Marks";
    Points: Decimal;
begin
    StudentMarks.Reset();
    StudentMarks.SetRange("Subject Name", Certification."Subject Name");
    StudentMarks.SetRange("Teacher No", Certification."Teacher No");
    StudentMarks.SetRange("Group No", Certification."Group No.");
    StudentMarks.SetRange("Student No.", Certification."Student No.");
    if StudentMarks.FindSet() then
        repeat
            Points += StudentMarks.Mark;
        until StudentMarks.Next() = 0;
    if Selected = 1 then
        exit(Points >= Subject."First Cert. Min. Points")
    else
        exit(Points >= Subject."Second Cert. Min. Points");
    end;
end;

```

```

procedure MandatoryMarksPassed(Selected: Integer; var Certification: Record "SCA Certification")
: Boolean

```

```

var
    StudentMarks: Record "SCA Student Marks";
    Points: Decimal;
begin
    StudentMarks.Reset();
    StudentMarks.SetRange("Subject Name", Certification."Subject Name");
    StudentMarks.SetRange("Teacher No", Certification."Teacher No");

```

```

StudentMarks.SetRange("Group No", Certification."Group No.");
StudentMarks.SetRange("Student No.", Certification."Student No.");
if Selected = 1 then begin
    StudentMarks.SetRange("Mandatory For First Cert.", true);
    StudentMarks.SetRange(Mark, 0);
    if StudentMarks.IsEmpty then
        exit(true);
end else begin
    StudentMarks.SetRange("Mandatory For Second Cert.", true);
    StudentMarks.SetRange(Mark, 0);
    if StudentMarks.IsEmpty then
        exit(true);
end;
end;

var
    myInt: Integer;
}

report 50500 "SCA Certifications"
{
    DefaultLayout = RDLC;
    RDLCLayout = 'SCA Certifications.rdl';
    Caption = 'Звіт Атестацій';
    PreviewMode = PrintLayout;

    dataset
    {
        dataitem(SCACertification; "SCA Certification")
        {
            column(Student_No_; "Student No.")
            {
            }
        }
    }
}

```

```

column(Student_Name; "Student Name")
{
}
column(Subject_Name; "Subject Name")
{
}
column(First_Certification; FirstCertStatus)
{
}
column(Second_Certification; SecondCertStatus)
{
}
column(Group_No_; "Group No.")
{
}
column(Teacher_Name; "Teacher Name")
{
}
trigger OnAfterGetRecord()
var
    Subject: Record "SCA Subject";
begin
    if "First Certification" then
        FirstCertStatus := 'A'
    else
        FirstCertStatus := 'H/A';

    if "Second Certification" then
        SecondCertStatus := 'A'
    else
        SecondCertStatus := 'H/A';

    Subject.Get(SCACertification."Subject Name", SCACertification."Teacher No", SCACertifi

```

```

cation."Group No.");
    end;
}
}
var
    FirstCertStatus: Text[20];
    SecondCertStatus: Text[20];
}

```

page 50508 "SCA Subject Subpage"

```

{
    PageType = ListPart;
    ApplicationArea = All;
    UsageCategory = Administration;
    SourceTable = "SCA Subject Marks";

    layout
    {
        area(Content)
        {
            repeater(GroupName)
            {
                field(Description; Rec.Description)
                {
                    Caption = 'Опис Оцінки';
                    ApplicationArea = All;
                }
                field("Teacher Name"; Rec."Mandatory For First Cert.")
                {
                    Caption = 'Обов'язково Для 1 Аттестації';
                    ApplicationArea = All;
                }
            }
        }
    }
}

```

```

field("Group No"; Rec."Mandatory For Second Cert.")
{
    Caption = 'Обов'язково Для 2 Аттестації';
    ApplicationArea = All;
}
}
}
}

actions
{
    area(Processing)
    {
        action(StudentMarks)
        {
            Caption = 'Оцінки';
            ApplicationArea = All;

            trigger OnAction()
            var
                SCASStudentMarks: Record "SCA Student Marks";
            begin
                SCASStudentMarks.Reset();
                SCASStudentMarks.SetRange("Group No", Rec."Group No");
                SCASStudentMarks.SetRange("Subject Name", Rec."Subject Name");
                SCASStudentMarks.SetRange("Teacher No", Rec."Teacher No");
                Page.RunModal(0, SCASStudentMarks);
            end;
        }
    }
}

var

```

```
    myInt: Integer;  
}
```

ДОДАТОК В

Система аналізу результатів проведення атестації студентів на кафедрі
на основі Microsoft Dynamics 365 Business Central

Опис програми

УКР.НТУУ"КПІ" _ТЕФ_АПЕПС_ТР-71241_21Б
13-1

Аркушів 10

Київ — 2021

АНОТАЦІЯ

Додаток містить опис застосунку для аналізу атестацій студентів на кафедрі, та виконує супутні завдання визначенні в розділі 1, а саме:

- Збереження даних студентів.
- Збереження даних викладачів.
- Збереження даних груп.
- Ведення відомостей успішності студентів.
- Збереження даних про предмет.
- Виставлення атестацій.
- Виведення даних про успішність студентів.

Застосунок розроблено мовою програмування AL з використанням технології RDLC у редакторі вихідного коду Visual Studio Code.

ЗМІСТ

1. ЗАГАЛЬНІ ВІДОМОСТІ	66
2. ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ.....	67
3. ОПИС ЛОГІЧНОЇ СТРУКТУРИ	68
4. ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ.....	69
5. ВИКЛИК І ЗАВАНТАЖЕННЯ.....	70

ЗАГАЛЬНІ ВІДОМОСТІ

У додатку міститься опис основних компонентів застосунку для аналізу атестацій студентів на кафедрі. Додаток Б містить програмний код цих компонентів.

Для роботи застосунку потрібно встановити Microsoft Dynamics Business Central. Також необхідно мати підключення до Інтернету.

Застосунок розроблено мовою програмування AL з використанням технології RDLC у редакторі вихідного коду Visual Studio Code.

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Розроблений застосунок надає користувачу наступний функціонал:

- Реєстрація/авторизація в системі.
- Збереження даних студентів.
- Збереження даних викладачів.
- Збереження даних груп.
- Ведення відомостей успішності студентів.
- Збереження даних про предмет.
- Виставлення атестацій.
- Виведення даних про успішність студентів.
- Зміна налаштувань застосунку.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Згідно з архітектурою, застосунок складається з декількох частин, основні з яких: дані студенту, дані групи, дані викладача, дані предмету.

Аналіз атестації містить в собі наступні складові: дані викладача, дані студента, дані успішності, звіт атестації.

Дані студентів, викладачів та груп містять в собі відповідну інформацію та відображення в інтерфейсі застосунку.

Дані предмету- клас що містить в собі основні данні про предмет та з ключом та стартовою точкою для заповнення відомостей студентів, на основі яких відбувається аналіз атестацій.

Аналіз атестацій відбувається на основі атестаційних відомостей що складаються на основі отриманих студентом оцінок. В автоматичному або мануальному режимі. Після агрегації даних користувач має отримати звіт про успішність студентів.

ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ

Для роботи застосунку потрібно мати пристрій що підтримує роботу додатку Microsoft Dynamics Business Central 180. Також необхідно мати підключення до Інтернету під час роботи в застосунку.

ВИКЛИК І ЗАВАНТАЖЕННЯ

Для встановлення мобільного застосунку на необхідно завантажити Microsoft Business Central з офіційного сайту Microsoft, після чого завантажити додаток на сервер, використавши PowerShell.