

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

Кваснюка Дениса Івановича

Прізвище, ім'я, по батькові студента(ки)

на здобуття ступеня вищої освіти Бакалавра

Система планування ресурсів

Назва теми

навчального закладу на базі Microsoft Dynamics 365

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного
забезпечення»

Шифр КвРІПЗ.2101075.01.06.ПЗ

Виконав студент IV курсу, група ПЗ-21-1


Підпис

Денис КВАСНЮК

Ім'я, ПРІЗВИЩЕ

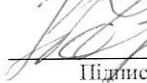
Керівник канд. тех. наук, доцент
Науковий ступінь, звання


Підпис

Оксана ЯШИНА

Ім'я, ПРІЗВИЩЕ

Нормоконтролер канд. тех. наук, доцент


Підпис

Юрій ФОРКУН

Ім'я, ПРІЗВИЩЕ

До захисту допускаю:
Завідувач кафедри інженерії
програмного забезпечення


Підпис

Леонід БЕДРАТЮК

Ім'я, ПРІЗВИЩЕ

5 червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Інженерії програмного забезпечення

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Л. П. Бедратюк

02. 01. 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Кваснюку Денису Івановичу

Прізвище, ім'я, по батькові студента

1. Тема кваліфікаційної роботи Система планування ресурсів навчального закладу на базі Microsoft Dynamics 365

Керівник кваліфікаційної роботи Яшина Оксана Миколаївна, канд. техн. наук, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом роботи на кафедру 01.06.2025 р.

3. Вихідні дані до роботи Матеріали переддипломної практики

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження предметної області та постановка задачі, проектування програмної системи, програмна реалізація та тестування системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Презентаційні матеріали (слайди, 16 шт.), діаграма IDEF-0, діаграма декомпозиції, схема багаторівневої архітектури, схема архітектури Single-Tenant, діаграма розгортання, ER-діаграма, діаграма програмних модулів, діаграма послідовностей для процесу створення нової кафедри, діаграма послідовностей для процесу створення нового факультету, діаграма послідовностей для процесу створення нового студента, діаграма послідовностей для процесу створення нової групи, діаграма послідовностей для створення нового викладача.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Форкун Ю. В. канд. техн. наук, доцент	12.05.2025	28.05.2025
Антиплагіат	Форкун Ю. В. канд. техн. наук, доцент	12.05.2025	28.05.2025

7. Дата видачі завдання «02» 01 2025р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Ознайомлення з тематикою дипломного проектування, визначення та узгодження індивідуальних тем кваліфікаційних робіт (КВР)	01.12 – 31.12.2024	
2 Збір матеріалу за темою КВР; дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання	01.01 – 20.02.2025	
3 Проектування програмного забезпечення	21.02 – 20.03.2025	
4 Програмна реалізація з використанням відповідних засобів розробки. Тестування ПЗ	21.03 – 30.04.2025	
5 Написання вступу, загальних висновків, оформлення переліку джерел посилання та додатків. Оформлення пояснювальної записки КВР згідно вимог	01.05 – 25.05.2025	
6 Попередній захист КВР	Травень 2025	Згідно графіка
7 Перевірка КВР на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошурування (зшиття) пояснювальної записки.	26.05 – 30.05.2025	
8 Здача КВР на кафедру; підготовка КВР для розміщення у репозитарії ХНУ; підготовка до захисту та захист КВР	з 01.06.2025	

Студент

Керівник роботи


Підпис


Підпис

Денис КВАСНЮК

Ім'я, прізвище

Оксана ЯШИНА

Ім'я, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система планування ресурсів навчального закладу на базі Microsoft Dynamics 365».

Автор роботи: Кваснюк Денис Іванович.

Керівник роботи: Яшина Оксана Миколаївна

Пояснювальна записка: 80 с., 20 рис., 11 табл., 2 дод., 40 джерел.

Графічна частина: 16 слайдів.

СИСТЕМА ПЛАНУВАННЯ РЕСУРСІВ, ERP, BUSINESS CENTRAL, MICROSOFT DYNAMICS 365.

Мета роботи полягає у розробці та впровадженні інтегрованої системи планування ресурсів навчального закладу на базі Microsoft Dynamics 365. Система забезпечить централізоване управління матеріальними, людськими та фінансовими ресурсами, підвищить прозорість адміністративних процесів та оптимізує розподіл ресурсів відповідно до стратегічних цілей закладу.

У кваліфікаційній роботі було проведено глибокий аналіз предметної області, визначені можливості схожих цифрових рішень, визначено чіткі вимоги до функціоналу та продуктивності системи, спроектовано її структурну схему та реалізовано основні програмні компоненти із застосуванням актуальних методів тестування програмного забезпечення.

В основу реалізації покладено можливості Microsoft Dynamics 365, що забезпечують високу гнучкість та інтеграцію з іншими бізнес-додатками. Для забезпечення ефективної взаємодії між модулями системи використано передові підходи до інтеграції даних.

Ця система може бути корисною для керівництва та адміністративного персоналу освітніх установ, які прагнуть модернізувати та централізувати управління всіма аспектами своєї діяльності.

04.06.2025
Дата


Підпис

ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КвРІПЗ.2101075.01.06.ПЗ	Пояснювальна записка	80		
2	A4		Завдання на кваліфікаційну роботу	1		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
4	A4	КвРІПЗ.2101075.01.06.E8	Логічна модель бази	1		
5	A4	КвРІПЗ.2101075.01.06.E8	Контекстна діаграма	1		
6	A4	КвРІПЗ.2101075.01.06.E8	Діаграма декомпозиції	1		
7	A4	КвРІПЗ.2101075.01.06.E8	Презентаційні матеріали	16		

					КвРІПЗ.2101075.01.06.ВД			
Змн.	Арк.	№ докум.	Підпис	Дата	Система планування ресурсів навчального закладу на базі Microsoft Dynamics 365	Літ.	Арк.	Аркушів
Виконав		Кваснюк Д.І.		09.06			1	1
Керівник		Яшина О.М.		09.06				
Н. Контр.		Форкун Ю.В.		09.06				
Зав. Каф.		Бедратюк Л.П.		09.06	Відомість документів	ХНУ, ІПЗ-21-1		

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	7
ВСТУП.....	8
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	10
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей	10
1.2 Аналіз наявного програмно-технічного забезпечення предметної області	15
1.3 Визначення функціональних та нефункціональних вимог до програмного забезпечення.....	22
1.4 Висновки дослідження предметної області та постановки задачі.	25
2 ПРОЄКТУВАННЯ СИСТЕМИ ПЛАНУВАННЯ РЕСУРСІВ	26
2.1 Проєктування архітектури та структури системи.....	26
2.2 Проєктування логічної моделі бази даних	34
2.3 Детальне проєктування програмних модулів	43
2.4 Висновки проєктування системи планування ресурсів.....	50
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ ПЛАНУВАННЯ РЕСУРСІВ	51
3.1 Програмна реалізація модулів.....	51
3.2 Розгортання системи планування ресурсів	63
3.3 Вимоги до технічних та програмних засобів.....	66
3.4 Тестування системи планування ресурсів	67
3.5 Висновки програмної реалізації та тестування системи планування ресурсів	73
ВИСНОВКИ	74
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	77
ДОДАТОК А Код програми	81
ДОДАТОК Б Презентаційні слайди	122

					КвРІПЗ.2101075.01.06.ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Система планування ресурсів навчального закладу на базі Microsoft Dynamics 365	Лім.	Арк.	Аркушів
Виконав		Кваснюк Д.І.		04.06			5	80
Керівник		Яшина О.М.		04.06				
Н. Контр.		Форкун Ю.В.		04.06				
Зав. Каф.		Бедратюк Л.П.		05.06	Пояснювальна записка	ХНУ, ІПЗ-21-1		

ПЕРЕЛІК СКОРОЧЕНЬ

БД	–	База даних
ПЗ	–	Програмне забезпечення
ERP	–	Enterprise Resource Planning
IDEF	–	Integrated DEFinition
BI	–	Business Intelligence
ECTS	–	European Credit Transfer and Accumulation System
SaaS	–	Software as a Service
PDF	–	Portable Document Format
SQL	–	Structured Query Language
ER	–	Entity-Relationship
UML	–	Unified Modeling Language
AL	–	Application Language
GDPR	–	General Data Protection Regulation
URL	–	Uniform Resource Locator
CDX	–	Customer Digital Experiences
UI	–	User Interface
AAA	–	Arrange-Act-Assert
CRUD	–	Create, Read, Update, Delete

ВСТУП

У сучасних реаліях ефективного планування ресурсів у світі цифрових технологій стає одним із визначальних факторів успішної роботи як комерційних, так і некомерційних організацій. В умовах постійного зростання обсягів даних підвищується і складність основних бізнес-процесів, які потребують постійної оптимізації. Саме цей невідворотний процес і сприяє активному впровадженню автоматизованих систем планування ресурсів або ERP-систем. Спеціалізовані системи дозволяють автоматизувати цілу низку рутинних процесів та централізовано керувати основними складовими діяльності організації, до яких можна віднести: кадровий персонал, фінанси, звітність, комунікацію та інші.

Попри впровадження новітніх технологій, більшість навчальних закладів досі стикається з проблемами в делегуванні управлінських процесів, браком автоматизованих рішень для обробки великих обсягів даних та неефективним використанням ресурсів. Низький рівень автоматизації призводить до втрати часу, зайвих витрат і ускладнює прийняття управлінських рішень. Це зумовлює потребу у створенні комплексної системи, здатної усунути ці недоліки та вдосконалити наявні підходи.

Таким чином, порушена проблема, на вирішення якої націлена дана кваліфікаційна робота, включає в себе розробку уніфікованого рішення спрямованого на оптимізацію управління основними фінансовими, кадровими та навчальними процесами, застосуванню сучасних підходів для обробки великих об'ємів даних та забезпечення інтеграції розподілених підсистем в одне єдине управлінське середовище.

Актуальність теми обумовлена гострою необхідністю впровадження комплексних цифрових рішень для оптимізації адміністративних процесів у сучасних освітніх установах. Сучасні навчальні заклади стикаються з викликами ефективного розподілу різноманітних ресурсів, що безпосередньо впливає на якість освітніх послуг та фінансову стабільність закладу.

					КВРППЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		8

Мета роботи полягає у розробці та впровадженні інтегрованої системи планування ресурсів навчального закладу на базі Microsoft Dynamics 365. Система забезпечить централізоване управління матеріальними, людськими та фінансовими ресурсами, підвищить прозорість адміністративних процесів та оптимізує розподіл ресурсів відповідно до стратегічних цілей закладу.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести детальний аналіз існуючих систем планування ресурсів в освітніх закладах та оцінити функціональні можливості Microsoft Dynamics 365 для задоволення специфічних потреб освітньої галузі;
- визначити ключові вимоги до системи на основі аналізу бізнес-процесів навчального закладу;
- розробити архітектурне рішення для програмної системи з урахуванням модульності та масштабованості;
- здійснити налаштування та розробку у середовищі Microsoft Dynamics 365 для створення специфічних функціональних модулів;
- забезпечити інтеграцію з існуючими інформаційними системами навчального закладу;
- провести всебічне тестування системи на відповідність функціональним та нефункціональним вимогам.

Запропонована система призначена для адміністративного персоналу навчальних закладів, викладачів та керівництва, яким необхідно ефективно управляти ресурсами закладу.

Вона дозволить автоматизувати рутинні процеси, забезпечить доступ до актуальної аналітичної інформації, сприятиме прийняттю обґрунтованих управлінських рішень та підвищить загальну ефективність використання ресурсної бази навчального закладу.

Таким чином, дана робота спрямована на створення комплексного програмного рішення на базі Microsoft Dynamics 365, яке відповідатиме сучасним вимогам цифрової трансформації освітньої галузі, оптимізує планування ресурсів та підвищить якість роботи.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						9
Зм.	Арк	№ докум.	Підпис	Дата		

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

В умовах динамічного розвитку сучасних бізнес-середовищ та невинного зростання конкуренції у різних галузях, раціональне планування ресурсів стає ключовим чинником успіху будь-якого підприємства. Досліджувана предметна область охоплює комплекс бізнес-процесів, що забезпечують стабільну діяльність численних корпорацій. У межах детального аналізу предметної області, першочергово варто розглянути її ключові аспекти, термінологію, невирішені питання та існуючі проблеми.

Предметна область характеризується циклічною взаємодією та функціонуванням ресурсів підприємств. Аналізуючи структурні елементи, важливо наголосити не лише на ресурсі як на окремій одиниці, але й на його ролі у функціонуванні всієї системи. Адже основоположна робота систем планування ресурсів ґрунтується на управлінні активами та їх безпосередньому залученні до функціональних процесів підприємства в цілому.

Зрештою, у цьому контексті необхідно звернути увагу на функціональні особливості предметної області. До них належать автоматизовані процеси функціонування бізнесу, що виступає об'єктом автоматизації. При більш глибокому аналізі було визначено основні напрямки, що потребують автоматизації, серед яких варто виокремити: генерацію звітності, управління запасами та, відповідно, планування всіх ресурсів.

Попередній аналіз ключових характеристик предметної області системи, що розробляється, на основі виявлених аспектів автоматизації, також дав змогу ідентифікувати критичні проблеми та невирішені питання, які розглядаються з позиції впровадження інформаційних технологій.

Для точного аналізу в даному випадку доцільно звзвити широку предметну область до певної бізнес-сфери.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						10
Зм.	Арк	№ докум.	Підпис	Дата		

Саме тому ключовим об'єктом автоматизації обрано навчальний заклад як організацію, що володіє певним комплексом ресурсів.

В ERP (Enterprise Resource Planning) системі [1], яка призначена для навчального закладу передбачено планування різноманітних видів ресурсів, які забезпечують ефективність освітнього процесу та адміністративної діяльності. Основними категоріями таких ресурсів є матеріальні, людські, фінансові та відповідно інформаційні.

Матеріальні ресурси охоплюють серверне обладнання для розміщення ERP-системи, комп'ютери та робочі місця для адміністрації, викладачів та інших співробітників, а також мережеву інфраструктуру для стабільного доступу до системи. Сюди ж належать навчальні аудиторії, лабораторії, бібліотеки, гуртожитки та інші об'єкти закладу, планування яких може здійснюватися через відповідну ERP-систему.

Людські ресурси включають усіх учасників освітнього процесу: студентів, викладачів, адміністративний персонал та технічних працівників. Система дозволяє ефективно керувати кадровими питаннями, зокрема розподілом навантаження викладачів, обліком навчальних груп, складанням розкладу та веденням електронних журналів.

Фінансові ресурси передбачають управління бюджетом навчального закладу, планування та облік фінансових операцій, нарахування заробітної плати, контроль стипендіальних виплат, формування фінансової звітності та моніторинг витрат на матеріально-технічне забезпечення.

Інформаційні ресурси охоплюють усі дані, що циркулюють у навчальному закладі: особисті дані студентів та викладачів, навчальні плани, розклади, електронні журнали, бібліотечні каталоги, методичні матеріали, а також аналітичні звіти та статистику успішності.

Підсумовуючи аналіз ресурсів, якими може володіти навчальний заклад у контексті своєї діяльності, виявлено, що потреба в автоматизованому вирішенні поставленого завдання зумовлена необхідністю спрощення обліку та управління активами, а також забезпечення зручного доступу до даних, що генеруються в

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						11
Зм.	Арк	№ докум.	Підпис	Дата		

процесі функціонування навчального закладу як організації. Для вирішення поставленої задачі визначено такі завдання:

- розробити концептуальну модель бази даних, що відобразить структуру обліку ресурсів навчального закладу, враховуючи їхні типи, характеристики, місцезнаходження, відповідальних осіб, історію їхнього використання та загальну інформацію про них;

- спроектувати логічну схему бази даних, визначивши таблиці, атрибути, первинні та зовнішні ключі, а також зв'язки між таблицями для забезпечення цілісності даних;

- розробити користувацький інтерфейс, який забезпечить інтуїтивно зрозумілий доступ до функціоналу системи обліку ресурсів, включаючи додавання, редагування, видалення, пошук та фільтрацію даних;

- реалізувати функціонал автоматизованого обліку ресурсів, що охоплює реєстрацію нових активів, відстеження їх переміщення, списання застарілих ресурсів та формування звітів за визначеними критеріями;

- провести тестування розробленої системи для виявлення та усунення можливих помилок, а також оцінити її продуктивність та відповідність вимогам;

- розробити документацію для користувачів та адміністраторів для З метою глибокого аналізу функціональних та структурних аспектів досліджуваної предметної області, вважається доцільним використання формалізованих моделей. У цьому контексті застосовано методологію функціонального моделювання IDEF0 (Integrated DEFinition), що дозволило візуалізувати ключові взаємозв'язки та процеси взаємодії між основними компонентами програмної системи, що розробляється.

IDEF0 [2] – це методологія функціонального моделювання та графічна нотація, яка використовується для формалізації та опису бізнес-процесів. Загалом, вона зосереджена на визначенні взаємозв'язків та залежностей між різними видами діяльності.

У контексті системи планування ресурсів навчального закладу, IDEF0 дає змогу чітко відобразити всі ключові аспекти: від вхідних ресурсів (фінансових,

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		12

людських, інформаційних, матеріальних) до вихідних результатів (документації, звітності). Ця методологія також дозволила продемонструвати, як нормативні документи та внутрішні положення регулюють процеси, а організаційно-технічні засоби та персонал забезпечують їх виконання.

Концептуальне представлення досліджуваної системи відображено на контекстній IDEF0-діаграмі нульового рівня (A-0), візуалізація якої наведена на рисунку 1.1.1.

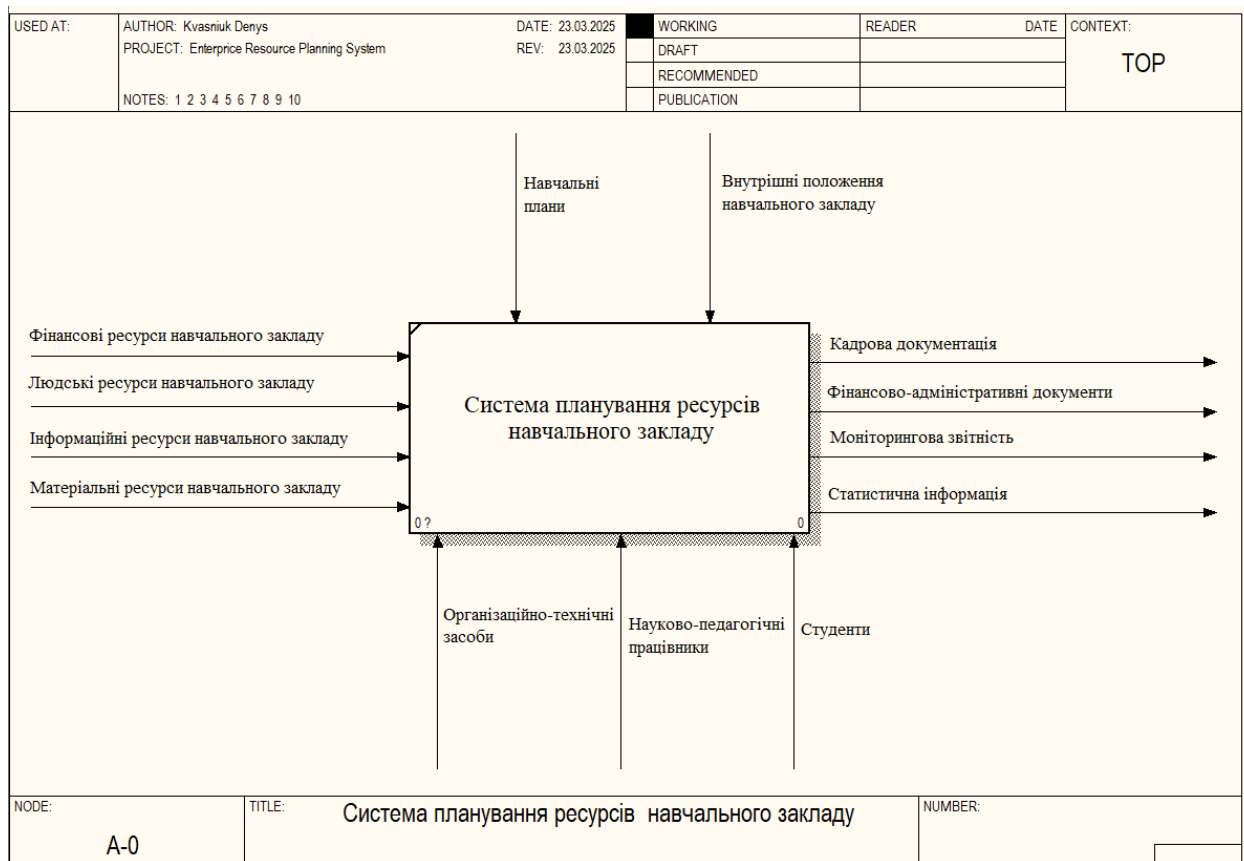


Рисунок 1.1.1 – Діаграма IDEF-0 (A-0)

Такий підхід до структурної декомпозиції дає змогу чітко окреслити основні складові системи планування ресурсів навчального закладу та встановити між ними логічні зв'язки.

Діаграма ілюструє комплексний метод управління ресурсами, що враховує як матеріальні, так і нематеріальні активи освітньої установи. Особливу увагу приділено документообігу та звітності, які є критично важливими для ефективної роботи навчального закладу.

На рисунку 1.1.2, в свою чергу зображено діаграму декомпозиції.

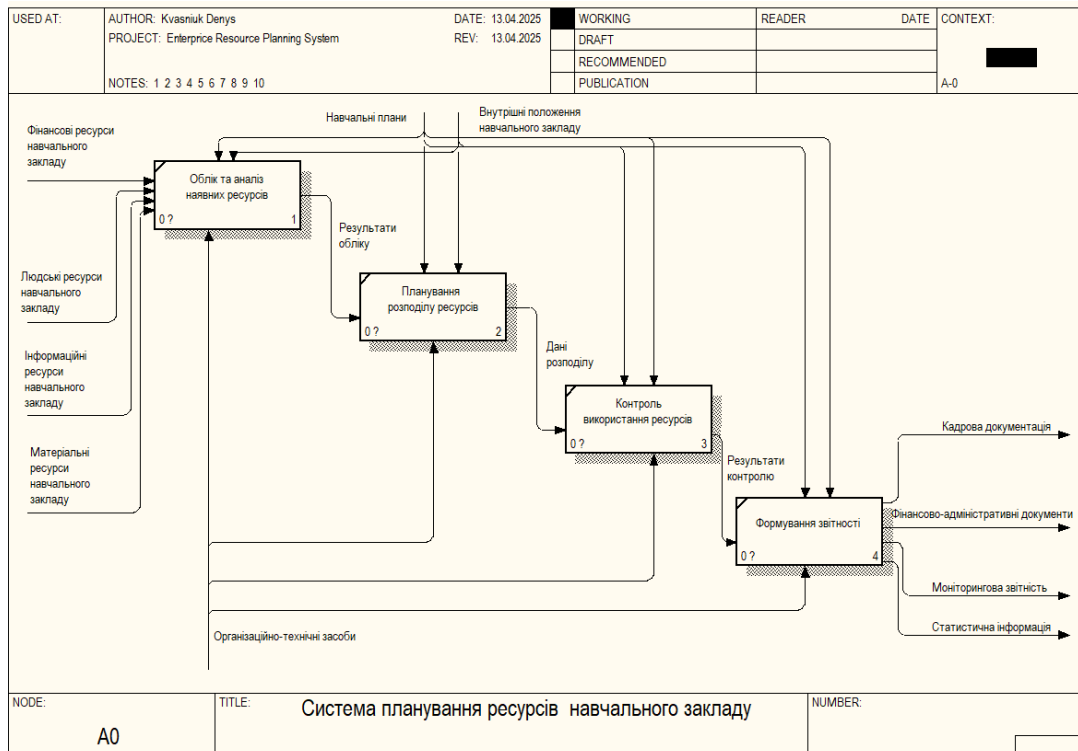


Рисунок 1.1.2 – Діаграма декомпозиції

Початковий процес (A1) охоплює всебічний облік та аналіз наявного ресурсного потенціалу закладу, включаючи фінансові, людські, інформаційні та матеріальні активи. На цьому етапі створюється єдина інформаційна база щодо поточного стану ресурсного забезпечення.

Другий етап (A2) полягає у стратегічному та оперативному плануванні розподілу ресурсів на основі узагальнених даних попереднього етапу. Результати планування визначають оптимальне використання наявних ресурсів відповідно до потреб освітнього процесу.

Третій етап (A3) забезпечує систематичний контроль за цільовим використанням розподілених ресурсів, що дозволяє відстежувати ефективність їх застосування та відповідність запланованим показникам.

Завершальний етап (A4) передбачає формування комплексної звітності, яка включає кадрову документацію, фінансово-адміністративні документи, моніторингові звіти та статистичну інформацію для прийняття рішень.

Функціонування всієї системи регулюється навчальними планами та внутрішніми нормативними документами закладу, а реалізація процесів забезпечується наявними організаційно-технічними засобами.

1.2 Аналіз наявного програмно-технічного забезпечення предметної області

У межах дослідження предметної області було здійснено детальний аналіз існуючих програмних рішень для планування ресурсів навчальних закладів. Мета цього аналізу – вивчити досвід провідних розробників програмного забезпечення та застосувати їхні найкращі практики при створенні системи планування ресурсів на базі Microsoft Dynamics 365. Такий підхід дозволить розробити продукт, що відповідає сучасним ринковим вимогам та ефективно вирішує завдання планування ресурсів навчальних закладів.

SAP for Education [4] є комплексним ERP-рішенням, спеціально розробленим для управління освітніми установами різного рівня. Система автоматизує ключові бізнес-процеси навчального закладу, включаючи фінансове управління, адміністрування студентів, управління людськими ресурсами та матеріально-технічною базою.

Продукт розроблений компанією SAP SE [5] (Systems, Applications & Products in Data Processing) – німецькою транснаціональною корпорацією, одним зі світових лідерів у розробці програмного забезпечення для управління бізнес-процесами підприємств.

Інтерфейс SAP for Education має модульну структуру з чітким розмежуванням функціональних зон. Основні вікна програми включають:

- панель адміністрування для управління користувачами, ролями та правами доступу;
- модуль фінансового управління з інтерфейсами для бюджетування, обліку витрат та фінансової звітності;
- модуль управління студентами, що містить вікна для реєстрації, моніторингу успішності та управління розкладом;
- інтерфейс управління ресурсами для планування використання аудиторій, обладнання та інших матеріальних активів;
- аналітичні панелі для моніторингу ключових показників ефективності.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						15
Зм.	Арк	№ докум.	Підпис	Дата		

Далі описано переваги:

- висока надійність та стабільність, забезпечені багаторічним досвідом SAP у розробці корпоративних систем;
- глибока інтеграція між модулями, що створює єдиний інформаційний, функціональний простір;
- потужні аналітичні можливості, включаючи предикативну аналітику та інструменти бізнес-аналітики;
- масштабованість, що дозволяє адаптувати рішення для навчальних закладів різного розміру;
- відповідність міжнародним стандартам безпеки та захисту даних.

Далі описано недоліки:

- висока вартість ліцензій та впровадження, що робить рішення недоступним для багатьох закладів з обмеженим бюджетом;
- необхідність залучення висококваліфікованих фахівців для підтримки та розвитку системи;
- надлишкова функціональність для невеликих закладів, що ускладнює роботу користувачів.

Blackbaud [6] є спеціалізованою платформою, призначеною для навчальних закладів, з основним акцентом на управлінні взаємовідносинами з різними категоріями зацікавлених сторін. Система надає функціонал для управління базами даних студентів, автоматизації процесів фандрайзингу, маркетингових інструментів та взаємодії з випускниками.

Розробником продукту є американська корпорація Blackbaud Inc. [7], заснована у 1981 році, яка спеціалізується на створенні програмного забезпечення для некомерційних організацій, зокрема освітніх установ.

Інтерфейс Blackbaud вирізняється сучасним дизайном та інтуїтивно зрозумілою навігацією. Ключові елементи інтерфейсу програми включають:

- консоль управління, що забезпечує доступ до всіх модулів системи;
- модуль управління студентами з інтерфейсами для реєстрації, моніторингу академічної успішності та управління комунікаціями;

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						16
Зм.	Арк	№ докум.	Підпис	Дата		

- інтерфейс управління фандрайзингом, що містить інструменти для планування кампаній, відстеження пожертв та управління грантами;
- модуль взаємодії з випускниками, який надає інструменти для підтримки зв'язків та організації заходів;
- аналітичні панелі з візуалізацією ключових показників та можливістю формування звітів.

Далі описано переваги:

- спеціалізація на потребах освітніх установ, що забезпечує високу відповідність функціоналу специфічним вимогам навчальних закладів;
- потужні інструменти для управління взаємовідносинами з різними категоріями зацікавлених сторін;
- інтуїтивно зрозумілий інтерфейс, що полегшує освоєння системи новими користувачами;
- хмарна архітектура, що надає доступ до системи з будь-якого місця та типу пристрою;
- регулярні оновлення та розвиток функціоналу відповідно до введених змін у сфері освіти.

Далі описано недоліки:

- обмежені можливості у сфері планування матеріальних ресурсів навчального закладу;
- недостатня гнучкість для адаптації до специфічних бізнес-процесів українських навчальних закладів;
- відсутність локалізації для українського ринку, включаючи мовну адаптацію та відповідність нормативно-правовим вимогам;
- обмежені можливості інтеграції з іншими інформаційними системами, що використовуються в навчальних закладах;
- висока вартість передплати для доступу до повного функціоналу програмної системи.

Ellucian Banner [8] – це інтегрована система, розроблена спеціально для вищих навчальних закладів, що забезпечує комплексну автоматизацію

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						17
Зм.	Арк	№ докум.	Підпис	Дата		

академічних процесів, фінансового менеджменту, управління персоналом та формування звітності.

Продукт створений американською корпорацією Ellucian [9], яка спеціалізується на розробці програмного забезпечення для вищої освіти. Компанія була заснована у 2012 році шляхом злиття Datatel та SunGard Higher Education і наразі є одним із провідних гравців на ринку ERP-систем для освітніх установ.

Інтерфейс Ellucian Banner характеризується функціональністю та орієнтацією на потреби різних категорій користувачів. Основні компоненти інтерфейсу включають:

- адміністративну панель для управління користувачами та налаштуваннями системи;
- модуль управління академічними процесами з інструментами для планування наявних навчальних програм, створення розкладу та моніторингу успішності студентів;
- фінансовий модуль з функціоналом для бюджетування, обліку витрат та фінансової звітності;
- модуль управління персоналом, що містить інструменти для планування навантаження викладачів та оцінки їх ефективності;
- інтерфейс для формування звітності з можливістю створення як стандартних, так і модифікованих звітів.

Далі описано переваги:

- глибока спеціалізація на потребах вищих навчальних закладів, що забезпечує високу відповідність функціоналу;
- комплексний підхід до автоматизації бізнес-процесів, що дозволяє створити єдиний інформаційний простір;
- гнучкість налаштувань відповідно до організаційної структури та процесів конкретного навчального закладу;
- потужні інструменти для планування та оптимізації використання всіх наявних ресурсів;

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						18
Зм.	Арк	№ докум.	Підпис	Дата		

- широкі можливості для інтеграції з іншими системами та сервісами.

Далі описано недоліки:

- складний інтерфейс, що вимагає тривалого навчання користувачів;
- висока вартість ліцензій та впровадження, що обмежує доступність рішення для багатьох навчальних закладів;
- орієнтація на американську модель вищої освіти, що ускладнює адаптацію до українських реалій;
- потреба у значних обчислювальних ресурсах для забезпечення стабільної роботи системи;
- тривалий цикл впровадження, що може негативно впливати на операційну діяльність навчального закладу.

Microsoft Dynamics 365 Business Central [10] є комплексним ERP-рішенням, розробленим корпорацією Microsoft. Яке призначене для малого та середнього бізнесу, з можливістю адаптації для потреб навчальних закладів. Система автоматизує ключові бізнес-процеси, такі як фінансовий менеджмент, управління закупівлями, запасами, проектами, а також надає потужні інструменти для аналітики та звітності.

Продукт розроблений корпорацією Microsoft [11], однією з найбільших технологічних компаній світу, заснованою в 1975 році. Microsoft має значний досвід у розробці корпоративних систем та розгалужену партнерську мережу для впровадження та підтримки своїх рішень.

Інтерфейс Microsoft Dynamics 365 Business Central характеризується сучасним дизайном та інтеграцією з іншими продуктами Microsoft. Основні елементи інтерфейсу включають:

- єдину панель навігації з доступом до всіх модулів системи;
- модуль управління взаємовідносинами зі студентами, що охоплює інструменти для відстеження повного життєвого циклу студента;
- фінансовий модуль з інтерфейсами для бюджетування, обліку витрат та фінансової аналітики;
- модуль управління людськими ресурсами з інструментами;

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						19
Зм.	Арк	№ докум.	Підпис	Дата		

- модуль управління матеріальними ресурсами для планування використання аудиторій, обладнання та інших активів;
- аналітичні панелі на базі Power BI для візуалізації ключових показників та прийняття управлінських рішень.

Далі описано переваги:

- глибока інтеграція з іншими продуктами Microsoft (Office 365, SharePoint, Teams), що створює єдине інформаційне середовище;
- гнучка модульна архітектура, що дозволяє поетапне впровадження та вибір лише необхідних компонентів;
- потужні інструменти для аналізу даних та формування звітності на базі Power BI;
- хмарна архітектура, що забезпечує доступ до системи з будь-якого місця та пристрою;
- широкі можливості для модифікації та розширення функціоналу відповідно до специфічних потреб навчального закладу;
- наявність локальної підтримки та партнерської мережі в Україні.

Далі описано недоліки:

- відносно висока вартість ліцензій;
- необхідність додаткових налаштувань для повної відповідності специфічним процесам українських навчальних закладів;
- складність впровадження без присутності сертифікованих компаній-партнерів Microsoft;
- потреба в регулярному оновленні системи для забезпечення доступу до нових функцій;
- залежність від екосистеми Microsoft, що може обмежувати інтеграцію з рішеннями інших постачальників.

Для визначення оптимального підходу до розробки системи планування ресурсів для навчального закладу було проведено порівняльний аналіз розглянутих рішень за ключовими критеріями, результати якого представлені в таблиці 1.2.1

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						20
Зм.	Арк	№ докум.	Підпис	Дата		

Таблиця 1.2.1 – Аналіз ERP-рішень для навчальних закладів

Критерій	SAP for Education	Blackbaud	Ellucian Banner	Microsoft Dynamics 365
Функціональна повнота	Висока	Середня	Висока	Висока
Гнучкість налаштування	Середня	Середня	Середня	Висока
Інтеграційні можливості	Високі	Середні	Високі	Високі
Вартість впровадження	Дуже висока	Висока	Висока	Середня
Складність впровадження	Висока	Середня	Висока	Середня
Локалізація	Часткова	Відсутня	Відсутня	Наявна
Хмарне розгортання	Доступне	Основний варіант	Доступне	Основний
Аналітичні можливості	Високі	Середні	Середні	Високі
Масштабованість	Висока	Середня	Висока	Висока
Мобільний доступ	Обмежений	Наявний	Обмежений	Повний

На основі проведеного аналізу можна дійти висновку, що Microsoft Dynamics 365 є найбільш оптимальним базовим рішенням для розробки системи планування ресурсів навчального закладу в українських умовах.

Водночас, розробка системи планування ресурсів на базі Microsoft Dynamics 365 має враховувати специфічні потреби українських навчальних закладів та включати додаткові функціональні модулі, відсутні у стандартній конфігурації. Такий підхід дозволить створити рішення, що поєднує переваги готової платформи з можливостями, необхідними для планування ресурсів навчального закладу.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		21

1.3 Визначення функціональних та нефункціональних вимог до програмного забезпечення

Чітке визначення вимог є критично важливим етапом у процесі розробки програмного забезпечення. Цей етап формує основу для всього проекту, окреслюючи його межі, функціональність та якісні характеристики.

Вимоги до програмного забезпечення традиційно класифікуються на функціональні та нефункціональні [12]. Функціональні вимоги описують, що саме система повинна виконувати, які функції та можливості вона має надавати користувачам. Нефункціональні вимоги, зі свого боку, визначають, якими характеристиками має володіти система, включаючи продуктивність, надійність, безпеку, зручність використання та інші якісні показники.

Для визначення вимог до системи планування ресурсів навчального закладу було проведено всебічний аналіз потреб різних категорій користувачів, таких як адміністрація, викладачі, студенти та технічний персонал. Також було враховано специфіку діяльності навчальних закладів в Україні, нормативні вимоги Міністерства освіти і науки України та передові практики у сфері управління освітніми установами.

Особливу увагу приділено інтеграції з платформою Microsoft Dynamics 365, обраною як базова для розробки системи. Це дозволило врахувати як можливості, так і обмеження даної платформи при формулюванні вимог до програмного забезпечення.

Результатом цієї роботи стало формування комплексного набору функціональних та нефункціональних вимог, детально описаних у наступних підрозділах. Ці вимоги слугують основою для подальшого проектування архітектури системи, розробки програмного коду та тестування продукту.

Функціональні вимоги [13] визначають специфіку поведінки системи, її ключові функції та можливості, необхідні для ефективного планування ресурсів у навчальному закладі. Ці вимоги встановлюють, які завдання система має виконувати, які дані обробляти та які результати надавати користувачам.

					КВРППЗ. 2101075.01.06.ПЗ	Арк.
						22
Зм.	Арк	№ докум.	Підпис	Дата		

Формування функціональних вимог базується на аналізі бізнес-процесів навчального закладу, потреб різних категорій користувачів та освітніх нормативних стандартів. Вони охоплюють усі основні аспекти діяльності, включаючи управління навчальним процесом, людськими та матеріальними ресурсами, а також фінансове планування та аналітику. Нижче розглянуто функціональні вимоги для кожного з цих напрямків, які, попри свою специфіку, мають бути інтегровані в єдину систему для забезпечення комплексного управління навчальним закладом.

Система повинна надавати інструменти для формування та управління навчальними планами, зокрема створення, редагування та архівування навчальних програм. Користувачі повинні мати можливість визначати структуру навчальних планів, включаючи перелік дисциплін, обсяг годин, форми контролю та кредити ECTS [14].

У сфері управління людськими ресурсами система має забезпечувати ведення бази даних викладачів та співробітників, що містить інформацію про їхню кваліфікацію, спеціалізацію, наукові ступені та вчені звання. Ключовою функцією є планування та розподіл навчального навантаження викладачів з урахуванням їхньої кваліфікації, спеціалізації та побажань, а також моніторинг виконання цього навантаження.

Система має забезпечувати генерацію стандартних звітів відповідно до вимог Міністерства освіти і науки України та внутрішніх потреб закладу. Важливою є можливість створення користувацьких звітів для різних категорій користувачів. Система повинна підтримувати аналіз ефективності використання ресурсів із візуалізацією ключових показників та динаміки. Необхідно передбачити можливість експорту звітів у форматах PDF, Excel та Word для подальшого використання та розповсюдження.

Система повинна забезпечувати інтеграцію з іншими інформаційними системами навчального закладу, такими як системи електронного документообігу, електронного навчання та бухгалтерського обліку. Важливою є можливість імпорту та експорту даних у стандартних форматах.

					КВРППЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		23

Нефункціональні вимоги [15] визначають якісні характеристики системи, що впливають на її продуктивність, надійність, безпеку та зручність. На відміну від функціональних, вони описують не що робить система, а як вона це робить.

При визначенні нефункціональних вимог до системи планування ресурсів було враховано специфіку діяльності українських навчальних закладів, технічні характеристики платформи Microsoft Dynamics 365, а також актуальні стандарти розробки програмного забезпечення. Особливу увагу було приділено таким аспектам, як продуктивність та масштабованість, надійність та доступність, безпека та захист даних, зручність використання, підтримка та супровід, а також відповідність нормативним вимогам.

Далі деталізовано нефункціональні вимоги за кожним із зазначених аспектів, які разом визначають якісні характеристики системи.

Система має підтримувати одночасну роботу не менше 500 користувачів без помітної втрати продуктивності. Час відгуку для стандартних операцій не повинен перевищувати 2 секунд, для аналітичних — 10 секунд. Дані повинні зберігатися з можливістю масштабування системи при зростанні навантаження.

Система має забезпечувати надійну аутентифікацію користувачів із використанням сучасних методів, зокрема багатофакторної аутентифікації, а також чітке розмежування прав доступу відповідно до ролей і структури закладу для безпечної роботи.

Інтерфейс повинен бути інтуїтивним і простим у використанні, з адаптивним дизайном для роботи на різних пристроях — комп'ютерах, планшетах і смартфонах.

Окрему увагу варто приділити підтримці багатомовності: інтерфейс повинен бути локалізований принаймні українською та англійською мовами, а також мати гнучкий механізм додавання нових мов за потреби.

Запропонований набір вимог є комплексним та охоплює всі основні напрями функціонування навчального закладу. Такий підхід дозволяє забезпечити цілісну інтеграцію інформаційних потоків та ефективне планування ресурсів у межах єдиної цифрової екосистеми.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		24

1.4 Висновки дослідження предметної області та постановки задачі.

У першому розділі кваліфікаційної роботи проведено комплексний аналіз предметної області планування ресурсів навчального закладу. Визначено структурні та функціональні особливості досліджуваної сфери, де ключовими категоріями ресурсів виступають матеріальні, людські, фінансові та інформаційні активи.

За допомогою методології функціонального моделювання IDEF0 візуалізовано основні процеси системи: облік та аналіз ресурсного потенціалу, стратегічне й оперативне планування розподілу ресурсів, контроль за їх використанням та формування звітності. Створені діаграми демонструють логічні взаємозв'язки між компонентами системи та дають цілісне уявлення про функціонування навчального закладу.

Здійснено порівняльний аналіз існуючих програмних рішень у даній галузі, зокрема SAP for Education, Blackbaud, Ellucian Banner та Microsoft Dynamics 365 Business Central.

На основі оцінки за ключовими критеріями (функціональна повнота, гнучкість налаштування, інтеграційні можливості, вартість впровадження, локалізація та інші) визначено, що Microsoft Dynamics 365 є оптимальною платформою для розробки системи планування ресурсів в умовах українських навчальних закладів.

Сформульовано детальний перелік функціональних вимог до програмного забезпечення, які охоплюють управління навчальним процесом, людськими ресурсами, фінансами, матеріально-технічною базою, звітністю та інтеграцією. Визначено також нефункціональні вимоги щодо продуктивності, надійності, безпеки, зручності використання та відповідності нормативним документам.

Таким чином, проведений аналіз створює ґрунтовну базу для подальшої розробки системи планування ресурсів навчального закладу, яка відповідатиме сучасним потребам освітніх установ та забезпечуватиме ефективне управління їхніми ресурсами.

					КВРІПЗ. 2101075.01.06.ПЗ	Арк.
						25
Зм.	Арк	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ СИСТЕМИ ПЛАНУВАННЯ РЕСУРСІВ

2.1 Проєктування архітектури та структури системи

При розробці програмної системи вибір оптимальної структури є одним із ключових етапів. Правильно спроектована структура програмного продукту визначає розміщення модулів, їх комунікацію та принципи функціонування всієї системи. Вона охоплює ключові рішення щодо організації цифрового рішення, включно з набором базових компонентів, їхніх з'єднань та взаємодії між ними. Одним із головних призначень структурної організації програми є спрощення алгоритмів функціонування при створенні комплексних цифрових систем, як системи планування ресурсами.

Тому перед початком розробки було детально описано схему будови програмного продукту на базі Microsoft Dynamics 365. Першим етапом у проєктуванні схеми системи був вибір моделі розгортання, яка визначає подальший успіх впровадження та експлуатації інформаційної системи. Особливо це стосується комплексних програмних систем, наприклад систем планування ресурсів.

Розробляючи систему планування ресурсів навчального закладу, необхідно ретельно зважити всі переваги та недоліки різних підходів до розгортання, враховуючи специфіку освітньої галузі, особливості бізнес-процесів навчального закладу та наявні ресурси. Microsoft Dynamics 365 пропонує дві основних моделі розгортання: традиційне локальної (On-Premises) та хмаране (Software as a Service, SaaS). Саме тому в рамках даної дипломної роботи було вирішено провести ґрунтовний аналіз саме цих двох основних моделей розгортання.

On-Premises [16] – це модель розгортання, при якій ІТ-інфраструктура, системи, програмне забезпечення та дані, розташовані на території організації, наприклад, в офісних будівлях або центрах обробки даних. Організація володіє та управляє цією інфраструктурою на місці, на відміну від хмарних обчислень, де послуги надаються сторонніми провайдерами через Інтернет.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		26

При локальному розгортанні Microsoft Dynamics 365 навчальний заклад повинен забезпечити відповідну інфраструктуру: придбати сервери, системи зберігання даних, мережеве обладнання, ліцензії на операційні системи та системи управління базами даних. Крім того, необхідно мати кваліфікований ІТ-персонал для встановлення, налаштування та подальшого обслуговування цілої системи.

Локальне розгортання надає навчальному закладу повний контроль над системою та даними. Адміністратори можуть самостійно визначати політики безпеки, налаштовувати системні параметри, планувати оновлення та резервне копіювання. Це особливо важливо для навчальних закладів, які працюють з конфіденційними даними студентів та викладачів і мають специфічні вимоги щодо їх зберігання та обробки.

Однак, разом з перевагами контролю, локальне розгортання має суттєві недоліки. Насамперед, це значні початкові інвестиції в інфраструктуру та ліцензії. Навчальний заклад повинен не лише придбати обладнання та програмне забезпечення, але й забезпечити відповідні умови для їх розміщення.

Крім того, локальне розгортання вимагає постійних витрат на підтримку та оновлення системи. Навчальний заклад повинен мати штат ІТ-спеціалістів або залучати зовнішніх консультантів для обслуговування системи, що також збільшує загальну вартість розгортання системи.

SaaS [17] – це хмарна модель надання програмного забезпечення, за якої окремі особи або організації підписуються на додатки замість того, щоб купувати та встановлювати їх локально. Клієнти отримують доступ до програмного забезпечення через Інтернет, зазвичай через веб-браузер, а постачальник хмарних послуг керує базовою інфраструктурою, безпекою, обслуговуванням та усіма оновленнями системи. При розгортанні за моделлю SaaS, навчальний заклад отримує доступ до повнофункціонального рішення без необхідності інвестувати в інфраструктуру та займатися технічним обслуговуванням системи. Усі серверні компоненти, бази даних та системне програмне забезпечення розміщуються в центрах обробки даних Microsoft.

Однією з ключових переваг SaaS-моделі є передбачуваність витрат. Замість значних початкових інвестицій, навчальний заклад сплачує регулярну абонентську плату, яка залежить від кількості користувачів та набору функцій. Це дозволяє більш ефективно планувати бюджет та уникати непередбачених витрат на обслуговування інфраструктури.

Крім того, SaaS-модель забезпечує автоматичне оновлення системи до найновіших версій, що дозволяє навчальному закладу завжди використовувати актуальні функції та отримувати виправлення безпеки без необхідності планування та проведення складних процедур оновлення. Також модель спрощує масштабування та зменшує навантаження на ІТ-відділ, оскільки підтримку забезпечує постачальник.

Узагальнений порівняльний аналіз основних моделей розгортання був зведений у Таблицю 2.1.1.

Таблиця 2.1.1 – Порівняльний аналіз моделей розгортання

Характеристика	On-Premises	SaaS
Інфраструктура	Потребує придбання серверів, систем зберігання даних	Не потребує власної інфраструктури
Початкові інвестиції	Значні (обладнання, ліцензії, приміщення)	Мінімальні (відсутні капітальні витрати)
Модель оплати	Одноразова покупка ліцензій, постійні витрати на обслуговування	Регулярна абонентська плата залежно від кількості користувачів
Контроль	Повний контроль над системою та даними	Обмежений контроль, залежність від провайдера
Безпека даних	Повний контроль над політиками безпеки	Відповідальність за безпеку розподілена

Продовження таблиці 2.1.1

Оновлення	Потребує планування та проведення оновлень	Автоматичні оновлення без додаткових зусиль
Обслуговування	Потребує власного ІТ-персоналу	Обслуговується провайдером
Масштабованість	Обмежена наявною інфраструктурою, потребує інвестицій	Легко масштабується відповідно до потреб
Доступність	Доступ переважно з локальної мережі	Доступ з будь-якого місця через Інтернет
Передбачуваність витрат	Менш передбачувані витрати через можливі непередбачені витрати на обслуговування	Висока передбачуваність витрат завдяки фіксованій платі
Відповідальність	Повністю на організації	Переважно на провайдері послуг

Отже, за результатами порівняльного аналізу наявних моделей розгортання було вирішено обрати модель SaaS для реалізації поточного рішення. Вирішальними факторами виступили легка масштабованість та низькі початкові інфраструктурні інвестиції.

Наступним етапом у проектуванні став огляд базової архітектури, на якій базується Microsoft Dynamics 365, та порівняльний аналіз розширеної архітектури. Загальне функціонування архітектури Microsoft Dynamics 365 базується на багаторівневій структурі, що забезпечує високу доступність, масштабованість та безпеку даних.

Архітектура включає рівень представлення (клієнтські додатки), рівень бізнес-логіки (сервери додатків) та рівень даних (бази даних SQL Server).

Багаторівнева архітектура [18] – це клієнт-серверна архітектура, в якій презентація, обробка прикладної логіки та управління даними є логічно окремими процесами. Вона складається з трьох основних рівнів:

– рівень представлення (Presentation Tier), який відповідає за інтерфейс користувача та взаємодію з користувачем. Цей рівень представлений клієнтськими додатками, які можуть бути веб-браузерами, мобільними додатками або настільними програмами. Основна функція цього рівня - відображення інформації та збір даних від користувача;

– рівень бізнес-логіки (Business Logic Tier), також відомий як рівень додатків або середній рівень. Цей рівень містить функціональну логіку системи, обробляє команди, приймає рішення, виконує розрахунки та переміщує дані між двома іншими рівнями. Він представлений серверами додатків, які обробляють бізнес-правила та логіку;

– рівень даних (Data Tier) – відповідає за зберігання та отримання даних. Цей рівень складається з систем управління базами даних, які зберігають дані та надають доступ до них.

На рисунку 2.1.1 зображено схему багаторівневої архітектури.

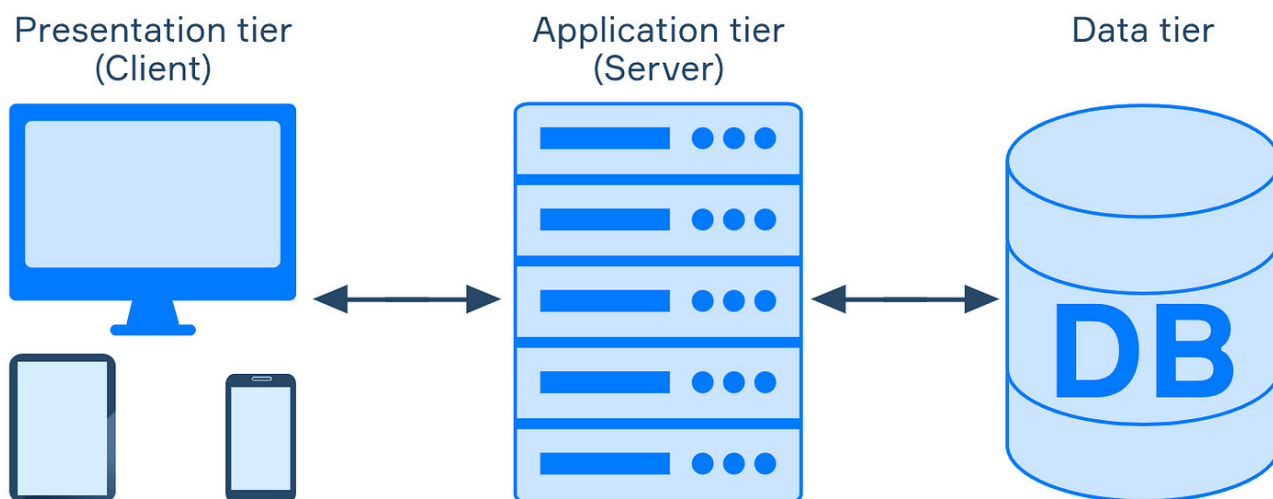


Рисунок 2.1.1 – Схема багаторівневої архітектури

У контексті обраної SaaS-моделі розгортання є доступними декілька архітектурних підходів, які визначають спосіб організації даних та ресурсів для користувачів системи.

Архітектура Single-Tenant [19] представляє собою тип архітектури, при якому кожен клієнт (організація) отримує виділену копію програмного забезпечення та інфраструктури, тобто кожен клієнт має власний екземпляр програмного забезпечення, бази даних та пов'язаних ресурсів. Також передбачає виділення окремих серверів та баз даних для кожного клієнта. Кожен клієнт має власний виділений сервер у шарі Business Central Service та власну базу даних у шарі SQL Server. Це забезпечує повну ізоляцію даних та обчислювальних ресурсів між різними клієнтами.

Архітектура Single-Tenant має ряд переваг, особливо для організацій з високими вимогами до безпеки даних та продуктивності. Повна ізоляція даних забезпечує вищий рівень безпеки, оскільки дані різних клієнтів фізично розділені. Це також дозволяє організаціям мати більший контроль над своїм середовищем, включаючи можливість планування оновлень та налаштування системи відповідно до специфічних потреб.

Однак, цей тип архітектури зазвичай вимагає більших витрат на інфраструктуру та обслуговування оскільки ресурси не розподіляються між кількома клієнтами.

Архітектура Multi-Tenant [20] – представляє собою тип архітектури, при якому один сервіс програмного забезпечення обслуговує багатьох клієнтів. У цій архітектурі всі клієнти використовують одну й ту саму інфраструктуру та поточну версію програмного забезпечення, але їхні дані логічно ізольовані один від одного.

При використанні цієї організація отримує доступ до спільної інфраструктури, яка обслуговується та підтримується Microsoft. Усі клієнти використовують одну й ту саму версію програмного забезпечення, що дозволяє Microsoft ефективно керувати ресурсами, забезпечувати оновлення та постійну підтримку середовищ.

Однією з ключових переваг цієї архітектури є економічна ефективність, швидке та безперебійне оновлення системи, оскільки Microsoft керує процесом оновлення централізовано, що дозволяє всім клієнтам одночасно отримувати

доступ до нових функцій та виправлень безпеки без необхідності планування та проведення складних процедур оновлення.

Однак, разом з перевагами, ця архітектура має певні обмеження щодо налаштування. Оскільки всі клієнти використовують одну й ту саму версію програмного забезпечення, можливості для глибокої модифікації системи можуть бути обмеженими.

Узагальнений порівняльний аналіз основних моделей розгортання був зведений у Таблицю 2.1.2.

Таблиця 2.1.2 – Порівняльний аналіз архітектурних підходів

Характеристика	Single-tenant	Multi-tenant
Ізоляція даних	Повна фізична ізоляція даних та інфраструктури	Логічна ізоляція даних на спільній інфраструктурі
Налаштування	Широкі можливості для глибокої кастомізації	Обмежені можливості, стандартизовані налаштування
Безпека	Вищий рівень безпеки завдяки повній ізоляції	Залежність від логічної ізоляції
Вартість	Вища вартість через виділені ресурси	Нижча вартість завдяки спільним ресурсам

Отже, за результатами порівняльного аналізу наявних архітектурних підходів було вирішено обрати Single-Tenant архітектуру.

Вирішальними факторами виступили високий рівень безпеки та конфіденційності даних завдяки повній фізичній ізоляції, повний контроль над інфраструктурою та можливість модифікації системи, в залежності від специфічних бізнес-процесів.

Загальна схема функціонування архітектурного підходу Single-Tenant наявна на рисунку 2.1.2.

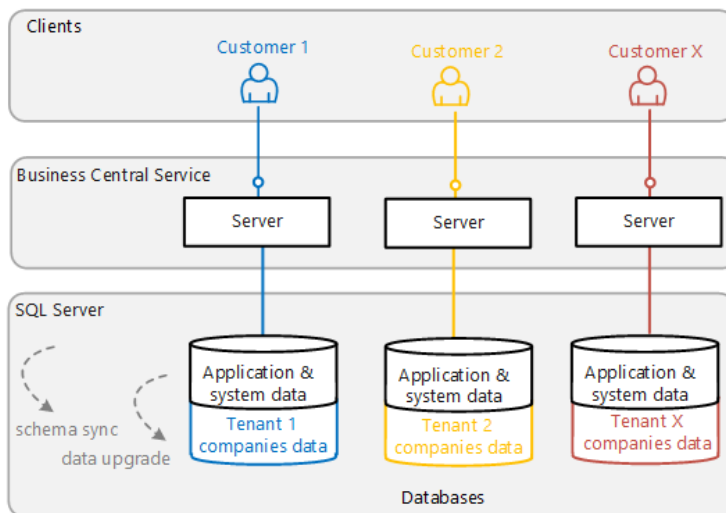


Рисунок 2.1.2 – Схема архітектури Single-Tenant

Таким чином, за результатами порівняльних аналізів методів розгортання та архітектурних підходів було сформовано архітектуру та структуру програмної системи, яка представлена у вигляді діаграми розгортання на рисунку 2.1.3.

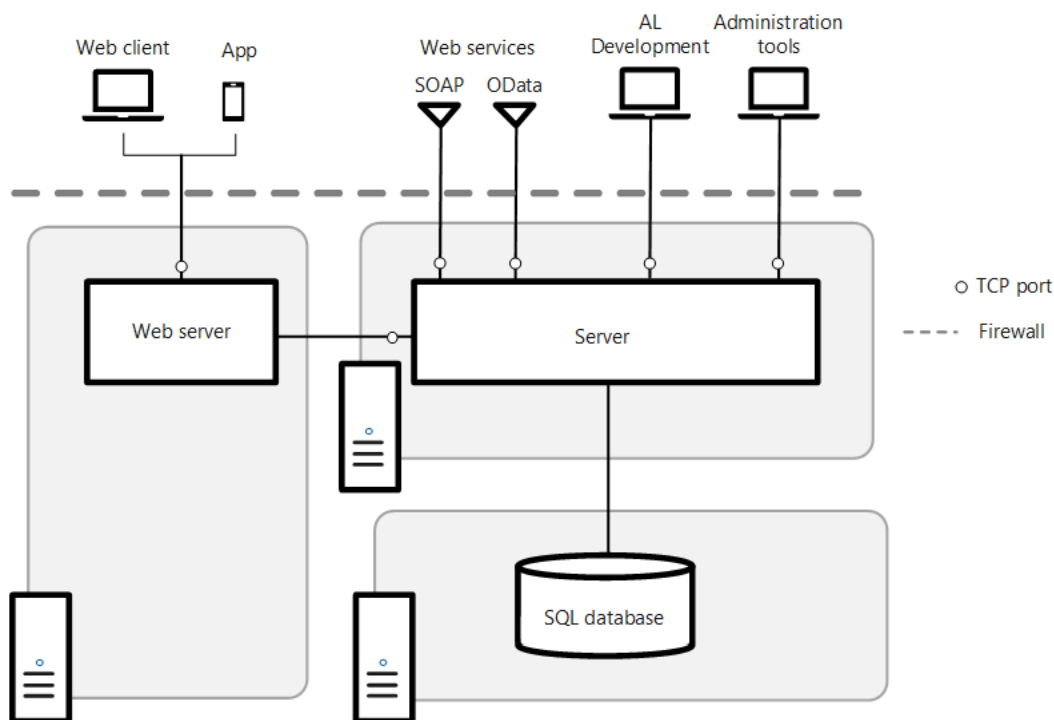


Рисунок 2.1.3 – Діаграма розгортання

2.2 Проектування логічної моделі бази даних

Проектування логічної моделі бази даних є одним із найважливіших етапів розробки інформаційної системи. Саме на цьому етапі формується структура даних, яка визначатиме ефективність, надійність та масштабованість майбутньої програмної системи. У контексті розробки програмного забезпечення, проектування логічної моделі бази даних набуває особливого значення, оскільки від правильності її побудови залежить успішність інтеграції з існуючими системами та можливість подальшого розвитку.

Логічна модель бази даних [21] – це абстрактне представленням даних, незалежне від конкретної системи управління базами даних, яке і описує структуру даних, їх взаємозв'язки та обмеження.

Перш ніж перейти до побудови логічної моделі, необхідно було уніфікувати та стандартизувати її складові, тобто забезпечити, щоб всі сутності, атрибути та зв'язки були чітко визначені, описані та відповідали єдиним правилам і підходам.

Першим кроком у стандартизації логічної моделі бази даних є визначення переліку сутностей та інформації, яка зберігатиметься у базі даних. У контексті проектування бази даних, сутність [22] – це об'єкт реального світу, концепція, явище, інформацію про яке необхідно зберігати в базі даних.

За результатами первинного аналізу функціональних та структурних особливостей предметної області було виділено наступний набір сутностей.

Студент – особа, яка навчається у навчальному закладі. Містить інформацію про особисті дані студента, його академічний статус, групу, факультет, кафедру.

Група – об'єднання студентів для спільного навчання. Містить інформацію про назву групи, факультет, кафедру, куратора.

Факультет – структурний підрозділ навчального закладу, що об'єднує кафедри. Містить інформацію про назву факультету, декана, кількість кафедр, співробітників та студентів.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		34

Кафедра – структурний підрозділ факультету, що здійснює навчальну, методичну та наукову діяльність. Містить інформацію про назву кафедри, факультет, завідувача, кількість співробітників та студентів.

Академічний співробітник – особа, яка працює у навчальному закладі та здійснює навчальну, методичну та наукову діяльність. Містить інформацію про особисті дані співробітника, його науковий ступінь, факультет, кафедру тощо.

Навчальна дисципліна – предмет, який вивчається студентами. Містить інформацію про назву дисципліни, кількість кредитів, форми контролю тощо.

Оцінка з дисципліни – результат оцінювання знань студента з певної дисципліни. Містить інформацію про студента, дисципліну, оцінку, дату оцінювання тощо.

Наступним кроком у стандартизації логічної моделі бази даних є визначення набору атрибутів для кожної наявної сутності. Цей етап є критично важливим для забезпечення цілісності та логічної узгодженості всієї системи. Правильно визначені атрибути дозволяють точно описати характеристики кожної сутності та забезпечити коректну обробку даних у майбутньому. На основі доступної інформації про загальну модель функціонування системи, було ідентифіковано окремі сутності з характерними для них атрибутами. Ці сутності разом з відповідними наборами атрибутів було структуровано у формі таблиць та наповнено репрезентативними даними (Таблиці 1-6).

Таблиця 2.2.1 – Сутність «Студент»

id	No.	Name	Birth_Day	Faculty	Department	Group	Form	...
1	ST001	Круть Олександр Петрович	15.07.2001	FI	DCS	ІПЗ-21	FullTime	...
2	ST002	Верть Марія Іванівна	23.04.2000	FE	DCS	ЕК-20	FullTime	...
3	ST003	Вервет Андрій Степанович	12.09.1998	FL	DCS	ПП-19	PartTime	...

Таблиця 2.2.2 – Сутність «Група»

id	No.	Abbreviation	Name	Faculty	Department	Curator	Student
1	GR001	ПЗ-21	Інженерія ПЗ	FI	DCS	TE001	25
2	GR002	ЕК-20	Економіка підприємства	FE	DEC	TE002	22
3	GR003	ПР-19	Правознавство	FL	DLW	TE003	18

Таблиця 2.2.3 – Сутність «Факультет»

id	No.	Abbreviation	Name	Dean	Departments	Employees	Student
1	FI	FI	Факультет інформатики	TE004	3	45	350
2	FE	FE	Факультет економіки	TE005	4	52	420
3	FL	FL	Факультет права	TE006	2	38	280

Таблиця 2.2.4 – Сутність «Кафедра»

id	No.	Name	Faculty	Head	Employees	Students
1	DCS	Кафедра комп'ютерних наук	FI	TE007	18	150
2	DEC	Кафедра економіки	FE	TE008	15	180
3	DLW	Кафедра права	FL	TE009	12	120

Таблиця 2.2.5 – Сутність «Викладач»

id	No.	Name	Academic Degree	Faculty	Department	Image
1	TE001	Петренко Ігор Васильович	Доцент, к.т.н.	FI	DCS	teacher1.jpg
2	TE002	Сидоренко Олена Михайлівна	Професор, д.е.н.	FE	DEC	teacher2.jpg
3	TE003	Іваненко Сергій Олександрович	Доцент, к.ю.н.	FL	DLW	teacher3.jpg

Таблиця 2.2.6 – Сутність «Предмет»

id	No.	Name	Credits	Department	Teacher	Control Form
1	SB001	Програмування	5.0	DCS	TE001	Exam
2	SB002	Мікроекономіка	4.0	DEC	TE002	Exam
3	SB003	Теорія держави і права	6.0	DLW	TE003	Exam

Таблиця 2.2.7 – Сутність «Оцінка»

id	Entry_No	Student	Subject	Teacher	Grade	Date	Year	Semestr
1	1001	ST001	SB001	TE001	85	15.01.2023	2022-2023	1
2	1002	ST002	SB002	TE002	92	18.01.2023	2022-2023	1
3	1003	ST003	SB003	TE003	78	20.01.2023	2022-2023	1

Після ідентифікації ключових сутностей та визначення їхніх характеристик було вирішено розпочати етап формування взаємозв'язків між ними [23]. З цією метою було здійснено додатковий аналіз розгорнутого сценарію функціонування системи. Загалом виділяють три основних типи зв'язку між сутностями у базі даних:

Зв'язок «один до одного» (1:1) [24]. Цей тип зв'язку встановлюється, коли один запис у таблиці А відповідає лише одному запису в таблиці Б і навпаки.

Зв'язок «один до багатьох» (1:M) [25]. Найпоширеніший тип зв'язку, коли один запис у таблиці А може бути пов'язаний з багатьма записами в таблиці Б, але кожен запис у таблиці Б пов'язаний лише з одним записом у таблиці А.

Зв'язок «багато до багатьох» (M:N) [26]. Виникає, коли багато записів у таблиці А можуть бути пов'язані з багатьма записами в таблиці Б. Цей тип зв'язку реалізується через таблицю, яку зазвичай називають таблицею асоціацій.

У контексті навчального процесу кожен студент зазвичай належить до певної академічної групи, яка об'єднує здобувачів освіти за спеціальністю, роком вступу та іншими критеріями. Аналізуючи цей зв'язок, можна охарактеризувати його як «багато-до-одного», оскільки одна група може

включати значну кількість студентів (від кількох осіб до кількох десятків), тоді як кожен окремий студент у певний момент часу може належати лише до однієї групи. Важливо зазначити, що система повинна передбачати можливість існування студентів, які тимчасово не належать до жодної групи, наприклад, у випадку академічної відпустки, відрахування або переведення.

Розглядаючи зв'язок між студентом та факультетом, спостерігаємо подібну модель «багато-до-одного». Кожен факультет, як структурний підрозділ навчального закладу, об'єднує сотні або навіть тисячі студентів різних спеціальностей та років навчання. При цьому кожен студент може навчатися лише на одному факультеті в конкретний момент часу. Цей зв'язок є обов'язковим, оскільки належність до факультету є невід'ємною частиною академічного статусу студента та визначає напрям його підготовки.

Аналогічно до попереднього, зв'язок між студентом і кафедрою також характеризується як «багато-до-одного». Кафедра, будучи базовим структурним підрозділом факультету, забезпечує підготовку студентів за певними спеціальностями та спеціалізаціями. Один студент може бути закріплений лише за однією випусковою кафедрою, яка відповідає за його фахову підготовку, керує курсовими та дипломними роботами. Натомість кафедра може обслуговувати десятки або сотні студентів різних курсів та спеціальностей. Цей зв'язок також є обов'язковим для забезпечення повноцінного функціонування навчального процесу.

Розглядаючи структурну ієрархію навчального закладу, можна визначити, що зв'язок між групою та факультетом також відповідає моделі «багато-до-одного». Факультет, як більша організаційна одиниця, об'єднує множину академічних груп різних спеціальностей та років навчання.

При цьому кожна група може належати лише до одного факультету, що визначає її місце в структурі закладу та напрям підготовки студентів. Цей зв'язок є обов'язковим, оскільки кожна група повинна бути чітко закріплена за певним факультетом для правильної організації навчального процесу. Такий підхід також спрощує адміністрування та дозволяє уникнути дублювання даних.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		38

Подібно до попереднього, зв'язок між групою та кафедрою також характеризується як «багато-до-одного». Кафедра може бути випусковою для кількох академічних груп, забезпечуючи їх фахову підготовку та супровід. Водночас кожна група закріплюється за однією випусковою кафедрою, яка відповідає за реалізацію навчального плану відповідної спеціальності. Цей зв'язок є обов'язковим для ефективної організації навчального процесу та забезпечення якості підготовки фахівців.

Аналізуючи організаційну структуру навчального процесу, можна виділити зв'язок між групою та академічним співробітником у контексті кураторства. Цей зв'язок також відповідає моделі «багато-до-одного», оскільки один викладач може бути куратором для кількох академічних груп, здійснюючи виховну роботу та організаційний супровід. При цьому кожна група може мати лише одного куратора в певний момент часу. Важливо зазначити, що цей зв'язок не є обов'язковим, оскільки група може функціонувати і без призначеного куратора, хоча це не є бажаним з точки зору організації коректного навчально-виховного процесу.

У структурній ієрархії навчального закладу кафедра є підрозділом факультету, що визначає зв'язок між ними як «багато-до-одного». Один факультет може об'єднувати кілька кафедр різного профілю, які забезпечують підготовку фахівців за різними спеціальностями та спеціалізаціями. При цьому кожна кафедра може належати лише до одного факультету, що визначає її місце в організаційній структурі закладу. Цей зв'язок є обов'язковим, оскільки кожна кафедра повинна бути чітко закріплена за певним факультетом.

Розглядаючи адміністративну структуру кафедри, можна виділити зв'язок між кафедрою та академічним співробітником у контексті керівництва. Цей зв'язок відповідає моделі «багато-до-одного», оскільки один викладач теоретично може бути завідувачем кількох кафедр. При цьому кожна кафедра може мати лише одного завідувача в певний момент часу. Варто зазначити, що цей зв'язок не є обов'язковим, оскільки кафедра може функціонувати і без призначеного завідувача, хоча це не є нормальним станом.

Аналогічно до попереднього, зв'язок між факультетом та академічним співробітником у контексті керівництва також характеризується як «багато-до-одного». Один викладач теоретично може бути деканом кількох факультетів (хоча на практиці це малоймовірно). При цьому кожен факультет може мати лише одного декана в певний момент часу. Цей зв'язок не є обов'язковим, оскільки факультет може функціонувати і без призначеного декана, хоча це не є бажаним з точки зору ефективного управління.

Розглядаючи трудові відносини в навчальному закладі, можна визначити, що зв'язок між академічним співробітником та факультетом відповідає моделі «багато-до-одного». Один факультет може об'єднувати десятки або сотні викладачів різних спеціальностей та кваліфікацій. При цьому кожен викладач може бути закріплений лише за одним факультетом, що визначає його місце в організаційній структурі закладу. Цей зв'язок є обов'язковим, оскільки кожен академічний співробітник повинен бути чітко закріплений за певним визначеним факультетом.

Подібно до попереднього, зв'язок між академічним співробітником та кафедрою також характеризується як «багато-до-одного». Одна кафедра може об'єднувати кілька або кільканадцять викладачів різних спеціальностей та кваліфікацій. При цьому кожен викладач може бути закріплений лише за однією кафедрою, що визначає його місце в організаційній структурі закладу та напрям його педагогічної та наукової діяльності. Цей зв'язок є обов'язковим, оскільки кожен академічний співробітник повинен бути чітко закріплений за певною кафедрою для ефективно організації навчального процесу.

Аналізуючи організацію навчального процесу, можна визначити, що зв'язок між навчальною дисципліною та кафедрою відповідає моделі «багато-до-одного». Одна кафедра може забезпечувати викладання десятків різних дисциплін відповідно до навчальних планів різних спеціальностей. При цьому кожна дисципліна може бути закріплена лише за однією кафедрою, яка відповідає за її методичне забезпечення та викладання. Цей зв'язок обов'язковий, бо кожна дисципліна повинна чітко закріплюватись за кафедрою.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		40

Розглядаючи організацію викладання, можна виділити зв'язок між навчальною дисципліною та академічним співробітником у контексті відповідальності за її викладання. Цей зв'язок відповідає моделі «багато-до-одного», оскільки один викладач може бути відповідальним за кілька різних дисциплін, забезпечуючи їх методичне наповнення та викладання. При цьому кожна дисципліна може мати лише одного відповідального викладача в певний момент часу. Варто зазначити, що цей зв'язок не є обов'язковим, оскільки дисципліна може існувати і без призначеного відповідального викладача.

Аналізуючи процес оцінювання знань, можна визначити, що зв'язок між оцінкою та студентом відповідає моделі «багато-до-одного». Один студент протягом навчання отримує десятки або сотні різних оцінок з різних дисциплін, модулів, контрольних заходів. При цьому кожна конкретна оцінка може належати лише одному студенту, відображаючи рівень його знань з певної теми чи дисципліни. Цей зв'язок є обов'язковим, оскільки кожна оцінка повинна бути чітко пов'язана з конкретним студентом для правильного формування рейтингу.

Подібно до попереднього, зв'язок між оцінкою та навчальною дисципліною також характеризується як «багато-до-одного». З однієї дисципліни може бути виставлено сотні або тисячі оцінок різним студентам за різні види контролю протягом різних років. При цьому кожна конкретна оцінка може стосуватися лише однієї дисципліни, відображаючи рівень знань студента саме з цього предмету. Цей зв'язок є обов'язковим, оскільки кожна оцінка повинна бути чітко пов'язана з конкретною дисципліною.

Завершуючи аналіз процесу оцінювання, можна визначити, що зв'язок між оцінкою та академічним співробітником відповідає моделі «багато-до-одного». Один викладач протягом своєї педагогічної діяльності виставляє сотні або тисячі різних оцінок різним студентам з різних дисциплін.

При цьому кожна конкретна оцінка може бути виставлена лише одним викладачем, який проводив відповідний контрольний захід. Цей зв'язок є обов'язковим, оскільки кожна оцінка повинна бути чітко пов'язана з конкретним викладачем.

					КВРППЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		41

Усі виявлені сутності з детально описаними атрибутами та чітко окресленими типами зв'язків були систематизовані й узагальнені з метою формалізації структури даних. Отримані результати візуалізовано у вигляді ER-діаграми [27], спроектованої за спеціальною нотацією «Crow's Foot» [28] та представленої на рисунку 2.2.1.

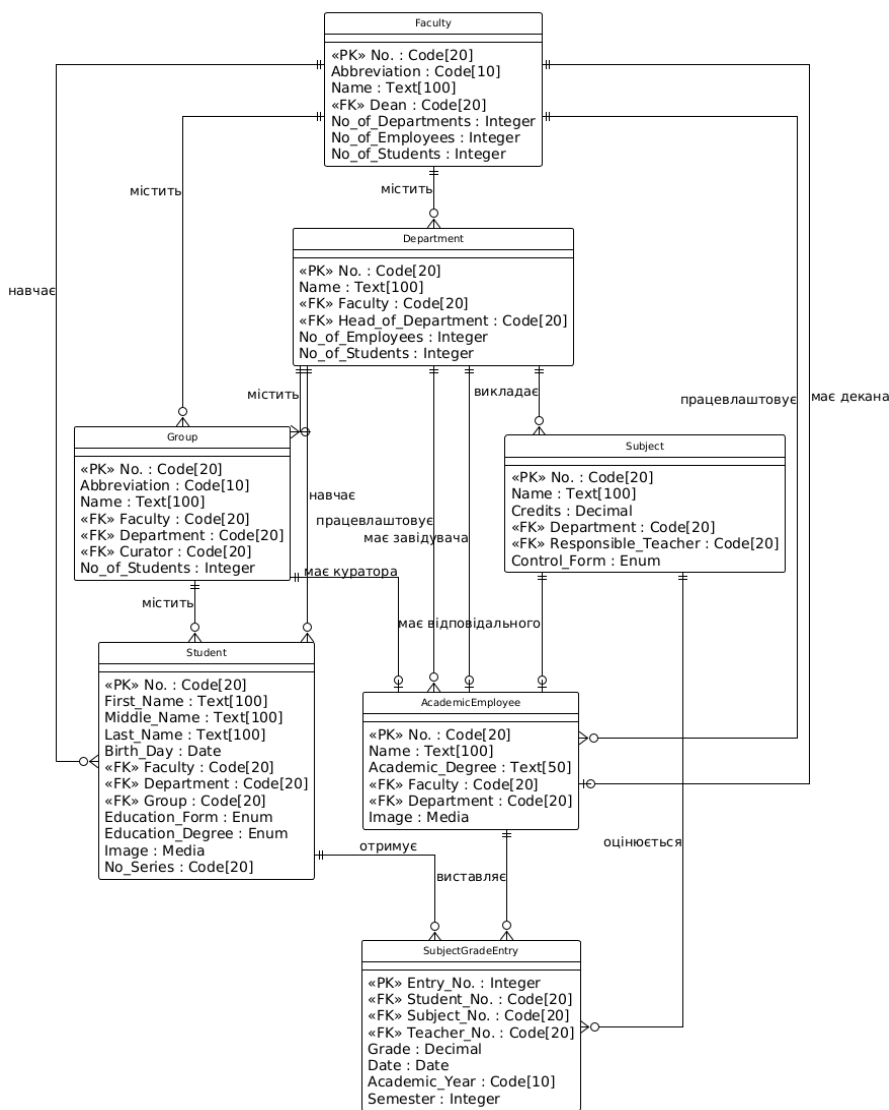


Рисунок 2.2.1 – ER-діаграма

У процесі проектування логічної моделі даних було здійснено перехід від концептуального представлення інформаційної структури до більш формалізованої схеми, придатної для реалізації в конкретній системі керування базами даних. На цьому етапі уточнено типи атрибутів, встановлено первинні та зовнішні ключі, нормалізовано таблиці з метою усунення надлишковості та забезпечення цілісності даних.

2.3 Детальне проектування програмних модулів

Процес детального проектування дозволяє не лише визначити структурні та поведінкові аспекти кожного модуля системи, але й забезпечує формалізацію взаємозв'язків між ними, що є необхідною передумовою для створення цілісної, інтегрованої системи з високим рівнем надійності та продуктивності.

У контексті проектування системи, програмний модуль – це логічно завершений, функціонально самостійний фрагмент програмного забезпечення, який виконує певну задачу або набір взаємопов'язаних задач і може використовуватися як окремий компонент у складі більшої програмної системи.

Для забезпечення гнучкості та масштабованості системи планування ресурсів навчального закладу доцільно використовувати модульний підхід. Це дозволить розділити функціональність системи на логічні компоненти, кожен з яких відповідатиме за певний аспект роботи навчального закладу.

Аналізуючи потреби навчального закладу, можна виділити два основні напрямки функціонування: управління структурою закладу та управління навчальним процесом

Відповідно до цього, пропонується розділити систему на два основні блоки модулів, кожен з яких міститиме підмодулі для вирішення конкретних специфічних завдань.

Модулі управління структурою, відповідатимуть за організаційну структуру навчального закладу та включатимуть в себе:

- модуль управління факультетами, який забезпечує функціональність для створення, редагування та видалення факультетів, призначення деканів, управління документацією факультетів;
- модуль управління кафедрами, який відповідає за управління кафедрами, призначення завідувачів кафедр, розподіл викладачів;
- модуль управління групами, який забезпечує функціональність для створення та управління академічними групами, формування структури груп, відстеження успішності груп.

					КВРППЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		43

Модулі управління навчальним процесом, які відповідатимуть за безпосередню організацію навчального процесу та включатимуть в себе:

- модуль управління студентами, який забезпечує функціональність для ведення особових справ студентів, відстеження їхньої успішності;
- модуль управління викладачами, який відповідає за управління даними про викладачів, їхнє навантаження, діяльність;
- модуль управління дисциплінами, який забезпечує функціональність для створення та управління навчальними дисциплінами, навчальними планами, програмами дисциплін;
- модуль управління оцінками, який відповідає за облік успішності студентів, формування відомостей, аналіз успішності.

Усі визначені модулі з детально описаними функціями та чітко окресленими взаємодіями були систематизовані та візуалізовані у формі діаграми модулів, що розташована на рисунку 2.3.1.

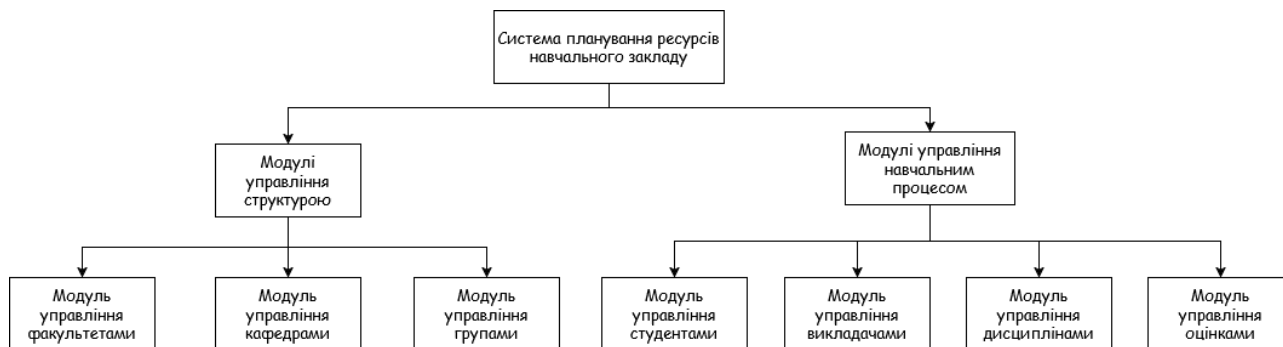


Рисунок 2.3.1 – Діаграма програмних модулів

Оскільки повна модель проектування формується на підґрунті системних вимог, ідентифікованих та специфікованих на етапі аналізу, для створення комплексного та всебічного представлення архітектури системи доцільно застосовувати інструментарій, що є невід’ємною складовою уніфікованого процесу розробки програмного забезпечення.

Саме до такого інструментарію належать UML-діаграми, які забезпечують стандартизований підхід до візуалізації програмних систем на різних рівнях абстракції, полегшуючи розуміння архітектури та логіки взаємодії компонентів.

UML-діаграми [29] представляють собою потужний механізм візуального моделювання, що використовується для формалізації, документування та специфікації програмних систем. Ці діаграми дозволяють абстрагуватися від деталей реалізації та зосередитися на концептуальних аспектах системи, що сприяє більш глибокому розумінню її структури та поведінки.

Для глибшого розуміння поведінки системи та взаємозв'язків між її елементами було вирішено спершу застосувати діаграми послідовності.

Діаграми послідовності [30] дозволяють візуалізувати взаємодію між об'єктами системи в хронологічному порядку, що є особливо важливим для розуміння складних бізнес-процесів навчального закладу. На відміну від статичних діаграм, які показують структуру системи, діаграми послідовності демонструють, як різні компоненти системи взаємодіють між собою.

Для забезпечення комплексного розуміння функціональності системи було проведено детальний аналіз архітектури кожного програмного модуля. У процесі дослідження було ідентифіковано ключові структурні елементи модулів, визначено їхні функціональні характеристики та встановлено логічні взаємозв'язки між ними. На основі систематизованої інформації про структуру програмних модулів було розроблено діаграми послідовності.

Модуль управління факультетами є одним із ключових компонентів системи, оскільки оперує факультетом, як основною структурною одиницею навчального закладу. Цей модуль забезпечуватиме функціональність для створення, редагування, видалення та перегляду інформації про факультети.

Структурними елементами модуля будуть:

- таблиця «Faculty», яка зберігатиме основну інформацію про вказані в ній факультети;
- сторінка «Faculty Card», яка забезпечувати інтерфейс для роботи із кожним окремим факультетом;
- сторінка «Faculties List», яка відображатиме список усіх факультетів;
- сторінка «Faculty Departments», яка відображатиме список кафедр, що належать до конкретного факультету.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		45

Діаграма послідовності для процесу створення нового факультету представлена на рисунку 2.3.2.

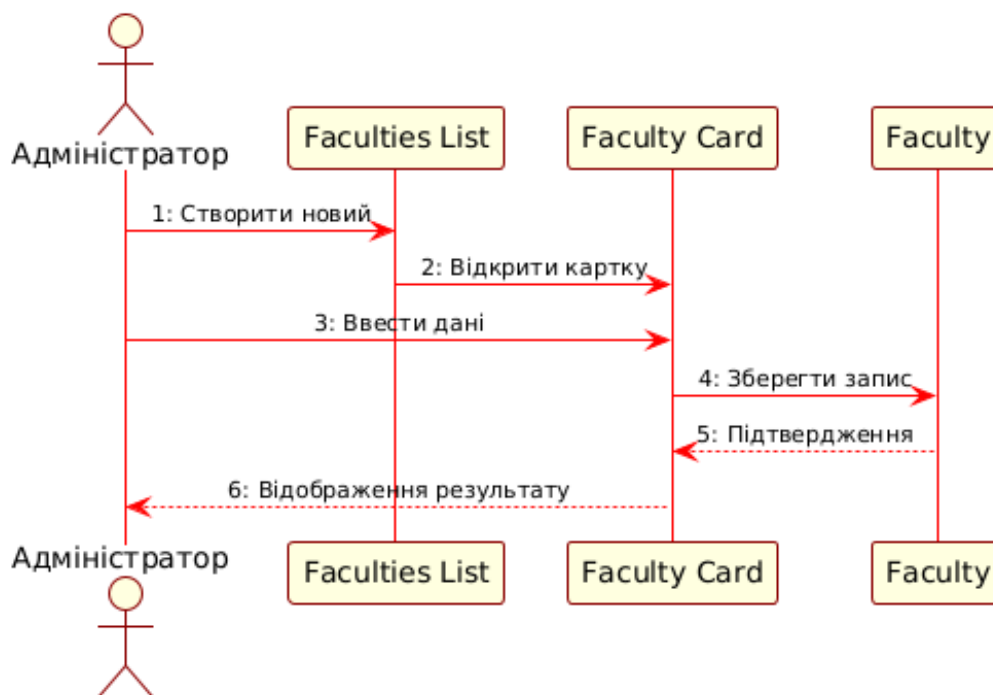


Рисунок 2.3.2 – Діаграма послідовності "Створення нового факультету"

Модуль управління кафедрами забезпечує функціональність для роботи з кафедрами як структурними підрозділами факультетів. Він тісно інтегрований із модулем управління факультетами та дозволяє створювати, редагувати, видаляти й переглядати інформацію про кафедри. Окрім базових операцій, модуль також може включати можливості призначення завідувачів кафедр, відстеження кадрового складу та взаємодії з іншими підрозділами. Цей модуль використовується для ефективного керування кафедрами як ключовими навчально-науковими одиницями факультетів.

Структурними елементами модуля будуть:

- таблиця «Department», зберігатиме основну інформацію про кафедри;
- сторінка «Department Card», яка забезпечуватиме інтерфейс для роботи з окремою кафедрою;
- сторінка «Departments List», яка відобразить список усіх кафедр;
- сторінка «Department Groups», яка відобразить список груп, що належать до конкретної кафедри.

Діаграма послідовності для процесу створення нової кафедри представлена на рисунку 2.3.3.

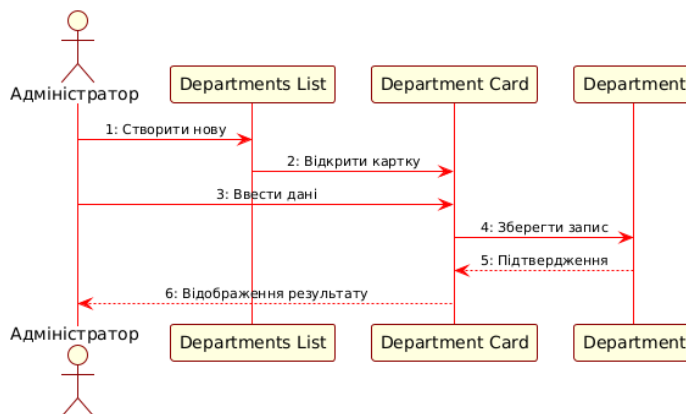


Рисунок 2.3.3 – Діаграма послідовності "Створення нової кафедри"

Модуль управління групами відповідає за керування навчальними групами студентів. Цей модуль забезпечує функціональність для створення, редагування, видалення та перегляду інформації про академічні групи.

Структурними елементами модуля будуть:

- таблиця «Group», яка зберігатиме основну інформацію про групи;
- сторінка «Group Card», яка забезпечуватиме інтерфейс для роботи з кожною окремою групою;
- сторінка «Groups List», яка відображатиме список усіх груп;
- сторінка «Group Students», яка відображатиме список студентів, що належать до конкретної групи.

Діаграма послідовності для процесу створення нової групи представлена на рисунку 2.3.4.

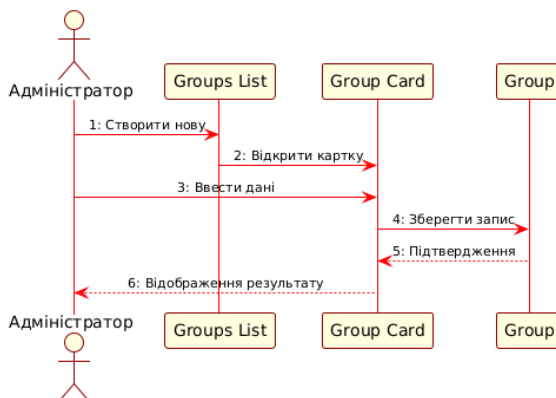


Рисунок 2.3.4 – Діаграма послідовності "Створення нової групи"

Модуль управління студентами забезпечує функціональність для керування інформацією про студентів. Цей модуль дозволяє створювати, редагувати, видаляти та переглядати інформацію про студентів.

Структурними елементами модуля будуть:

- таблиця «Student», яка зберігатиме основну інформацію про студентів;
- сторінка «Student Card», яка забезпечуватиме інтерфейс для роботи з кожним окремим студентом;
- сторінка «Students List», яка відобразить список усіх студентів;
- сторінка «Subject Grade Entry», яка відобразить інформацію про успішність студента.

Діаграма послідовності для процесу створення нового студента представлена на рисунку 2.3.5.

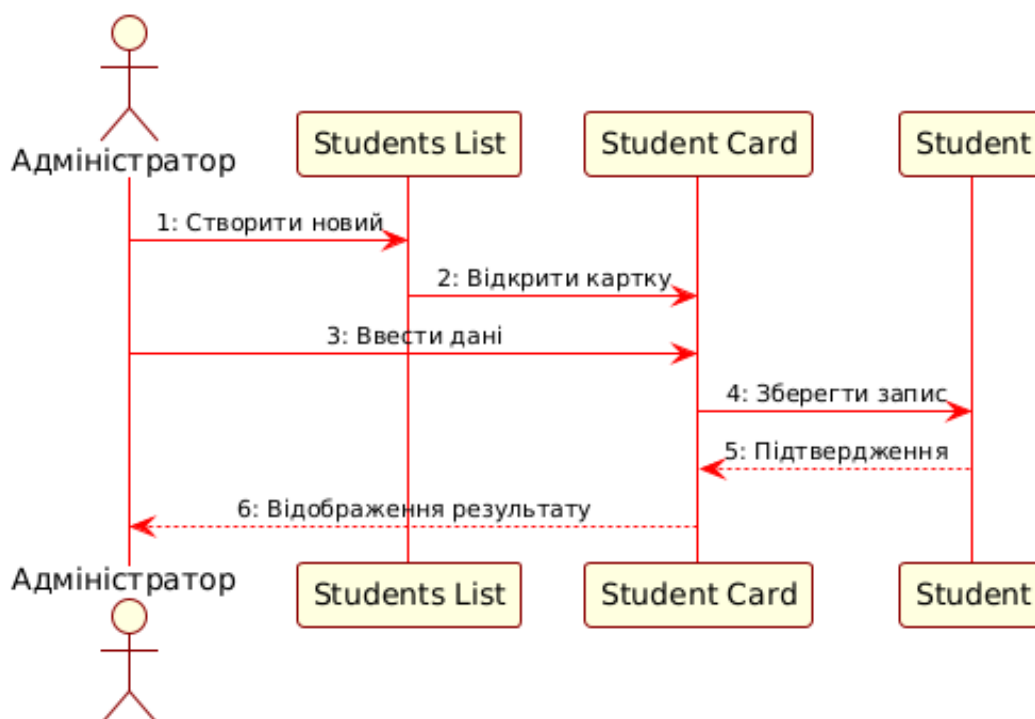


Рисунок 2.3.5 – Діаграма послідовності "Створення нового студента"

Модуль управління викладачами забезпечує функціональність для керування інформацією про викладацький склад. Цей модуль дозволяє створювати, редагувати, видаляти та переглядати інформацію про викладачів. А також працювати із навантаженням.

Структурними елементами модуля будуть:

- таблиця «Teacher», яка зберігатиме основну інформацію про усіх наявних викладачів;
- сторінка «Teacher Card», яка забезпечуватиме інтерфейс для роботи з кожним окремим викладачем;
- сторінка «Teachers List», яка відобразатиме список усіх викладачів;
- сторінка «Teacher Subject», яка відобразатиме інформацію про навантаження викладача.

Діаграма послідовності для процесу створення нового викладача представлена на рисунку 2.3.6.

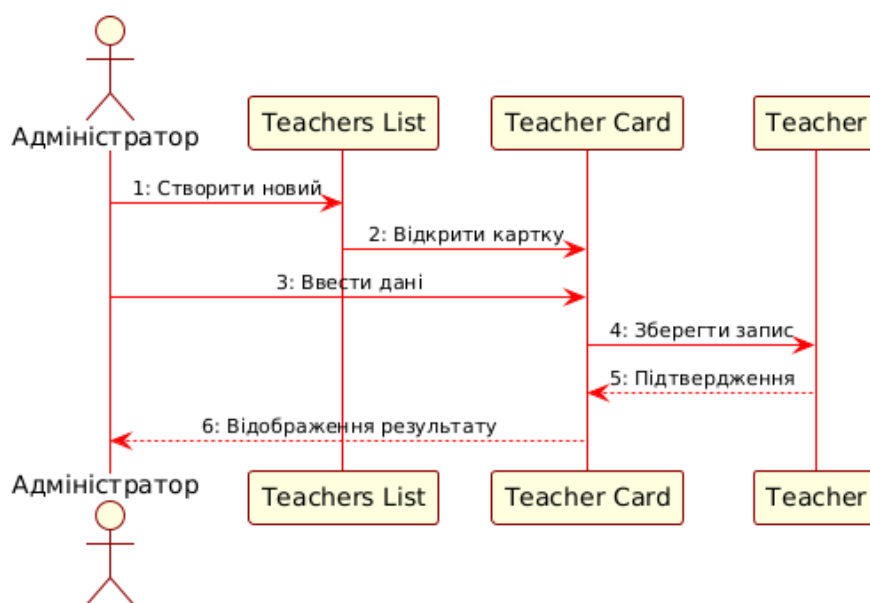


Рисунок 2.3.6 – Діаграма послідовності «Створення нового викладача»

Таким чином, у ході виконання детального проектування програмних модулів було розроблено архітектуру системи, що складається з взаємопов'язаних компонентів, кожен з яких відповідає за певний функціональний аспект програмного забезпечення. Розроблені діаграми послідовності наочно демонструють взаємодію між компонентами системи та послідовність виконання операцій, що дозволяє краще зрозуміти логіку роботи програмного забезпечення та спрощує подальшу реалізацію. Результати детального проектування будуть використані на етапі програмної реалізації системи та дозволять забезпечити відповідність розробленого програмного забезпечення поставленим вимогам та очікуванням користувачів.

2.4 Висновки проєктування системи планування ресурсів

У даному розділі кваліфікаційної роботи було здійснено комплексне проєктування програмної системи, яке охопило два ключові етапи проєктування: проєктування архітектури та структури системи, а також проєктування логічної моделі наявної бази даних.

На першому етапі, при проєктуванні архітектури та структури, було проведено детальний аналіз моделей розгортання. Після порівняння переваг та недоліків локальної та хмарної моделей, було обґрунтовано вибір хмарної моделі. Ключовими факторами на користь цього рішення стали низькі початкові інвестиції в інфраструктуру та висока масштабованість, що є важливим для освітніх закладів. Далі, розглядаючи архітектурні підходи в контексті хмари, було проаналізовано Single-Tenant та Multi-Tenant архітектури. В результаті порівняльного аналізу перевага була надана Single-Tenant архітектурі через забезпечення вищого рівня безпеки та конфіденційності даних завдяки повній фізичній ізоляції, а також надання повного контролю над інфраструктурою.

На другому етапі було розроблено логічну модель бази даних. Цей процес включав уніфікацію та стандартизацію її компонентів. Було ідентифіковано ключові сутності предметної області.

Для кожної сутності було визначено набір релевантних атрибутів, що були представлені у вигляді таблиць з репрезентативними даними. Після цього було проведено детальний аналіз та встановлено типи взаємозв'язків між сутностями. Результати цього аналізу були формалізовані та представлені за допомогою ER-діаграми.

Завершальним кроком стало уточнення типів атрибутів, визначення первинних та зовнішніх ключів, а також нормалізація таблиць.

Таким чином, було детально обґрунтовано вибір архітектурних рішень, визначено структуру програмної системи та розроблено логічну модель бази даних. Ці результати створили міцну теоретичну та практичну основу для подальшої реалізації та впровадження системи планування ресурсів.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						50
Зм.	Арк	№ докум.	Підпис	Дата		

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ ПЛАНУВАННЯ РЕСУРСІВ

3.1 Програмна реалізація модулів

Розробка програмної системи є доволі комплексним процесом, що вимагає створення взаємопов'язаних модулів, які забезпечують повноцінне функціонування системи. Кожен модуль розробляється з використанням мови програмування AL яка є основною мовою для розширення функціональності Microsoft Dynamics 365.

Архітектура системи базується на об'єктно-орієнтованому підході, де основними функціональними блоками є об'єкти різних типів: таблиці, які визначають структуру даних; сторінки різних типів, що забезпечують інтерфейс користувача, звіти для аналізу. Також врахуємо інші об'єкти, такі як CodeUnit, XMLPort, Query, що забезпечують додаткову функціональність. Такий підхід сприяє модульності, повторному використанню коду та спрощує підтримку й масштабування системи.

Система розгортається за моделлю SaaS з архітектурним підходом «Single Tenant», що забезпечує ізоляцію даних кожного навчального закладу та можливість індивідуальної конфігурації системи відповідно до специфічних потреб закладу. Відповідно до архітектури Microsoft Dynamics 365 та мови AL, розробку було потрібно починати із створення таблиць. Процес створення таблиці використанням мови AL завжди відповідає певному патерну, який забезпечує структурованість та функціональність об'єкта даних. На початковому етапі встановлюються загальні властивості таблиці, які визначають її поведінку в системі:

```
table 50101 "UNV Faculty"
{
    Caption = 'Faculty';
    DataCaptionFields = "No.", Name;
    DataClassification = CustomerContent;
    DrillDownPageID = "UNV Faculties List";
    LookupPageID = "UNV Faculties List";
}
```

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						51
Зм.	Арк	№ докум.	Підпис	Дата		

Властивість «Caption» визначає заголовок таблиці, який відобразатиметься в інтерфейсі користувача. Заголовок буде демонструватися в різних частинах системи, де використовується ця таблиця, наприклад, у заголовках сторінок, списках та звітах.

Властивість «DataCaptionField» визначає, які поля будуть використовуватися для формування підпису запису при його відображенні. У даному випадку, коли користувач переглядає запис факультету, система відобразатиме комбінацію значень полів коду факультету та назви факультету як підпис запису. Це допомагає користувачам швидко ідентифікувати конкретний запис.

Властивість визначає «DataClassification» класифікацію даних з точки зору конфіденційності та відповідності вимогам GDPR (Загального регламенту про захист даних).

Значення «CustomerContent» вказує, що дані в цій таблиці належать до категорії «Вміст клієнта», що має певні наслідки для обробки, зберігання та захисту цих даних відповідно до політик безпеки та конфіденційності. Дані класифікуються як такі, що належать і можуть містити інформацію, яка потребує захисту відповідно до політик безпеки організації.

Властивість «DrillDownPageID» визначає сторінку, яка буде відкриватися при виконанні операції деталізації на записі факультету. Коли користувач виконує деталізацію (наприклад, подвійний клік на полі, що містить посилання на факультет), система відкриє відповідну сторінку.

Властивість «LookupPageID» визначає сторінку, яка буде використовуватися для пошуку та вибору факультету в полях, що мають відношення до цієї таблиці. Коли користувач натискає кнопку пошуку у полі, що має відношення до факультету, система відкриє відповідні сторінку.

Наступним кроком є визначення структури полів таблиці в блоці «fields». Цей етап є критично важливим, оскільки він визначає, які дані будуть зберігатися в таблиці та як вони будуть структуровані. Блок полів містить визначення всіх полів таблиці.

Семантика таблиці відповідно повністю відповідає сформованій раніше моделі даних:

```

field(1; "No."; Code[20])
{
    Caption = 'Faculty Code';
}

field(2; Abbreviation; Code[10])
{
    Caption = 'Abbreviation';
    OptimizeForTextSearch = true;
}

field(3; Name; Text[100])
{
    Caption = 'Name';
    OptimizeForTextSearch = true;
}

field(4; Dean; Code[20])
{
    Caption = 'Dean';
    TableRelation = "UNV Academic Employee";

    trigger OnValidate()
    begin
        this.CalcFields("Dean Name");
    end;
}

field(5; "Dean Name"; Text[100])
{
    Caption = 'Dean Name';
    FieldClass = FlowField;
    CalcFormula = lookup
        ("UNV Academic Employee".Name where ("No." =
        field(Dean)));
    Editable = false;
}

field(7; "No. of Departments"; Integer)
{
    Caption = 'No. of Employee';
    FieldClass = FlowField;
    CalcFormula = count
        ("UNV Department" where(Faculty = field("No.")));
    Editable = false;
}

field(8; "No. of Employees"; Integer)
{
    Caption = 'No. of Employee';
    FieldClass = FlowField;
    CalcFormula = count
        ("UNV Academic Employee" where(Faculty = field("No.")));
    Editable = false;
}

field(9; "No. of Students"; Integer)
{
    Caption = 'No. of Students';
    FieldClass = FlowField;
    CalcFormula = count
        ("UNV Student" where("Faculty" = field("No.")));
    Editable = false;
}

```

					КВРІПЗ. 2101075.01.06.ПЗ	<i>Арк.</i>
<i>Зм.</i>	<i>Арк</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		53

Однак, просте визначення полів з їх типами даних недостатньо для створення функціональної системи. Саме тому кожне поле може містити додатковий набір властивостей, які розширюють його функціональність, визначають поведінку взаємодію з іншими об'єктами системи.

Властивість «FieldClass» визначає тип поля з точки зору його поведінки в системі. Існує три основні типи:

- Normal – звичайне поле, яке зберігає дані безпосередньо в таблиці;
- FlowField – обчислюване поле, значення якого не зберігається в таблиці, а обчислюється на основі даних з інших таблиць за допомогою певної визначеної формули;
- FlowFilter – поле-фільтр, яке використовується для фільтрації даних.

Властивість «CalcFormula» визначає формулу обчислення для полів типу «FlowField». Ця формула може використовувати різні агрегатні функції (sum, count, average, min, max) та умови фільтрації для отримання даних з інших пов'язаних таблиць.

Властивість «Editable» визначає, чи можна редагувати поле в інтерфейсі користувача. Якщо встановлено значення «false», поле буде доступне лише для читання. У нашому випадку, ця властивість використовується для визначення обчислюваних полів.

Властивість «NotBlank» визначає, чи може поле бути порожнім. Якщо встановлено значення «true», система не дозволить зберегти запис з порожнім значенням у цьому полі. Ця властивість є одним із механізмів забезпечення цілісності даних.

Властивість «TableRelation» визначає зв'язок поля з іншою таблицею, створюючи відношення "зовнішній ключ". Це дозволяє забезпечити цілісність даних та реалізувати функціональність пошуку. Ця властивість може включати умови фільтрації для обмеження доступних значень та використовувати поля поточного запису для динамічної фільтрації.

Властивість «OptimizeForTextSearch» оптимізує поле для текстового пошуку, що покращує продуктивність при виконанні пошукових операцій. Ця

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						54
Зм.	Арк	№ докум.	Підпис	Дата		

властивість особливо корисна для полів, які часто використовуються в пошукових запитах.

Властивість «ExtendedDatatype» визначає розширений тип даних для спеціальних форматів, таких як електронна пошта, телефонний номер, URL тощо. Це дозволяє системі застосовувати спеціальні правила валідації та форматування для цих типів даних.

Після визначення полів та ключів таблиці, наступним важливим кроком є визначення тригерів таблиці. Тригери – це спеціальні процедури, які автоматично виконуються при настанні певних подій, пов'язаних з таблицею. Вони дозволяють реалізувати складну бізнес-логіку, забезпечити цілісність даних та автоматизувати різноманітні процеси.

У середовищі AL існує кілька типів тригерів таблиці, кожен з яких виконується в певний момент життєвого циклу запису.

Тригер «OnInsert» виконується при генерації нового запису в таблицю, перед його фактичним збереженням у базі даних. У наявній програмній системі тригер використовувався для:

- ініціалізації полів значеннями за замовчуванням;
- генерації унікальних ідентифікаторів;
- перевірки бізнес-правил перед вставкою;
- створення пов'язаних записів в інших таблицях.

Нижче наведений код реалізації тригера для таблиці «Факультет»:

```
trigger OnInsert ()
var
    UniversitySetup: Record "UNV University Setup";
    NoSeriesManagement: Codeunit "NoSeries Management";
begin
    if "No." = '' then begin
        FacultySetup.Get ();
        FacultySetup.TestField("Faculty Nos.");
        "No." := NoSeriesManagement.GetNextNo(FacultySetup."Faculty Nos.",
WorkDate(), true);
        end;

        "Creation Date" := WorkDate();
        "Created By" := UserId();
        Status := Status::Active;
    end;
```

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						55
Зм.	Арк	№ докум.	Підпис	Дата		

Тригер «OnModify» виконується при модифікації існуючого запису, перед збереженням змін у базі даних. У наявній програмній системі тригер використовувався для:

- перевірки бізнес-правил перед оновленням;
- відстеження змін у записі;
- оновлення пов'язаних записів в інших таблицях;
- ведення журналу змін.

Нижче наведений код реалізації тригера для таблиці «Факультет»:

```
trigger OnModify()
var
  FacultyLog: Record "UNV Faculty Log";
begin
  if (xRec.Name <> Name) or (xRec.Dean <> Dean) then begin
    FacultyLog.Init();
    FacultyLog."Faculty Code" := "No.";
    FacultyLog."Date Time" := CurrentDateTime();
    FacultyLog."User ID" := UserId();

    if xRec.Name <> Name then begin
      FacultyLog."Field No." := FieldNo(Name);
      FacultyLog."Old Value" := xRec.Name;
      FacultyLog."New Value" := Name;
      FacultyLog.Insert();
    end;

    if xRec.Dean <> Dean then begin
      FacultyLog."Field No." := FieldNo(Dean);
      FacultyLog."Old Value" := xRec.Dean;
      FacultyLog."New Value" := Dean;
      FacultyLog.Insert();
    end;
  end;

  "Last Modified Date" := CurrentDateTime();
  "Last Modified By" := UserId();
end;
```

Тригер «onDelete» виконується при видаленні запису з таблиці, перед його фактичним видаленням з бази даних. У наявній програмній системі тригер використовувався для:

- перевірки можливості видалення запису;
- видалення пов'язаних записів в інших таблицях;
- ведення журналу видалення;
- запобігання випадковому видаленню важливих даних.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						56
Зм.	Арк	№ докум.	Підпис	Дата		

Нижче наведений код реалізації тригера для таблиці «Факультет»:

```
trigger OnDelete ()
var
    Department: Record "UNV Department";
    Student: Record "UNV Student";
    CannotDeleteErr: Label 'Faculty %1 cannot be deleted, because it
contains %2 departments and %3 students.';
begin
    Department.SetRange(Faculty, "No.");
    if not Department.IsEmpty() then begin
        Student.SetRange("Faculty", "No.");
        if not Student.IsEmpty() then
            Error(CannotDeleteErr, "No.", Department.Count(),
Student.Count());
        end;
    end;
end;
```

Також важливою частиною роботи з даними в є автоматичне присвоєння унікальних номерів для нових записів, і для цього використовується механізм присвоєння «No. Series». Це підхід дозволяє системі автоматично генерувати послідовні та унікальні ідентифікатори для документів та будь-яких інших сутностей, що потребують нумерації.

Спочатку в спеціальній таблиці налаштувань визначається, яка саме серія номерів буде використовуватися для конкретного типу записів. Коли створюється новий запис, наприклад, новий факультет, система перевіряє, чи поле «No.» порожнє.

Якщо так, це означає, що запис новий і йому потрібно присвоїти номер. Система звертається до таблиці налаштувань, щоб отримати код серії номерів, призначений для конкретної сутності. Після цього викликається стандартний системний об'єкт «NoSeriesManagement» з функцією «GetNextNo». В результаті, функція повертає наступний вільний номер з указаної серії, який і присвоюється полю «No.» нового запису.

Застосовуючи аналогічний структурований підхід, було розроблено та реалізовано табличні об'єкти для всіх інших визначених у системі сутностей даних. Це забезпечило уніфіковане представлення інформації, цілісність даних та дозволило перейти до розробки інших структурних елементів функціональних модулів системи, таких як: сторінки для взаємодії з користувачем, звіти та основна бізнес-логіка модулів.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						57
Зм.	Арк	№ докум.	Підпис	Дата		

Після завершення етапу структурування даних та визначення табличної моделі, ключовим завданням у процесі розробки програмної системи стало проєктування та реалізація користувацького інтерфейсу. Саме через інтерфейс відбувається безпосередня взаємодія кінцевого користувача з системою, і від його якості значною мірою залежить ефективність роботи та загальне сприйняття розробленого рішення.

У середовищі AL для створення елементів інтерфейсу використовуються об'єкти типу «Page». У цьому випадку розробка користувацького інтерфейсу виходить за межі простого візуального представлення даних. Процес розробки включає: глибокого розуміння потреб користувачів, аналіз бізнес-процесів, які будуть підтримуватися системою, та врахування принципів ергономіки для забезпечення інтуїтивності та зручності використання системи.

Для створення користувацького інтерфейсу, який би не лише відповідав функціональним вимогам, але й забезпечував комфортну та продуктивну роботу, у середовищі AL існує гнучкий інструментарій у вигляді різноманітних типів сторінок.

Кожен тип сторінки має своє специфічне призначення, дозволяючи оптимально структурувати інформацію та взаємодію для різних сценаріїв. У рамках розробки нашої програмної системи передбачається використання декількох ключових типів сторінок, що забезпечить цілісність та інтуїтивність користувацького досвіду.

Для перегляду деталей окремих сутностей, чи то картка факультету, чи інформація про студента, будуть застосовані сторінки типу «Card». Вони надають детальний огляд всіх атрибутів об'єкта та інструменти для їх редагування. Коли ж виникає потреба оглянути та оперувати масивом однорідних даних, наприклад, переліком кафедр університету або списком навчальних дисциплін, у такому випадку у системі використовуватимуться сторінки типу «List». Їх функціонал дозволяє ефективно здійснювати пошук, сортування та фільтрацію, перетворюючи великі обсяги даних на керовані набори інформації.

Особливе місце в архітектурі інтерфейсу займає «RoleCenter». Це, по суті, персоналізований командний пункт для кожного користувача, що збирає найважливішу інформацію, актуальні завдання та швидкі посилання, специфічні для кожного користувача.

Для додавання на сторінки більшої кількості контекстної інформації та підвищення їхньої модульності, система буде використовувати сторінки типу «CardPart» та «ListPart». «CardPart» дозволить лаконічно відобразити ключові дані з пов'язаної картки, тоді як «ListPart» надасть можливість переглядати списки залежних записів. Нарешті, для акцентування уваги на критично важливих показниках або оперативних повідомленнях, особливо на робочих центрах, буде використано сторінку типу «HeadlinePart», що динамічно відображає ключові текстові дані у форматі помітних заголовків.

Першочергову увагу було приділено створенню основних сторінок для взаємодії з ключовими сутностями даних. Як правило, розробка у середовищі AL розпочинається починається зі спискових та карткових сторінок. Розглянемо процес розробки інтерфейсу користувача на прикладі реалізації структурних елементів модуля управління студентами.

Першочергова увага приділяється визначенню властивостей сторінки, які становлять основу її функціональності та поведінки. Властивості сторінки визначають такі критичні аспекти як тип сторінки, заголовок, джерело даних та функціональну область застосування. Правильний вибір властивостей дозволяє забезпечити інтеграцію сторінки в загальну архітектуру системи та визначити її доступність для користувачів у різних контекстах роботи з системою.

```
page 50100 "UNV Student Card"
{
    ApplicationArea = All;
    Caption = 'Student Card';
    PageType = Card;
    UsageCategory = None;
    SourceTable = "UNV Student";
}
```

Після визначення базових властивостей було проведено проектування полів, які забезпечують безпосередню взаємодію користувача з даними.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						59
Зм.	Арк	№ докум.	Підпис	Дата		

Поля на сторінці карточки зазвичай організуються в логічні групи для полегшення сприйняття інформації. Кожне поле пов'язується з відповідним полем таблиці даних через посилання на запис, що дозволяє синхронізувати дані між користувацьким інтерфейсом та базою даних. Для підвищення інформативності та зручності використання кожне поле супроводжується підказками та може містити додаткову логіку валідації через тригери, що забезпечує цілісність даних та інтерактивність інтерфейсу.

Нижче наведено код визначених полів на картці сутності «Студент»:

```
field("No."; Rec."No.")
{
    ApplicationArea = All;
    Tooltip = 'Specifies the value of the No. field.';
}
field("First Name"; Rec."First Name")
{
    ApplicationArea = All;
    Tooltip = 'Specifies the value of the First Name field.';
}
field("Middle Name"; Rec."Middle Name")
{
    ApplicationArea = All;
    Tooltip = 'Specifies the value of the Middle Name field.';
}
field("Last Name"; Rec."Last Name")
{
    ApplicationArea = All;
    Tooltip = 'Specifies the value of the Last Name field.';
}
field("Birth Day"; Rec."Birth Day")
{
    ApplicationArea = All;
    Tooltip = 'Specifies the value of the Birth Day field.';
}
field("Education Form"; Rec."Education Form")
{
    ApplicationArea = All;
    Tooltip = 'Specifies the value of the Education Form field.';
}
field("Education Degree"; Rec."Education Degree")
{
    ApplicationArea = All;
    Tooltip = 'Specifies the value of the Education Degree field.';
}
```

Також важливим аспектом проектування інтерфейсу є включення додаткових інформаційних панелей, які дозволяють користувачу бачити контекстну інформацію без необхідності переходу між різними сторінками. Це підвищує ефективність роботи та забезпечує комплексне представлення даних.

					КВРПЗ. 2101075.01.06.ПЗ	<i>Арк.</i>
<i>Зм.</i>	<i>Арк</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		60

У нашому випадку з картою студента такі панелі містять фотографію та прикріплені документи.

Нижче наведено код для визначення інформаційних панелей:

```
part(StudentPicture; "UNV Student Image Factbox")
{
    ApplicationArea = All;
    Caption = 'Photo';
    SubPageLink = "No." = field("No.");
}

part(DocumentsFactBox; "Document Attachment Factbox")
{
    ApplicationArea = All;
    Caption = 'Documents';
    SubPageLink = "Table ID" = const(Database::"UNV Student"),
        "No." = field("No.");
}
```

Також усі поля на картках зазвичай групуються за призначенням або типом інформацію. Для картки студента були додатково виділені групи полів, які містять інформацію про факультет, кафедру та групу, до якої відноситься конкретний студент. Загалом основний вигляд картки студента наявний на рисунку 3.1.1.

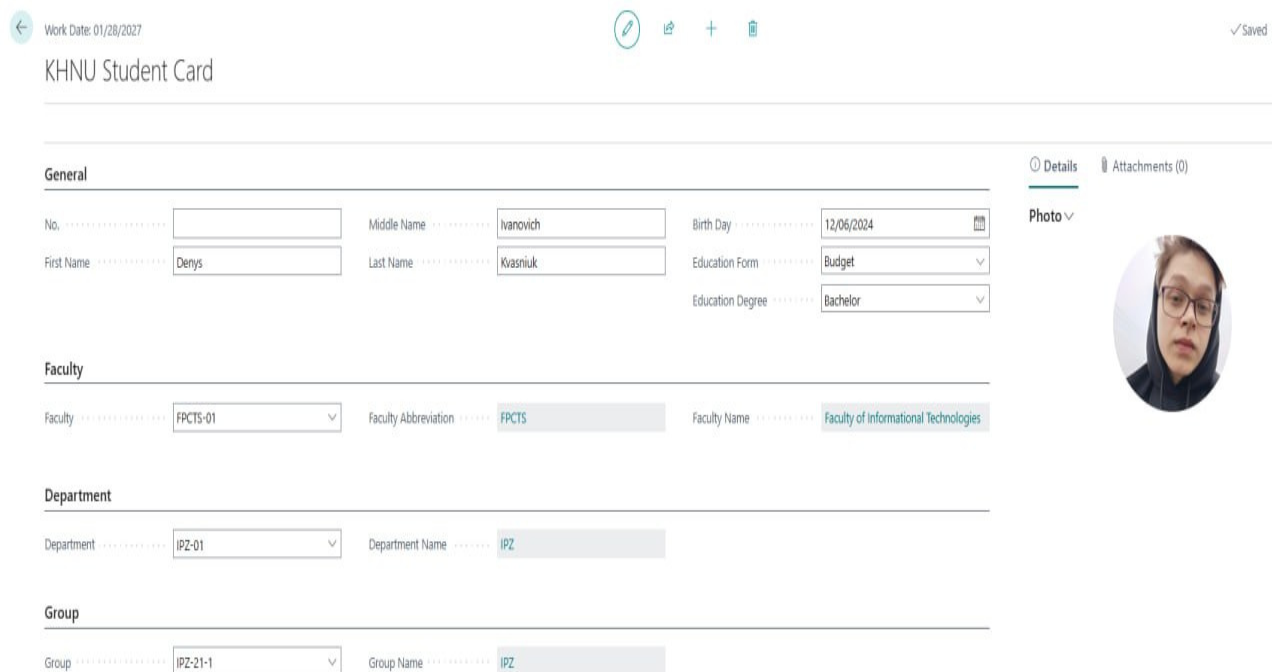


Рисунок 3.1.1 – Інтерфейс картки студента

					КВРІПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		61

Далі була реалізована спискова сторінка для перегляду усіх наявних студентів. Також була реалізована інформаційна панель із фотографією студента, в області навігації на списковій сторінці були реалізовані навігаційні кнопки для відкриття картки студента та картки групи. Загальний вигляд спискової сторінки для студентів наявний на рисунку 3.1.2.

Нижче наявний код реалізації навігаційних кнопок:

```

action (OpenCard)
{
    ApplicationArea = All;
    Caption = 'Student Card';
    Image = Card;
    RunObject = Page "KHNU Student Card";
    RunPageLink = "No." = field("No.");
    ToolTip = 'Executes the Student Card action.';
}
action (OpenGroup)
{
    ApplicationArea = All;
    Caption = 'Group Card';
    Image = Card;
    RunObject = Page "KHNU Group Card";
    RunPageLink = "No." = field(Group);
    ToolTip = 'Executes the Group Card action.';
}

```

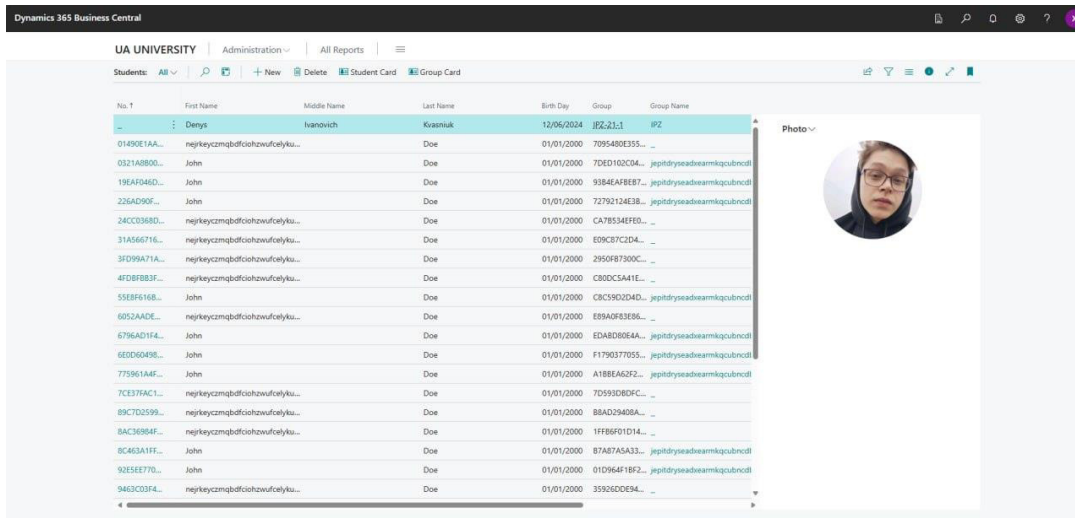


Рисунок 3.1.2 – Інтерфейс спискової сторінки студентів

За аналогічним підходом були реалізовані як карткові, так і лістові сторінки для решти визначених сутностей у проєкті. Для кожної з них були створені відповідні сторінки, що забезпечують як перегляд загального списку записів, так і детальну роботу з окремими записами.

3.2 Розгортання системи планування ресурсів

Розгортання програмної системи є критично важливим етапом життєвого циклу розробки програмного забезпечення, що полягає у підготовці та введенні системи в експлуатацію для кінцевих користувачів. Цей процес включає налаштування інфраструктури, інсталяцію програмних компонентів, конфігурацію параметрів та забезпечення доступності системи. Від успішності розгортання залежить стабільність роботи, продуктивність та безпека програмного продукту, а також загальне задоволення користувачів.

У контексті кваліфікаційної роботи, за результатами проектування архітектури програмної системи було прийнято архітектурне рішення провести розгортання за моделлю «Software as a Service» із застосуванням архітектурного підходу «Single Tenant».

Процес створення та налаштування ізолюваного середовища для програмної системи на базі Microsoft Dynamics 365 зазвичай розпочинається в середовищі Microsoft CDX.

Microsoft Customer Digital Experience (CDX) [31] – це платформа, яка дозволяє створювати демонстраційні ізолювані середовища, завантажувати покрокові сценарії демонстрацій та проводити інтерактивні сесії в повнофункціональних середовищах Microsoft 365. Платформа надає можливість отримати практичний досвід роботи з продуктами Microsoft 365, не впливаючи на робочі середовища та не вимагаючи додаткових фінансових витрат.

Для розгортання розробленої програмної системи було створено ізолюване середовище через платформу Microsoft CDX. Під час цього процесу було визначено основні параметри, такі як пакет вмісту «Dynamics 365 Business Central», географічне розташування та термін дії. Також було згенеровано необхідні облікові дані для адміністратора та користувачів системи, які дозволять оперувати центром адміністрування для програмної системи. Основна конфігурація середовища наявна на рисунку 3.2.1.

					КВРІПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		63

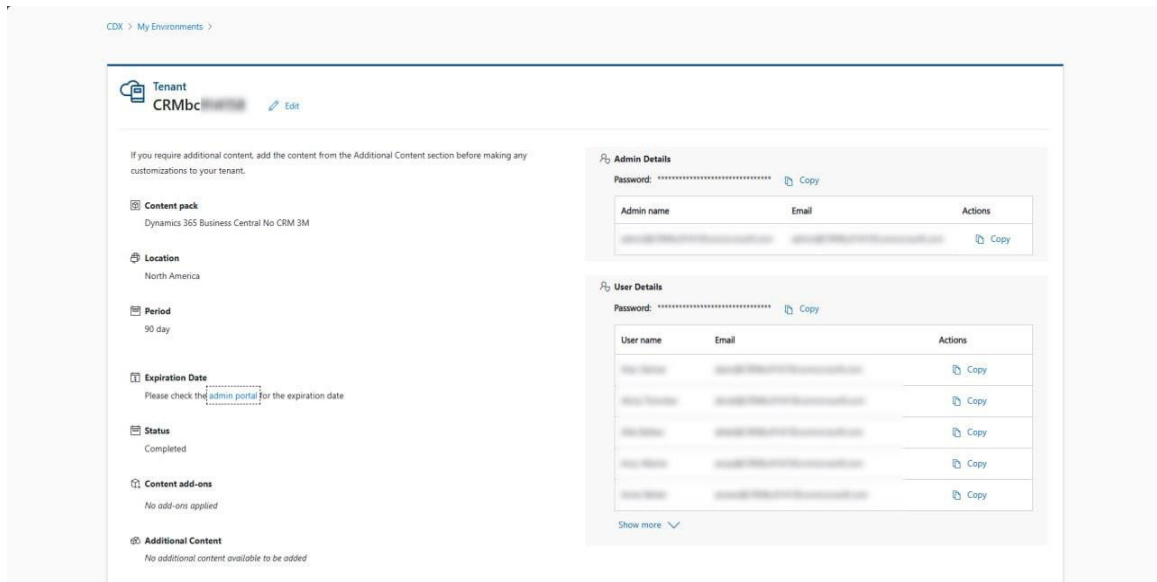


Рисунок 3.2.1 – Конфігурація розгорнутого ізольованого середовища

Після успішного створення ізольованого середовища та отримання необхідних облікових даних, наступним кроком було налаштування безпосередньо робочого простору Business Central. Для цього, використовуючи згенеровані дані, було здійснено вхід до адміністративного центру Dynamics 365 Business Central. В адміністративному центрі було ініційовано процес створення нового середовища для розгортання програмної системи. Для розгортання використовувалась остання версія Business Central v.26.0 з типом середовища «Production». Основна конфігурація середовища наявна на рисунку 3.2.2.

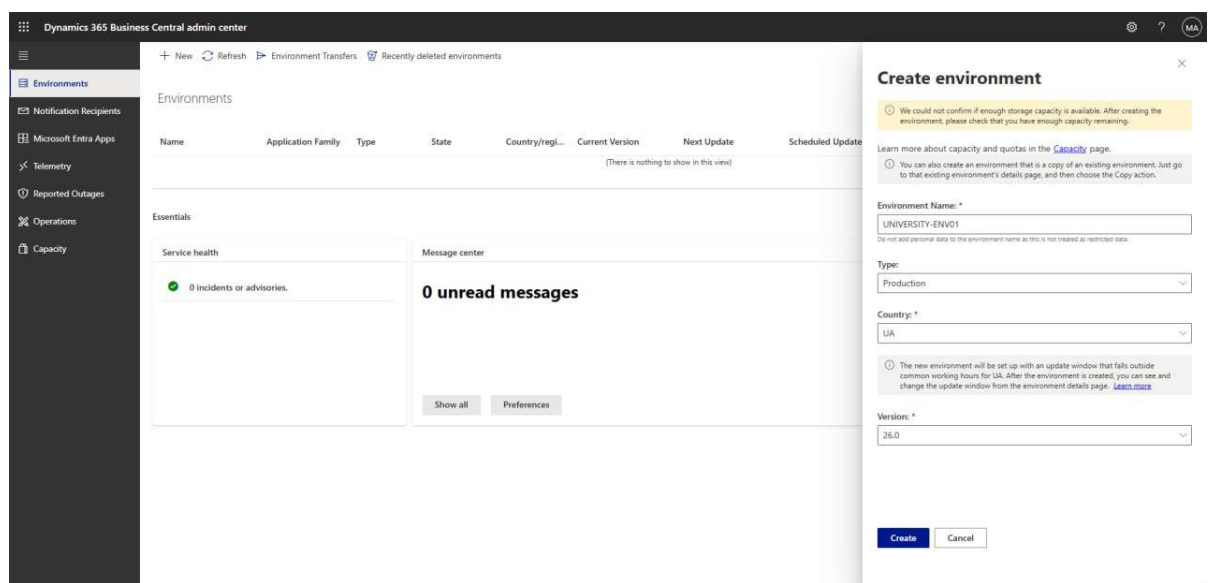


Рисунок 3.2.2 – Конфігурація середовища Dynamics 365 Business Central

3.3 Вимоги до технічних та програмних засобів

Спроектowana система планування ресурсів, що базується на платформі Microsoft Dynamics 365 Business Central, розрахована на експлуатацію в сучасній IT-інфраструктурі навчального закладу. Для гарантування її надійної, безперебійної та ефективної роботи, а також для забезпечення належного рівня продуктивності для всіх користувачів, критично важливо забезпечити відповідність апаратних платформ та програмного оточення встановленим системним специфікаціям, які зведені у таблицю 3.3.1.

Таблиця 3.3.1 – Вимоги до технічних та програмних засобів

Веб-браузери	Microsoft Edge (останні версії)
Операційна система (клієнт)	Windows 11 Windows 10 (версія 21H2 або новіша) macOS (останні дві версії)
Операційна система (сервер)	Windows Server 2022 Windows Server 2019
Процесор	1.6 ГГц або швидший, мінімум 2 ядра (рекомендовано 4 ядра або більше)
Оперативна пам'ять	Мінімум 4 ГБ (рекомендовано 8 ГБ або більше)
Вільне місце на диску	Мінімум 1 ГБ для клієнта Мінімум 20 ГБ для сервера
Роздільна здатність екрану	Мінімум 1366 x 768 (рекомендовано 1920 x 1080 або вище)
Мобільні пристрої	iOS 15.0 або новіше Android 9.0 або новіше
Бази даних	Microsoft SQL Server 2022
Додаткові компоненти	Microsoft .NET Framework 4.8 або новіше
	Microsoft Edge WebView2
	Runtime Microsoft Visual C++ Redistributable для Visual Studio 2019
Інтернет-з'єднання	Стабільне підключення (рекомендовано 10 Мбіт/с або швидше)
Інструменти розробки	Visual Studio Code з AL Language Extension
	Git для контролю версій

Чітке дотримання викладених тут технічних параметрів є запорукою успішного розгортання, адміністрування та подальшого розвитку програмної системи, дозволяючи повною мірою реалізувати її функціональний потенціал та забезпечити комфортну взаємодію для кінцевих користувачів.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		66

3.4 Тестування системи планування ресурсів

Тестування програмної системи є невід’ємним та критично важливим етапом життєвого циклу розробки будь-якого програмного продукту, зокрема такої комплексної системи. Процес тестування полягає у систематичній перевірці відповідності розробленої системи визначеним вимогам, виявленні дефектів або помилок та оцінці загальної якості програмного забезпечення перед його впровадженням та експлуатацією.

Тестування слугує ключовим інструментом для забезпечення надійності, стабільності та функціональної коректності програмної системи. Ретельне тестування дозволяє ідентифікувати та усунути потенційні проблеми на ранніх стадіях розробки, що значно знижує вартість їх виправлення порівняно з виявленням помилок вже після запуску програмної системи в повноцінне середовище функціонування.

Для тестування було вирішено використовувати підхід «Arrange-Act-Assert» (AAA) [33], який є широко визнаним стандартом у розробці програмного забезпечення для написання автоматизованих тестів і особливо добре підходить для тестування бізнес-логіки в системах планування ресурсів. Вибір цього підходу зумовлений його численними перевагами, зокрема чіткістю та структурованістю, які він надає кожному тесту.

Кожен тест логічно поділяється на три секції: підготовку, де готуються всі необхідні умови та дані; дію, де виконується безпосередньо тестова операція; перевірку, де результат виконання порівнюється з очікуваним за допомогою спеціальних тверджень.

Безпосередньо сам процес тестування системи реалізувався з використанням спеціалізованого інструментарію, відомого як «AL Test Toolkit». AL Test Toolkit [34] – це набір інструментів надає розробникам можливості для створення, управління та автоматизації тестів безпосередньо в середовищі розробки AL. Основою AL Test Toolkit є концепція створення тестів на базі AL-коду всередині спеціальних структурних об’єктів – кодових одиниць [35].

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						67
Зм.	Арк	№ докум.	Підпис	Дата		

Всередині цих тестових кодових одиниць створюються окремі тестові методи, кожен з яких перевіряє певну функціональність або логіку системи. Для управління процесом виконання тестів та їх інтеграції з іншими системами використовуються так звані набір інструментів «Test Runner» [36].

Він дозволяє запускати набори тестів, а також виконувати певні дії до та після виконання усього набору тестів. Інструментарій також підтримує концепцію тестових сторінок для перевірки логіки користувацького інтерфейсу без його реального відображення та обробники UI для автоматизації взаємодії з елементами інтерфейсу. Перед початком написання тестів, необхідно було налаштувати середовище, що включало встановлення «AL Test Toolkit» та усіх наявних залежностей.

Основні залежності інструментарію, які були встановлені:

- бібліотека «Library Assert» [37], яка надає набір методів для перевірки необхідних умов у тестах;
- бібліотека «Any» [38], яка надає набір методів для генерації випадкових даних та роботи з будь-якими типами даних у тестах;
- бібліотека «Test Runner» [39], яка є основним компонентом для запуску тестів та управління процесом тестування.

Наступним кроком у тестуванні системи стало безпосереднє створення тестових кодових одиниць. Для кожної сутності і функціонального блоку системи, була створена окрема тестова кодова одиниця. У цій кодовій одиниці були визначені тестові процедури, кожна з яких перевіряє певний аспект поведінки системи, а саме: перевірка операцій створення, модифікації, видалення записів, перевірка обчислювальних полів.

Відповідно до підходу «AAA», кожний тест у цій кодовій одиниці містить чітко визначені етапи. На етапі підготовки тесту готуються необхідні дані та умови. Наприклад, локальна процедура Initialize() , яка генерує унікальні значення для полів нової кафедри за допомогою бібліотеки «Any» та визначених в ній методів AlphanumericText() та AlphabeticText(). Нижче наявний код процедури для ініціалізації значень полів тестової сутності:

```

local procedure Initialize()
begin
    DepartmentNo := Any.AlphanumericText(20);
    DepartmentName := Any.AlphabeticText(50);
    FacultyNo := Any.AlphanumericText(20);
    HeadOfDepartment := Any.AlphanumericText(20);
end;

```

Далі, на етапі дії, виконується основна операція, що тестується. Наприклад, це створення нового запису кафедри за допомогою процедури.

Нижче наведений код процедури для створення нового запису:

```

local procedure CreateDepartment(var Department: Record "UNV Department")
begin
    Department.Init();
    Department."No." := DepartmentNo;
    Department.Name := DepartmentName;
    Department.Faculty := FacultyNo;
    Department."Head of Department" := HeadOfDepartment;
    Department.Insert(true);
end;

```

На завершальному етапі перевірки, за допомогою методів, визначених у бібліотеці «Library Assert», перевіряється, чи результат виконання дії відповідає очікуванням. Наприклад, перевіряється успішне створення кафедри та коректність заповнених полів, зміна даних сутності, видалення записів сутності.

Нижче наведений код процедури для перевірки створення кафедри та коректність заповнених полів цієї сутності:

```

[Test]
procedure TestCreateDepartment()
var
    Department: Record "UNV Department";
begin
    // [GIVEN]
    Initialize();

    // [WHEN]
    CreateDepartment(Department);

    // [THEN]
    Assert.IsTrue(Department.Get(DepartmentNo), 'Department should be
created');
    Assert.AreEqual(DepartmentName, Department.Name, 'Department name
should match');
    Assert.AreEqual(FacultyNo, Department.Faculty, 'Faculty should
match');
    Assert.AreEqual(HeadOfDepartment, Department."Head of Department",
'Head of Department should match');
end;

```

					КВРІПЗ. 2101075.01.06.ПЗ	Арк.
						69
Зм.	Арк	№ докум.	Підпис	Дата		

Нижче наведений код процедури для перевірки змін даних сутності, на прикладі повної назви кафедри:

```
[Test]
procedure TestModifyDepartment ()
var
    Department: Record "UNV Department";
    NewName: Text[100];
begin
    // [GIVEN]
    Initialize ();
    CreateDepartment (Department);
    NewName := Any.AlphabeticText (50);

    // [WHEN]
    Department.Name := NewName;
    Department.Modify (true);

    // [THEN]
    Department.Get (DepartmentNo);
    Assert.AreEqual (NewName, Department.Name, 'Department name should
be updated');
end;
```

Нижче наведений код процедури для перевірки видалення записів:

```
[Test]
procedure TestDeleteDepartment ()
var
    Department: Record "UNV Department";
begin
    // [GIVEN]
    Initialize ();
    CreateDepartment (Department);

    // [WHEN]
    Department.Delete (true);

    // [THEN]
    Assert.IsFalse (Department.Get (DepartmentNo), 'Department should be
deleted');
end;
```

Обчислювальні поля, які динамічно розраховують значення на основі інших даних у системі, також потребують особливо ретельної перевірки для забезпечення коректності обчислень та відображення інформації.

У процесі тестування обчислювальних полів основна увага приділяється верифікації правильності алгоритмів розрахунку, точності результатів та своєчасності оновлення даних. Тестові процедури передбачають моделювання різних сценаріїв використання, включаючи граничні випадки та нестандартні ситуації, щоб переконатися в стабільності та надійності роботи цих полів.

					КВРІПЗ. 2101075.01.06.ПЗ	Арк.
						70
Зм.	Арк	№ докум.	Підпис	Дата		

Наприклад, створення студента разом із пов'язаними записами факультету та групи, після чого виконується обчислення полів, що мають відобразити назви цих пов'язаних сутностей. Очікуваним результатом є точна відповідність між даними, що зберігаються у факультеті та групі, і значеннями, що обчислюються у відповідних полях студента.

Нижче наведений код процедури для перевірки обчислювальних полів сутності «Студент»:

```
[Test]
procedure TestStudentFlowFields()
var
    Student: Record "UNV Student";
    Faculty: Record "UNV Faculty";
    Group: Record "UNV Group";
    FacultyName: Text[100];
    GroupName: Text[100];
begin
    // [GIVEN]
    Initialize();

    FacultyName := Any.AlphabeticText(50);
    Faculty.Init();
    Faculty."No." := FacultyNo;
    Faculty.Name := FacultyName;
    Faculty.Insert(true);

    GroupName := Any.AlphabeticText(50);
    Group.Init();
    Group."No." := GroupNo;
    Group.Name := GroupName;
    Group.Faculty := FacultyNo;
    Group.Insert(true);

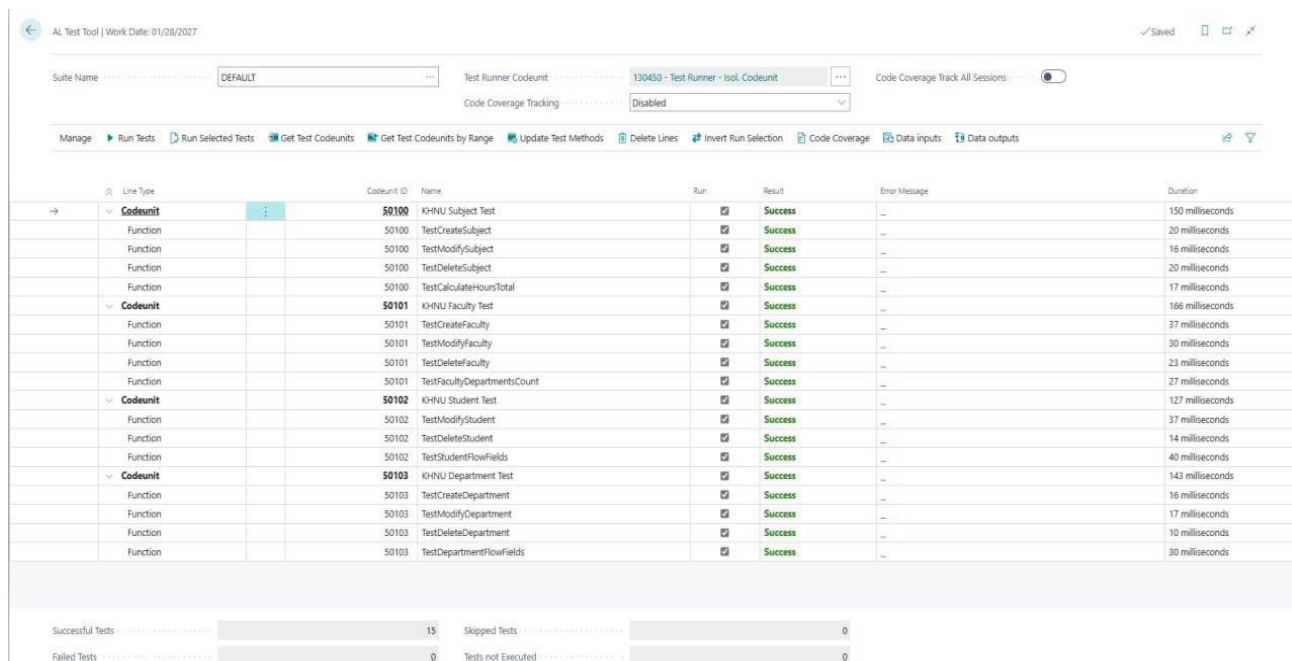
    CreateStudent(Student);

    // [WHEN]
    Student.CalcFields("Faculty Name", "Group Name");

    // [THEN]
    Assert.AreEqual(FacultyName, Student."Faculty Name", 'Faculty name
should match');
    Assert.AreEqual(GroupName, Student."Group Name", 'Group name
should match');
end;
```

Для решти сутностей програмної системи тестові кодові одиниці були розроблені за аналогічним підходом. Для кожної сутності було змодельовано відповідні початкові умови, виконано необхідні дії над даними та реалізовано перевірку очікуваних результатів, зокрема для обчислювальних полів, зв'язків

між записами та основних CRUD-операцій [40]. Такий підхід дозволив забезпечити послідовність у тестуванні, покрити основні сценарії використання та перевірити правильність логіки роботи системи на рівні кожної функціональної компоненти. Результати виконання тестових процедур продемонстровані на рисунку 3.4.1.



Line Type	Codeunit ID	Name	Run	Result	Error Message	Duration
Codeunit	50100	KHNU Subject Test	☑	Success	-	150 milliseconds
Function	50100	TestCreateSubject	☑	Success	-	20 milliseconds
Function	50100	TestModifySubject	☑	Success	-	16 milliseconds
Function	50100	TestDeleteSubject	☑	Success	-	20 milliseconds
Function	50100	TestCalculateHoursTotal	☑	Success	-	17 milliseconds
Codeunit	50101	KHNU Faculty Test	☑	Success	-	166 milliseconds
Function	50101	TestCreateFaculty	☑	Success	-	37 milliseconds
Function	50101	TestModifyFaculty	☑	Success	-	30 milliseconds
Function	50101	TestDeleteFaculty	☑	Success	-	23 milliseconds
Function	50101	TestFacultyDepartmentsCount	☑	Success	-	27 milliseconds
Codeunit	50102	KHNU Student Test	☑	Success	-	127 milliseconds
Function	50102	TestModifyStudent	☑	Success	-	37 milliseconds
Function	50102	TestDeleteStudent	☑	Success	-	14 milliseconds
Function	50102	TestStudentFlowFields	☑	Success	-	40 milliseconds
Codeunit	50103	KHNU Department Test	☑	Success	-	143 milliseconds
Function	50103	TestCreateDepartment	☑	Success	-	16 milliseconds
Function	50103	TestModifyDepartment	☑	Success	-	17 milliseconds
Function	50103	TestDeleteDepartment	☑	Success	-	10 milliseconds
Function	50103	TestDepartmentFlowFields	☑	Success	-	30 milliseconds

Successful Tests: 15 Skipped Tests: 0
Failed Tests: 0 Tests not Executed: 0

Рисунок 3.4.1 – Результати тестування програмної системи

У межах тестування були охоплені основні функціональні блоки системи, включно з перевіркою операцій створення, оновлення, видалення записів та правильності роботи обчислювальних полів, що є критичними для відображення динамічно пов'язаних даних. Також було перевірено обробку граничних випадків, реакцію системи на некоректні введення та відповідність фактичної поведінки очікуваним результатам у стандартних і нестандартних умовах.

Проведення тестування дозволило виявити та усунути потенційні помилки ще на ранніх етапах розробки, забезпечивши стабільність і передбачуваність поведінки системи в різних сценаріях її використання. У результаті вдалося не лише підтвердити правильність реалізованої логіки, але й сформувати надійний фундамент для подальшого розширення функціональності системи з впевненістю у якості її базових компонентів. Це, своєю чергою, мінімізує ризики виникнення критичних збоїв у майбутньому.

3.5 Висновки програмної реалізації та тестування системи планування ресурсів

У цьому розділі було висвітлено основні етапи розробки програмного забезпечення для системи планування ресурсів на базі Microsoft Dynamics 365 Business Central. Основна увага приділялася створенню функціональних компонентів і перевірці їхньої роботи в реальних умовах.

Розробка включала побудову структур зберігання даних та реалізацію зручних інтерфейсів для взаємодії з користувачем. Детально описано процес створення цих елементів, їхні функціональні особливості та роль у автоматизації бізнес-процесів. Для користувачів були створені різноманітні форми — від перегляду окремих записів до повноцінних списків — із наголосом на інтуїтивність та простоту у використанні.

Система була впроваджена за моделлю з архітектурою Single-Tenant що гарантує безпечне зберігання даних та гнучке налаштування під потреби конкретного бізнесу. Розгортання відбувалося у середовищі Microsoft CDX, яке забезпечує ізолюваність та контрольованість процесів.

Тестування функціоналу проводилося за методологією "підготовка – дія – твердження", що дозволило систематизувати процес перевірки. Для цього використовувалися спеціалізовані інструменти тестування AL-коду, які забезпечили точну верифікацію кожного модуля. Окрім цього, було визначено мінімальні технічні вимоги для стабільної роботи системи, що дозволяє забезпечити її ефективність на практиці.

Таким чином, проведена розробка та перевірка функціональних компонентів сформувала надійну основу для подальшого вдосконалення системи планування ресурсів. Отримані результати підтверджують доцільність обраного технічного підходу, що забезпечує гнучкість, масштабованість та високу продуктивність розробленої системи. Це дозволяє з упевненістю перейти до наступних етапів розробки, орієнтованих на розширення функціональності та інтеграцію нових модулів.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		73

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було успішно реалізовано комплексний проєкт зі створення системи планування ресурсів навчального закладу на базі Microsoft Dynamics 365. Основною метою роботи була розробка та впровадження інтегрованої системи планування ресурсів навчального закладу на базі Microsoft Dynamics 365, яка забезпечить централізоване управління матеріальними, людськими та фінансовими ресурсами, підвищить прозорість адміністративних процесів та оптимізує розподіл ресурсів відповідно до стратегічних цілей закладу.

На початковому етапі, було проведено ґрунтовне дослідження предметної області, яке включало в себе аналіз структурних та функціональних особливостей управління ресурсами в навчальних закладах, що дозволило виявити ключові проблеми, такі як недостатнє делегування управлінських процесів, відсутність автоматизованих інтегрованих рішень для ефективної обробки великих обсягів даних та недостатня ефективність використання наявних ресурсів.

Також було ретельно вивчено наявні програмно-технічні рішення, зокрема такі відомі системи як SAP for Education, Blackbaud та Ellucian Banner. Порівняльний аналіз цих систем дав можливість визначити їхні переваги та недоліки, а також обґрунтувати вибір Microsoft Dynamics 365 як оптимальної платформи для рішення, зважаючи на її гнучкість, широкі можливості інтеграції та адаптації до специфічних потреб освітньої сфери. На основі цих досліджень було сформульовано чіткі функціональні та нефункціональні вимоги до майбутньої програмної системи, які стали основою для подальшого проєктування та розробки.

У другому розділі було виконано етап проєктування програмної системи. Розроблено архітектуру та структуру системи, обираючи між підходами Single-Tenant та Multi-Tenant. Зважаючи на специфіку навчального закладу, було обрано архітектуру, яка забезпечує оптимальний баланс між гнучкістю,

					КВРІПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		74

безпекою даних та економічною ефективністю. Також було обґрунтовано вибір підходу до розгортання: між On-Premise та SaaS. Для рішення, що базується на Microsoft Dynamics 365, природним вибором став SaaS, що забезпечує легке масштабування, автоматичні оновлення та знижує витрати на підтримку інфраструктури. Спроектовано логічну модель бази даних, яка докладно відображає основні сутності, такі як студенти, викладачі, аудиторії, факультети, кафедри, групи та їхні взаємозв'язки. Це дозволило ефективно зберігати та обробляти всю необхідну інформацію про ресурси навчального закладу. Детальне проектування програмних модулів включало визначення їхньої логіки роботи, алгоритмів та взаємодії, що є критично важливим для коректного функціонування всієї системи.

Третій розділ був повністю присвячений програмній реалізації та тестуванню системи. Під час програмної реалізації модулів були створені ключові компоненти системи: таблиці для зберігання даних, сторінки для відображення та взаємодії з користувачем, роул-центри для швидкого доступу до ключової інформації та профілі для налаштування інтерфейсу під різні ролі користувачів. Розгортання програмної системи відбулося за допомогою Microsoft CDX (Customer Digital Experience), що дозволило ефективно розгорнути рішення в середовищі Microsoft Dynamics 365, враховуючи всі вимоги до технічних та програмних засобів для стабільної та продуктивної роботи. Проведено комплексне тестування програмної системи, використовуючи AL Test toolkit, Test Runner та Library Assert. Застосовувався підхід AAA (Arrange, Act, Assert), що дозволило систематично перевіряти кожен компонент системи, забезпечуючи її працездатність, відповідність вимогам та високу якість кінцевого продукту.

Впровадження розробленої системи надасть навчальним закладам значні переваги. Вона дозволить суттєво скоротити затрати часу на рутинні операції, такі як складання розкладу та реєстрація студентів. Це призведе до зменшення людських та фінансових ресурсів, оскільки багато процесів буде автоматизовано. Найголовніше, система підвищить ефективність управління

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		75

освітніми ресурсами, забезпечуючи більш раціональне використання аудиторій, обладнання, оптимізацію розкладу занять та ефективне управління персоналом. Завдяки централізованому доступу до даних, покращиться координація між різними відділами, а керівництво зможе приймати більш обґрунтовані рішення на основі актуальної інформації.

У перспективі вдосконалення системи може включати її з'єднання з цифровими освітніми середовищами для автоматизованого обміну інформацією щодо прогресу та присутності учнів, а також доповнення можливостями для ведення фінансових операцій навчальної установи, що охоплює планування бюджету та облік матеріальних цінностей.

Додатково, система може бути синхронізована з платформами онлайн-освіти для автоматичного узгодження планів занять та ресурсного забезпечення, а також збагачена прогресивними методами відображення даних та гнучким формуванням аналітичних оглядів за допомогою спеціалізованих програмних рішень для поглибленого вивчення розподілу ресурсів.

Таким чином, у ході виконання даної дипломної роботи було успішно здійснено комплексне завдання зі створення системи планування ресурсів на платформі Microsoft Dynamics 365. Робота охопила всі фундаментальні стадії: починаючи від формування концептуальних засад та поглибленого аналізу предметної області, через архітектурне проєктування та безпосередню розробку, і завершуючи ретельним тестуванням функціональності та підготовкою розгорнутого рішення до експлуатації.

Результатом проведених досліджень та практичної реалізації стало створення сучасного, надійного та інтуїтивно зрозумілого програмного середовища. Розроблена система не лише повною мірою відповідає завданням і цілям, що були сформульовані на початку даного дослідження, але й демонструє значну практичну цінність для оптимізації процесів планування.

Окрім того, вона відкриває широкі перспективи для подальшого вдосконалення, можливих модифікацій та розширення її функціональних можливостей у майбутньому, адаптуючись до нових викликів та потреб.

					КВРПЗ. 2101075.01.06.ПЗ	Арк.
						76
Зм.	Арк	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is ERP?. URL: <https://www.oracle.com/erp/what-is-erp/> (дата звернення: 01.01.2025).
2. Integrated DEFinition Methods (IDEF). URL: https://www.idef.com/idefo-function_modeling_method/ (дата звернення: 01.01.2025).
3. IDEF0 Diagram. URL: <https://vitechcorp.com/resources/GENESYS/onlinehelp/desktop/Views/IDEF0.htm> (дата звернення: 01.01.2025).
4. Education and Research. URL: <https://solutionportfolio.net.sap/industry/EDU> (дата звернення: 02.01.2025).
5. Відомості про компанію SAP. URL: <https://www.sap.com/ukraine/about/company.html> (дата звернення: 02.01.2025).
6. Blackbaud. URL: <https://www.blackbaud.com/> (дата звернення: 02.01.2025).
7. Blackbaud Inc: Overview. URL: <https://www.globaldata.com/company-profile/blackbaud-inc/> (дата звернення: 02.01.2025).
8. Ellucian Banner. URL: <https://www.ellucian.com/emea-ap/solutions/ellucian-banner> (дата звернення: 02.01.2025).
9. LinkedIn - Ellucian. URL: https://ua.linkedin.com/company/ellucian/?trk=products_details_guest_organization_page (дата звернення: 02.01.2025).
10. Dynamics 365 Business Central. URL: <https://www.microsoft.com/en-us/dynamics-365/products/business-central> (дата звернення: 02.01.2025).
11. Microsoft Corporation. URL: <https://www.britannica.com/money/Microsoft-Corporation> (дата звернення: 02.01.2025).
12. Chapter 6: Modeling Requirements. Requirements Engineering for Software and Systems. 4th ed. / М. Kassab, Р. Laplante. 4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN : Francis Group, LLC, 2022. С. 593.

					КВРІПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		77

13. What are Functional and Non-functional Requirements?. URL: <https://www.requiment.com/what-are-functional-and-non-functional-requirements/> (дата звернення: 15.01.2025).

14. European Credit Transfer and Accumulation System (ECTS). URL: <https://education.ec.europa.eu/education-levels/higher-education/inclusive-and-connected-higher-education/european-credit-transfer-and-accumulation-system> (дата звернення: 17.01.2025).

15. Nonfunctional Requirements. URL: <https://framework.scaledagile.com/nonfunctional-requirements> (дата звернення: 15.01.2025).

16. What are the differences between on-premises and cloud (SaaS) implementation?. URL: <https://www.digmatix.com/en/blogs/what-are-differences-between-premises-and-cloud-saas-implementation> (дата звернення: 21.02.2025).

17. What is software as a service (SaaS)?. URL: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-saas> (дата звернення: 21.02.2025).

18. Chapter 6: Building an N-tier Layered Architecture. Solutions Architect's Handbook / Saurabh Shrivastava, Neelanjali Srivastav. 2022. С. 590.

19. Planning Your Dynamics 365 Business Central Deployment. URL: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/deployment/plan-for-deployment> (дата звернення: 02.03.2025).

20. Understanding Business Central's Multi-Tenant Architecture: Customization, Security, and Scalability. URL: <https://archerpoint.com/microsoft-business-central-what-is-a-tenant-and-why-is-it-important/> (дата звернення: 02.03.2025).

21. Chapter 5: Advanced Data Modeling. Database Systems: Design, Implementation, & Management / C. Coronel, S. Morris. Boston, MA 02210 USA, 2022. С. 818.

22. Chapter 8.2.1: The Attributes. Database Design Using Entity-Relationship Diagrams / Sikha Saha Bagui, Richard Walsh Earp. 2023. С. 451.

					КВРІІЗ. 2101075.01.06.ІІЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		78

23. Chapter 5.3: Defining a Relationship for Our New Entity. Database Design Using Entity- Relationship Diagrams / Sikha Saha Bagui, Richard Walsh Earp. 2023. С. 451.

24. What Is a One-to-One Relationship in a Database?. URL: <https://vertabelo.com/blog/one-to-one-relationship-in-database/> (дата звернення: 10.03.2025).

25. What Is a One-to-Many Relationship in a Database? An Explanation with Examples. URL: <https://vertabelo.com/blog/one-to-many-relationship/> (дата звернення: 10.03.2025).

26. What Is a Many-to-Many Relationship in a Database? An Explanation with Three Examples. URL: <https://vertabelo.com/blog/many-to-many-relationship/> (дата звернення: 10.03.2025).

27. Chapter 4: Entity Relationship (ER) Modeling. Database Systems: Design, Implementation, & Management / C. Coronel, S. Morris. Boston, MA 02210 USA, 2022. С. 818

28. Appendix E: Comparison of ER Model Notations. Database Systems: Design, Implementation, & Management / C. Coronel, S. Morris. Boston, MA 02210 USA, 2022. С. 818.

29. The Unified Modeling Language. URL: <https://www.uml-diagrams.org/> (дата звернення: 16.03.2025).

30. Chapter 4. Business Process and Functional Modeling. Systems Analysis and Design: An Object-Oriented Approach with UML / A. Dennis, B. Wixom, D. Tegarden. John Wiley and Sons Ltd, 2021. С. 544.

31. Customer Digital Experience(CDX) to setup Business Central. URL: <https://community.dynamics.com/blogs/post/?postid=c95553d2-9137-4a47-944e-77e2686cdba1> (дата звернення: 22.03.2025).

32. Install and Uninstall Extensions (Apps) in Business Central. URL: <https://learn.microsoft.com/en-us/dynamics365/business-central/ui-extensions-install-uninstall> (дата звернення: 04.04.2025).

					КВРІІЗ. 2101075.01.06.ІЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		79

33. Unit Testing and the Arrange, Act and Assert (AAA) Pattern. URL: <https://medium.com/@pjbfgf/title-testing-code-ocd-and-the-aaa-pattern-df453975ab80> (дата звернення: 05.04.2025).

34. Chapter 4: The Test Tools, Standard Tests, and Standard Test Libraries. Automated Testing in Microsoft Dynamics 365 Business Central, 2nd ed. / L. Van Vugt. 2021. С. 406.

35. Extension Development Fundamentals. Mastering Microsoft Dynamics 365 Business Central: The complete guide for designing and integrating advanced Business Central solutions , 2nd ed. / S. Demiliani. 2024. С. 684.

36. AL Test Runner. URL: <https://marketplace.visualstudio.com/items?itemName=jamespearson.al-test-runner> (дата звернення: 20.04.2025).

37. Codeunit "Library Assert". URL: <https://microsoft.github.io/BCApps/reference/library-assert/Codeunit/System.TestLibraries.Utilities.Library-Assert.html> (дата звернення: 22.04.2025).

38. Codeunit Any. URL: <https://microsoft.github.io/BCApps/reference/any/Codeunit/System.TestLibraries.Utilities.Any.html> (дата звернення: 22.05.2025).

39. Create Test Runner Codeunits. URL: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-testrunner-codeunits> (дата звернення: 22.04.2025).

40. What is CRUD?. URL: <https://www.crowdstrike.com/en-us/cybersecurity-101/observability/crud/> (дата звернення: 25.05.2025).

41. Форкун Ю., Яшина О. Метод розробки та проектування архітектурної складової програмного застосунку. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES. 2023. С. 93.

42. Архітектурні методи оптимізації швидкодії та відмовостійкості програмних застосунків / Ю. Форкун та ін. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES. 2023. вип. 2. С. 196–201.

43. Порівняння програмних метрик для оцінки якості програмних продуктів / О. М. Яшина та ін. Вісник Хмельницького національного університету. 2021. вип. 5. С. 4.

					КВРІПЗ. 2101075.01.06.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		80

ДОДАТОК А
(обов'язковий)

КОД ПРОГРАМИ

```
table 50105 "UNV University Setup"
{
    Caption = 'University Setup';
    DataClassification = CustomerContent;

    fields
    {
        field(1; "Primary Key"; Code[10])
        {
            Caption = 'Primary Key';
            DataClassification = CustomerContent;
        }
        field(2; "Faculty No. Series"; Code[20])
        {
            Caption = 'Faculty Nos.';
            TableRelation = "No. Series".Code;
        }
        field(3; "Department No. Series"; Code[20])
        {
            Caption = 'Department Nos.';
            TableRelation = "No. Series".Code;
        }
        field(4; "Group No. Series"; Code[20])
        {
            Caption = 'Group Nos.';
            TableRelation = "No. Series".Code;
        }
        field(5; "Student No. Series"; Code[20])
        {
            Caption = 'Student Nos.';
            TableRelation = "No. Series".Code;
        }
        field(6; "Academic Employee No. Series"; Code[20])
        {
            Caption = 'Academic Employee Nos.';
            TableRelation = "No. Series".Code;
        }
    }
    keys
    {
        key(PK; "Primary Key")
        {
            Clustered = true;
        }
    }
}
```

```
table 50106 "UNV Subject Grade Entry"
{
    Caption = 'Subject Grade Entry';
    DataClassification = CustomerContent;

    fields
    {
```

```

field(1; "Entry No."; Integer)
{
    Caption = 'Entry No.';
    DataClassification = CustomerContent;
    AutoIncrement = true;
}
field(2; "Student No."; Code[20])
{
    Caption = 'Student No.';
    DataClassification = CustomerContent;
    TableRelation = "UNV Student"."No.";
}
field(3; "Subject No."; Code[20])
{
    Caption = 'Subject No.';
    DataClassification = CustomerContent;
    TableRelation = "UNV Subject"."No.";
}
field(4; "Teacher No."; Code[20])
{
    Caption = 'Teacher No.';
    DataClassification = CustomerContent;
    TableRelation = "UNV Academic Employee"."No.";
}
field(5; "Grade"; Integer)
{
    Caption = 'Grade';
    DataClassification = CustomerContent;
    MinValue = 0;
    MaxValue = 100;

    trigger OnValidate()
    begin
        case true of
            Grade >= 90:
                "Traditional Grade" := "Traditional Grade"::A;
            Grade >= 82:
                "Traditional Grade" := "Traditional Grade"::B;
            Grade >= 74:
                "Traditional Grade" := "Traditional Grade"::C;
            Grade >= 64:
                "Traditional Grade" := "Traditional Grade"::D;
            Grade >= 60:
                "Traditional Grade" := "Traditional Grade"::E;
            Grade >= 35:
                "Traditional Grade" := "Traditional Grade"::FX;
            else
                "Traditional Grade" := "Traditional Grade"::FX;
        end;
    end;
}
field(6; "Traditional Grade"; Enum "UNV Traditional Grade")
{
    Caption = 'Traditional Grade';
    DataClassification = CustomerContent;
    Editable = false;
}
field(7; "Date"; Date)
{
    Caption = 'Date';
    DataClassification = CustomerContent;
}
field(8; "Academic Year"; Text[10])
{
    Caption = 'Academic Year';
    DataClassification = CustomerContent;
}

```

```

    }
    field(9; "Semester"; Integer)
    {
        Caption = 'Semester';
        DataClassification = CustomerContent;
        MinValue = 1;
        MaxValue = 2;
    }
    field(10; "Comments"; Text[250])
    {
        Caption = 'Comments';
        DataClassification = CustomerContent;
    }
    field(11; "Is Final"; Boolean)
    {
        Caption = 'Is Final';
        DataClassification = CustomerContent;
        InitValue = true;
    }
    field(12; "Retake Count"; Integer)
    {
        Caption = 'Retake Count';
        DataClassification = CustomerContent;
        MinValue = 0;
        InitValue = 0;
    }
}

keys
{
    key(PK; "Entry No.")
    {
        Clustered = true;
    }
    key(Student; "Student No.")
    {
    }
    key(Subject; "Subject No.")
    {
    }
    key(Teacher; "Teacher No.")
    {
    }
    key(AcademicYear; "Academic Year", "Semester")
    {
    }
}

fieldgroups
{
    fieldgroup(DropDown; "Entry No.", "Student No.", "Subject No.", Grade)
    {
    }
}

trigger OnInsert()
begin
    if Date = 0D then
        Date := Today;
end;
}

table 50104 "UNV Subject"
{
    Caption = 'Subject';

```

```

DataClassification = CustomerContent;
DrillDownPageID = "UNV Subjects List";
LookupPageID = "UNV Subjects List";

fields
{
    field(1; "No."; Code[20])
    {
        Caption = 'No.';
        DataClassification = CustomerContent;
        NotBlank = true;
    }
    field(2; "Name"; Text[100])
    {
        Caption = 'Name';
        DataClassification = CustomerContent;
    }
    field(3; "Credits"; Decimal)
    {
        Caption = 'Credits';
        DataClassification = CustomerContent;
        DecimalPlaces = 1 : 1;
        MinValue = 0;
    }
    field(4; "Department"; Code[20])
    {
        Caption = 'Department';
        DataClassification = CustomerContent;
        TableRelation = "UNV Department"."No.";

        trigger OnValidate()
        begin
            CalcFields("Department Name");
        end;
    }
    field(5; "Responsible Teacher"; Code[20])
    {
        Caption = 'Responsible Teacher';
        DataClassification = CustomerContent;
        TableRelation = "UNV Academic Employee"."No.";

        trigger OnValidate()
        begin
            CalcFields("Teacher Name");
        end;
    }
    field(6; "Control Form"; Enum "UNV Control Form")
    {
        Caption = 'Control Form';
        DataClassification = CustomerContent;
    }
    field(7; "Description"; Text[250])
    {
        Caption = 'Description';
        DataClassification = CustomerContent;
    }
    field(8; "Hours Total"; Integer)
    {
        Caption = 'Hours Total';
        DataClassification = CustomerContent;
        MinValue = 0;
    }
    field(9; "Hours Lectures"; Integer)
    {
        Caption = 'Hours Lectures';
        DataClassification = CustomerContent;
    }
}

```

```

        MinValue = 0;
    }
    field(10; "Hours Practice"; Integer)
    {
        Caption = 'Hours Practice';
        DataClassification = CustomerContent;
        MinValue = 0;
    }
    field(11; "Hours Laboratory"; Integer)
    {
        Caption = 'Hours Laboratory';
        DataClassification = CustomerContent;
        MinValue = 0;
    }
    field(12; "Is Active"; Boolean)
    {
        Caption = 'Is Active';
        DataClassification = CustomerContent;
        InitValue = true;
    }
    field(13; "Department Name"; Text[100])
    {
        Caption = 'Department Name';
        FieldClass = FlowField;
        CalcFormula = lookup("UNV Department".Name where("No." =
field(Department)));
        Editable = false;
    }
    field(14; "Teacher Name"; Text[100])
    {
        Caption = 'Teacher Name';
        FieldClass = FlowField;
        CalcFormula = lookup("UNV Academic Employee".Name where("No." =
field("Responsible Teacher")));
        Editable = false;
    }
    field(15; "Semester"; Integer)
    {
        Caption = 'Semester';
        DataClassification = CustomerContent;
        MinValue = 1;
        MaxValue = 12;
    }
    field(16; "Academic Year"; Text[9])
    {
        Caption = 'Academic Year';
        DataClassification = CustomerContent;
    }
    field(17; "Credit Points"; Decimal)
    {
        Caption = 'Credit Points';
        DataClassification = CustomerContent;
        DecimalPlaces = 1 : 1;
        MinValue = 0;
    }
}

keys
{
    key(PK; "No.")
    {
        Clustered = true;
    }
    key(Name; Name)
    {
    }
}

```

```

        key(Department; Department)
        {
        }
        key(ResponsibleTeacher; "Responsible Teacher")
        {
        }
    }

    fieldgroups
    {
        fieldgroup(DropDown; "No.", Name, Credits, Department)
        {
        }
        fieldgroup(Brick; "No.", Name, Credits, "Control Form")
        {
        }
    }

    var
        CannotDeleteSubjectErr: Label 'You cannot delete subject %1, when an
active subject grade entry exists!', Comment = '%1 - subject';

    trigger OnDelete()
    var
        SubjectGradeEntry: Record "UNV Subject Grade Entry";
    begin
        SubjectGradeEntry.SetRange("Subject No.", "No.");
        if not SubjectGradeEntry.IsEmpty() then
            Error(CannotDeleteSubjectErr, "No.");
    end;
}

table 50100 "UNV Student"
{
    Caption = 'Student';
    DataClassification = CustomerContent;
    DrillDownPageID = "UNV Students List";
    LookupPageID = "UNV Students List";

    fields
    {
        field(1; "No."; Code[20])
        {
            Caption = 'No.';
        }
        field(2; "First Name"; Text[100])
        {
            Caption = 'First Name';
        }
        field(3; "Middle Name"; Text[100])
        {
            Caption = 'Middle Name';
        }
        field(4; "Last Name"; Text[100])
        {
            Caption = 'Last Name';
        }
        field(5; "Birth Day"; Date)
        {
            Caption = 'Birth Day';
        }
        field(6; Faculty; Code[20])
        {
            Caption = 'Faculty';
            TableRelation = "UNV Faculty";
        }
    }
}

```

```

        trigger OnValidate()
        begin
            this.CalcFields("Faculty Abbreviation");
            this.CalcFields("Faculty Name");
        end;
    }
    field(7; "Faculty Abbreviation"; Code[10])
    {
        Caption = 'Faculty Abbreviation';
        FieldClass = FlowField;
        CalcFormula = lookup("UNV Faculty".Abbreviation where("No." =
field(Faculty)));
        Editable = false;
    }
    field(8; "Faculty Name"; Text[100])
    {
        Caption = 'Faculty Name';
        FieldClass = FlowField;
        CalcFormula = lookup("UNV Faculty".Name where("No." =
field(Faculty)));
        Editable = false;
    }
    field(9; Department; Code[20])
    {
        Caption = 'Department';
        TableRelation = "UNV Department";

        trigger OnValidate()
        begin
            this.CalcFields("Department Name");
        end;
    }
    field(10; "Department Name"; Text[100])
    {
        Caption = 'Department Name';
        FieldClass = FlowField;
        CalcFormula = lookup("UNV Department".Name where("No." =
field(Department)));
        Editable = false;
    }
    field(11; Group; Code[20])
    {
        Caption = 'Group';
        TableRelation = "UNV Group";

        trigger OnValidate()
        begin
            this.CalcFields("Group Name");
        end;
    }
    field(12; "Group Name"; Text[100])
    {
        Caption = 'Group Name';
        FieldClass = FlowField;
        CalcFormula = lookup("UNV Group".Name where("No." = field(Group)));
        Editable = false;
    }
    field(13; "Education Form"; Enum "UNV Education Form")
    {
        Caption = 'Education Form';
    }
    field(14; "Education Degree"; Enum "UNV Education Degree")
    {
        Caption = 'Education Degree';
    }

```

```

    field(15; Image; Media)
    {
        Caption = 'Image';
        ExtendedDatatype = Person;
    }
    field(16; "No. Series"; Code[20])
    {
        Editable = false;
        TableRelation = "No. Series";
    }
}

keys
{
    key(PK; "No.")
    {
        Clustered = true;
    }
}

trigger OnInsert()
var
    UniversitySetup: Record "UNV University Setup";
    NoSeries: Codeunit "No. Series";
begin
    UniversitySetup.Get();

    "No. Series" := UniversitySetup."Student No. Series";
    if NoSeries.AreRelated(UniversitySetup."Student No. Series", xRec."No.
Series") then
        "No. Series" := xRec."No. Series";
    "No." := NoSeries.GetNextNo("No. Series");
end;
}

table 50102 "UNV Group"
{
    Caption = 'Group';
    DataClassification = CustomerContent;
    DrillDownPageID = "UNV Groups List";
    LookupPageID = "UNV Groups List";

    fields
    {
        field(1; "No."; Code[20])
        {
            Caption = 'No.';
        }
        field(2; Abbreviation; Code[10])
        {
            Caption = 'Abbreviation';
            OptimizeForTextSearch = true;
        }
        field(3; Name; Text[100])
        {
            Caption = 'Name';
            OptimizeForTextSearch = true;
        }
        field(4; Faculty; Code[20])
        {
            Caption = 'Faculty';
            TableRelation = "UNV Faculty";

            trigger OnValidate()
            begin

```

```

        this.CalcFields("Faculty Abbreviation");
        this.CalcFields("Faculty Name");
    end;
}
field(5; "Faculty Abbreviation"; Code[10])
{
    Caption = 'Faculty Abbreviation';
    FieldClass = FlowField;
    CalcFormula = lookup("UNV Faculty".Abbreviation where("No." =
field(Faculty)));
    Editable = false;
}
field(6; "Faculty Name"; Text[100])
{
    Caption = 'Faculty Name';
    FieldClass = FlowField;
    CalcFormula = lookup("UNV Faculty".Name where("No." =
field(Faculty)));
    Editable = false;
}
field(7; Department; Code[20])
{
    Caption = 'Department';
    TableRelation = "UNV Department";

    trigger OnValidate()
    begin
        this.CalcFields("Department Name");
    end;
}
field(8; "Department Name"; Text[100])
{
    Caption = 'Department Name';
    FieldClass = FlowField;
    CalcFormula = lookup("UNV Department".Name where("No." =
field(Department)));
    Editable = false;
}
field(9; Curator; Code[20])
{
    Caption = 'Curator';
    TableRelation = "UNV Academic Employee";

    trigger OnValidate()
    begin
        this.CalcFields("Curator Name");
    end;
}
field(10; "Curator Name"; Text[100])
{
    Caption = 'Curator Name';
    FieldClass = FlowField;
    CalcFormula = lookup("UNV Academic Employee".Name where("No." =
field(Curator)));
    Editable = false;
}
field(11; "No. of Students"; Integer)
{
    Caption = 'No. of Students';
    FieldClass = FlowField;
    CalcFormula = count("UNV Student" where(Group = field("No.")));
    Editable = false;
}
}
keys

```

```

    {
        key(PK; "No.")
        {
            Clustered = true;
        }
    }
}

```

table 50101 "UNV Faculty"

```

{
    Caption = 'Faculty';
    DataCaptionFields = "No.", Name;
    DataClassification = CustomerContent;
    DrillDownPageID = "UNV Faculties List";
    LookupPageID = "UNV Faculties List";

    fields
    {
        field(1; "No."; Code[20])
        {
            Caption = 'Faculty Code';
        }
        field(2; Abbreviation; Code[10])
        {
            Caption = 'Abbreviation';
            OptimizeForTextSearch = true;
        }
        field(3; Name; Text[100])
        {
            Caption = 'Name';
            OptimizeForTextSearch = true;
        }
        field(4; Dean; Code[20])
        {
            Caption = 'Dean';
            TableRelation = "UNV Academic Employee";

            trigger OnValidate()
            begin
                this.CalcFields("Dean Name");
            end;
        }
        field(5; "Dean Name"; Text[100])
        {
            Caption = 'Dean Name';
            FieldClass = FlowField;
            CalcFormula = lookup("UNV Academic Employee".Name where("No." =
field(Dean)));
            Editable = false;
        }
        field(7; "No. of Departments"; Integer)
        {
            Caption = 'No. of Employee';
            FieldClass = FlowField;
            CalcFormula = count("UNV Department" where(Faculty = field("No.")));
            Editable = false;
        }
        field(8; "No. of Employees"; Integer)
        {
            Caption = 'No. of Employee';
            FieldClass = FlowField;
            CalcFormula = count("UNV Academic Employee" where(Faculty =
field("No.")));
            Editable = false;
        }
    }
}

```

```

        field(9; "No. of Students"; Integer)
        {
            Caption = 'No. of Students';
            FieldClass = FlowField;
            CalcFormula = count("UNV Student" where("Faculty" = field("No.")));
            Editable = false;
        }
    }

keys
{
    key(PK; "No.")
    {
        Clustered = true;
    }
}
}

table 50107 "UNV Department"
{
    Caption = 'Department';
    DataCaptionFields = "No.", Name;
    DataClassification = CustomerContent;
    DrillDownPageID = "UNV Departments List";
    LookupPageID = "UNV Departments List";

    fields
    {
        field(1; "No."; Code[20])
        {
            Caption = 'Code';
        }
        field(2; Name; Text[100])
        {
            Caption = 'Name';
        }
        field(3; Faculty; Code[20])
        {
            Caption = 'Faculty';
            TableRelation = "UNV Faculty";

            trigger OnValidate()
            begin
                this.CalcFields("Faculty Abbreviation");
                this.CalcFields("Faculty Name");
            end;
        }
        field(4; "Faculty Abbreviation"; Code[10])
        {
            Caption = 'Faculty Abbreviation';
            FieldClass = FlowField;
            CalcFormula = lookup("UNV Faculty".Abbreviation where("No." =
field(Faculty)));
            Editable = false;
        }
        field(5; "Faculty Name"; Text[100])
        {
            Caption = 'Faculty Name';
            FieldClass = FlowField;
            CalcFormula = lookup("UNV Faculty".Name where("No." =
field(Faculty)));
            Editable = false;
        }
        field(6; "Head of Department"; Code[20])
        {
    }
}

```

```

        Caption = 'Head of Department';
        TableRelation = "UNV Academic Employee";

        trigger OnValidate()
        begin
            this.CalcFields("Head of Department Name");
        end;
    }
    field(7; "Head of Department Name"; Text[100])
    {
        Caption = 'Head of Department Name';
        FieldClass = FlowField;
        CalcFormula = lookup("UNV Academic Employee".Name where("No." =
field("Head of Department")));
        Editable = false;
    }
    field(8; "No. of Employees"; Integer)
    {
        Caption = 'No. of Employee';
        FieldClass = FlowField;
        CalcFormula = count("UNV Academic Employee" where(Department =
field("No.")));
        Editable = false;
    }
    field(9; "No. of Students"; Integer)
    {
        Caption = 'No. of Students';
        FieldClass = FlowField;
        CalcFormula = count("UNV Student" where(Department = field("No.")));
        Editable = false;
    }
    field(10; "No. of Groups"; Integer)
    {
        Caption = 'No. of Groups';
        FieldClass = FlowField;
        CalcFormula = count("UNV Group" where(Department = field("No.")));
        Editable = false;
    }
}
keys
{
    key(PK; "No.")
    {
        Clustered = true;
    }
}
}

```

```
table 50108 "UNV Activity Cue"
```

```

{
    Caption = 'UNV Activity Cue';

    fields
    {
        field(1; "Primary Key"; Code[10])
        {
            Caption = 'Primary Key';
        }
        field(2; "Total Students"; Integer)
        {
            Caption = 'Total Students';
            FieldClass = FlowField;
            CalcFormula = count("UNV Student");
        }
        field(4; "Total Faculties"; Integer)
    }
}

```

```

    {
        Caption = 'Total Faculties';
        FieldClass = FlowField;
        CalcFormula = count("UNV Faculty");
    }
    field(5; "Total Departments"; Integer)
    {
        Caption = 'Total Departments';
        FieldClass = FlowField;
        CalcFormula = count("UNV Department");
    }
    field(6; "Total Groups"; Integer)
    {
        Caption = 'Total Groups';
        FieldClass = FlowField;
        CalcFormula = count("UNV Group");
    }
    field(7; "Total Subjects"; Integer)
    {
        Caption = 'Total Subjects';
        FieldClass = FlowField;
        CalcFormula = count("UNV Subject");
    }
    field(8; "Total Employees"; Integer)
    {
        Caption = 'Total Employees';
        FieldClass = FlowField;
        CalcFormula = count("UNV Academic Employee");
    }
}

keys
{
    key(PK; "Primary Key")
    {
        Clustered = true;
    }
}

procedure CalculateCues()
begin
    CALCFIELDS(
        "Total Students",
        "Total Faculties",
        "Total Departments",
        "Total Groups",
        "Total Subjects",
        "Total Employees"
    );
end;
}

```

```

table 50103 "UNV Academic Employee"
{
    Caption = 'Academic Employee';
    DataClassification = CustomerContent;
    DrillDownPageID = "UNV Academic Employees List";
    LookupPageID = "UNV Academic Employees List";

    fields
    {
        field(1; "No."; Code[20])
        {
            Caption = 'Employee Code';
        }
    }
}

```

```

field(2; Name; Text[100])
{
    Caption = 'Employee Name';
}
field(3; "Academic Degree"; Text[50])
{
    Caption = 'Academic Degree';
}
field(4; Faculty; Code[20])
{
    Caption = 'Faculty';
    TableRelation = "UNV Faculty";

    trigger OnValidate()
    begin
        this.CalcFields("Faculty Abbreviation");
        this.CalcFields("Faculty Name");
    end;
}
field(5; "Faculty Abbreviation"; Code[10])
{
    Caption = 'Faculty Abbreviation';
    FieldClass = FlowField;
    CalcFormula = lookup("UNV Faculty".Abbreviation where("No." =
field(Faculty)));
    Editable = false;
}
field(6; "Faculty Name"; Text[100])
{
    Caption = 'Faculty Name';
    FieldClass = FlowField;
    CalcFormula = lookup("UNV Faculty".Name where("No." =
field(Faculty)));
    Editable = false;
}
field(7; Department; Code[20])
{
    Caption = 'Department';
    TableRelation = "UNV Department";

    trigger OnValidate()
    begin
        this.CalcFields("Department Name");
    end;
}
field(8; "Department Name"; Text[100])
{
    Caption = 'Department Name';
    FieldClass = FlowField;
    CalcFormula = lookup("UNV Department".Name where("No." =
field(Department)));
    Editable = false;
}
field(9; Image; Media)
{
    Caption = 'Image';
    ExtendedDatatype = Person;
}
field(10; "First Name"; Text[50])
{
    Caption = 'First Name';
    DataClassification = CustomerContent;
}
field(11; "Middle Name"; Text[50])
{
    Caption = 'Middle Name';
}

```

```

        DataClassification = CustomerContent;
    }
    field(12; "Last Name"; Text[50])
    {
        Caption = 'Last Name';
        DataClassification = CustomerContent;
    }
    field(13; "Birth Day"; Date)
    {
        Caption = 'Birth Day';
        DataClassification = CustomerContent;
    }
    field(14; "Academic Title"; Text[50])
    {
        Caption = 'Academic Title';
        DataClassification = CustomerContent;
    }
    field(15; "Position"; Text[50])
    {
        Caption = 'Position';
        DataClassification = CustomerContent;
    }
    field(16; "Phone No."; Text[30])
    {
        Caption = 'Phone No.';
        DataClassification = CustomerContent;
    }
    field(17; "Mobile Phone No."; Text[30])
    {
        Caption = 'Mobile Phone No.';
        DataClassification = CustomerContent;
    }
    field(18; "E-Mail"; Text[80])
    {
        Caption = 'E-Mail';
        DataClassification = CustomerContent;
    }
}

keys
{
    key(PK; "No.")
    {
        Clustered = true;
    }
}

```

```

profile "UNV Academic Profile"
{
    Caption = 'Academic Management';
    RoleCenter = "UNV Academic Role Center";
}

```

```

enum 50102 "UNV Traditional Grade"
{
    Extensible = true;

    value(0; A)
    {
        Caption = 'A';
    }
    value(1; B)
    {

```

```
        Caption = 'B';
    }
    value(2; C)
    {
        Caption = 'C';
    }
    value(3; D)
    {
        Caption = 'D';
    }
    value(4; E)
    {
        Caption = 'E';
    }
    value(5; FX)
    {
        Caption = 'FX';
    }
    value(6; F)
    {
        Caption = 'F';
    }
}
```

```
enum 50100 "UNV Education Form"
{
    Extensible = true;

    value(0; Budget)
    {
        Caption = 'Budget';
    }
    value(1; Contract)
    {
        Caption = 'Contract';
    }
}
```

```
enum 50101 "UNV Education Degree"
{
    Extensible = true;

    value(0; Bachelor)
    {
        Caption = 'Bachelor';
    }
    value(1; Master)
    {
        Caption = 'Master';
    }
    value(2; Doctor)
    {
        Caption = 'Doctor';
    }
}
```

```
enum 50103 "UNV Control Form"
{
    Extensible = true;

    value(0; Exam)
    {
```

```

        Caption = 'Exam';
    }
    value(1; Credit)
    {
        Caption = 'Credit';
    }
    value(2; "Diff Credit")
    {
        Caption = 'Diff Credit';
    }
}

```

page 50105 "UNV University Setup"

```

{
    ApplicationArea = All;
    Caption = 'University Setup';
    PageType = Card;
    SourceTable = "UNV University Setup";
    UsageCategory = Administration;

    layout
    {
        area(Content)
        {
            group(General)
            {
                Caption = 'General';

                field("Faculty No. Series"; Rec."Faculty No. Series")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Faculty No. Series
field.';
                }
                field("Department No. Series"; Rec."Department No. Series")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Department No. Series
field.';
                }
                field("Group No. Series"; Rec."Group No. Series")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Group No. Series
field.';
                }
                field("Student No. Series"; Rec."Student No. Series")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Student No. Series
field.';
                }
                field("Academic Employee No. Series"; Rec."Academic Employee No.
Series")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Academic Employee No.
Series field.';
                }
            }
        }
    }
    trigger OnOpenPage()
    begin
        Rec.Reset();
    end
}

```

```

        if not Rec.Get() then begin
            Rec.Init();
            Rec.Insert();
        end;
    end;
}

```

page 50118 "UNV Subjects List"

```

{
    ApplicationArea = All;
    Caption = 'Subjects List';
    PageType = List;
    SourceTable = "UNV Subject";
    CardPageID = "UNV Subject Card";
    Editable = false;
    UsageCategory = Lists;

    layout
    {
        area(content)
        {
            repeater(Subjects)
            {
                field("No."; Rec."No.")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the No. field.';
                }
                field(Name; Rec.Name)
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Name field.';
                }
                field(Credits; Rec.Credits)
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Credits field.';
                }
                field(Department; Rec.Department)
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Department field.';
                }
                field("Department Name"; Rec."Department Name")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Department Name
field.';
                }
                field("Responsible Teacher"; Rec."Responsible Teacher")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Responsible Teacher
field.';
                }
                field("Teacher Name"; Rec."Teacher Name")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Teacher Name field.';
                }
                field("Control Form"; Rec."Control Form")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Control Form field.';
                }
            }
        }
    }
}

```

```

field("Hours Total"; Rec."Hours Total")
{
    ApplicationArea = All;
    ToolTip = 'Specifies the value of the Hours Total field.';
}
field("Hours Lectures"; Rec."Hours Lectures")
{
    ApplicationArea = All;
    ToolTip = 'Specifies the value of the Hours Lectures
field. ';
}
field("Hours Practice"; Rec."Hours Practice")
{
    ApplicationArea = All;
    ToolTip = 'Specifies the value of the Hours Practice
field. ';
}
field("Hours_Laboratory"; Rec."Hours Laboratory")
{
    ApplicationArea = All;
    ToolTip = 'Specifies the value of the Hours Laboratory
field. ';
}
field("Is Active"; Rec."Is Active")
{
    ApplicationArea = All;
    ToolTip = 'Specifies the value of the Is Active field.';
}
field("Description"; Rec."Description")
{
    ApplicationArea = All;
    ToolTip = 'Specifies the value of the Description field.';
}
field("Semester"; Rec."Semester")
{
    ApplicationArea = All;
    ToolTip = 'Specifies the value of the Semester field.';
}
field("Academic Year"; Rec."Academic Year")
{
    ApplicationArea = All;
    ToolTip = 'Specifies the value of the Academic Year field.';
}
field("Credit Points"; Rec."Credit Points")
{
    ApplicationArea = All;
    ToolTip = 'Specifies the value of the Credit Points field.';
}
field("Teacher"; Rec."Responsible Teacher")
{
    ApplicationArea = All;
    ToolTip = 'Specifies the value of the Responsible Teacher
field. ';
}
}
}

actions
{
    area(Navigation)
    {
        action(OpenCard)
        {
            ApplicationArea = All;
            Caption = 'Open Card';
        }
    }
}

```

```

        Image = Card;
        RunObject = Page "UNV Subject Card";
        RunPageLink = "No." = field("No.");
        Promoted = true;
        PromotedCategory = Process;
        PromotedIsBig = true;
        ShortcutKey = 'Shift+F7';
        ToolTip = 'Executes the Open Card action.';
    }
    action(DepartmentCard)
    {
        ApplicationArea = All;
        Caption = 'Department Card';
        Image = Card;
        RunObject = Page "UNV Department Card";
        RunPageLink = "No." = field(Department);
        Promoted = true;
        PromotedCategory = Process;
        ToolTip = 'Executes the Department Card action.';
    }
    action(TeacherCard)
    {
        ApplicationArea = All;
        Caption = 'Teacher Card';
        Image = Card;
        RunObject = Page "UNV Academic Employee Card";
        RunPageLink = "No." = field("Responsible Teacher");
        Promoted = true;
        PromotedCategory = Process;
        ToolTip = 'Executes the Teacher Card action.';
    }
}
area(Processing)
{
    action(AttachDocument)
    {
        ApplicationArea = All;
        Caption = 'Attach Document';
        Image = Attach;
        ToolTip = 'Executes the Attach Document action.';

        trigger OnAction()
        var
            DocumentAttachmentDetails: Page "Document Attachment
Details";
            RecRef: RecordRef;
        begin
            RecRef.GetTable(Rec);
            DocumentAttachmentDetails.OpenForRecRef(RecRef);
            DocumentAttachmentDetails.RunModal();
        end;
    }
}
}

page 50121 "UNV Subject Grade Entries"
{
    ApplicationArea = All;
    Caption = 'Subject Grade Entries';
    PageType = List;
    SourceTable = "UNV Subject Grade Entry";
    UsageCategory = Lists;
    Editable = true;
}

```

```

layout
{
    area(content)
    {
        repeater(Controll)
        {
            field("Entry No."; Rec."Entry No.")
            {
                ApplicationArea = All;
                Editable = false;
                ToolTip = 'Specifies the value of the Entry No. field.';
            }
            field("Student No."; Rec."Student No.")
            {
                ApplicationArea = All;
                ToolTip = 'Specifies the value of the Student No. field.';

                trigger OnValidate()
                begin
                    CurrPage.Update();
                end;
            }
            field("Subject No."; Rec."Subject No.")
            {
                ApplicationArea = All;
                ToolTip = 'Specifies the value of the Subject No. field.';

                trigger OnValidate()
                begin
                    CurrPage.Update();
                end;
            }
            field("Teacher No."; Rec."Teacher No.")
            {
                ApplicationArea = All;
                ToolTip = 'Specifies the value of the Teacher No. field.';

                trigger OnValidate()
                begin
                    CurrPage.Update();
                end;
            }
            field("Grade"; Rec."Grade")
            {
                ApplicationArea = All;
                StyleExpr = GradeStyleExpr;
                ToolTip = 'Specifies the value of the Grade field.';
            }
            field("Traditional Grade"; Rec."Traditional Grade")
            {
                ApplicationArea = All;
                StyleExpr = GradeStyleExpr;
                ToolTip = 'Specifies the value of the Traditional Grade
field.';
            }
            field("Date"; Rec."Date")
            {
                ApplicationArea = All;
                ToolTip = 'Specifies the value of the Date field.';
            }
            field("Academic Year"; Rec."Academic Year")
            {
                ApplicationArea = All;
                ;
                ToolTip = 'Specifies the value of the Academic Year field.';
            }
        }
    }
}

```

```

        field("Semester"; Rec."Semester")
        {
            ApplicationArea = All;
            Tooltip = 'Specifies the value of the Semester field.';
        }
        field("Is Final"; Rec."Is Final")
        {
            ApplicationArea = All;
            Tooltip = 'Specifies the value of the Is Final" field.';
        }
        field("Retake Count"; Rec."Retake Count")
        {
            ApplicationArea = All;
            Tooltip = 'Specifies the value of the Retake Count field.';
        }
    }
}
area(factboxes)
{
    systempart(Notes; Notes)
    {
        ApplicationArea = Notes;
    }
    systempart(Links; Links)
    {
        ApplicationArea = RecordLinks;
    }
}
}

actions
{
    area(Navigation)
    {
        action(StudentCard)
        {
            ApplicationArea = All;
            Caption = 'Student Card';
            Image = Card;
            RunObject = Page "UNV Student Card";
            RunPageLink = "No." = field("Student No.");
            Promoted = true;
            PromotedCategory = Process;
            Tooltip = 'Executes the Student Card action.';
        }
        action(SubjectCard)
        {
            ApplicationArea = All;
            Caption = 'Subject Card';
            Image = Card;
            RunObject = Page "UNV Subject Card";
            RunPageLink = "No." = field("Subject No.");
            Promoted = true;
            PromotedCategory = Process;
            Tooltip = 'Executes the Subject Card action.';
        }
        action(TeacherCard)
        {
            ApplicationArea = All;
            Caption = 'Teacher Card';
            Image = Card;
            RunObject = Page "UNV Academic Employee Card";
            RunPageLink = "No." = field("Teacher No.");
            Promoted = true;
            PromotedCategory = Process;
            Tooltip = 'Executes the Teacher Card action.';
        }
    }
}

```

```

    }
}
area(Processing)
{
    action(SetFinal)
    {
        ApplicationArea = All;
        Caption = 'Set Final';
        Image = Completed;
        Promoted = true;
        PromotedCategory = Process;
        ToolTip = 'Executes the Set Final action.';

        trigger OnAction()
        begin
            Rec."Is Final" := true;
            Rec.Modify(true);
            CurrPage.Update();
        end;
    }
    action(IncrementRetake)
    {
        ApplicationArea = All;
        Caption = 'Increment Retake';
        Image = Refresh;
        Promoted = true;
        PromotedCategory = Process;
        ToolTip = 'Executes the Increment Retake action.';

        trigger OnAction()
        begin
            Rec."Retake Count" += 1;
            Rec.Modify(true);
            CurrPage.Update();
        end;
    }
}

trigger OnAfterGetRecord()
begin
    SetGradeStyle();
end;

var
    GradeStyleExpr: Text;

local procedure SetGradeStyle()
begin
    case true of
        Rec.Grade >= 90:
            GradeStyleExpr := 'Favorable';
        Rec.Grade >= 74:
            GradeStyleExpr := 'Ambiguous';
        Rec.Grade >= 60:
            GradeStyleExpr := 'Attention';
        else
            GradeStyleExpr := 'Unfavorable';
    end;
end;
}

page 50119 "UNV Subject Card"
{
    ApplicationArea = All;
    Caption = 'Subject Card';
}

```

```

PageType = Card;
SourceTable = "UNV Subject";
UsageCategory = Documents;

layout
{
    area(content)
    {
        group(General)
        {
            Caption = 'General';

            field("No."; Rec."No.")
            {
                ApplicationArea = All;
                Importance = Promoted;
                ToolTip = 'Specifies the value of the No. field.';
            }
            field(Name; Rec.Name)
            {
                ApplicationArea = All;
                Importance = Promoted;
                ToolTip = 'Specifies the value of the Name field.';
            }
            field(Credits; Rec.Credits)
            {
                ApplicationArea = All;
                Importance = Promoted;
                ToolTip = 'Specifies the value of the Credits field.';
            }
            field("Credit Points"; Rec."Credit Points")
            {
                ApplicationArea = All;
                ToolTip = 'Specifies the value of the Credit Points field.';
            }
            field("Control Form"; Rec."Control Form")
            {
                ApplicationArea = All;
                Importance = Promoted;
                ToolTip = 'Specifies the value of the Control Form field.';
            }
            field("Semester"; Rec."Semester")
            {
                ApplicationArea = All;
                ToolTip = 'Specifies the value of the Semester field.';
            }
            field("Academic Year"; Rec."Academic Year")
            {
                ApplicationArea = All;
                ToolTip = 'Specifies the value of the Academic Year field.';
            }
            field("Is Active"; Rec."Is Active")
            {
                ApplicationArea = All;
                ToolTip = 'Specifies the value of the Is Active field.';
            }
        }
        group(DepartmentInfo)
        {
            Caption = 'Department';

            field(Department; Rec.Department)
            {
                ApplicationArea = All;
                Importance = Promoted;
                ToolTip = 'Specifies the value of the Department field.';
            }
        }
    }
}

```

```

        trigger OnValidate()
        begin
            CurrPage.Update();
        end;
    }
    field("Department Name"; Rec."Department Name")
    {
        ApplicationArea = All;
        Editable = false;
        ToolTip = 'Specifies the value of the Department Name
field. ';
    }
    field("Responsible Teacher"; Rec."Responsible Teacher")
    {
        ApplicationArea = All;
        Importance = Promoted;
        ToolTip = 'Specifies the value of the Responsible Teacher
field. ';
    }

        trigger OnValidate()
        begin
            CurrPage.Update();
        end;
    }
    field("Teacher Name"; Rec."Teacher Name")
    {
        ApplicationArea = All;
        Editable = false;
        ToolTip = 'Specifies the value of the Teacher Name field. ';
    }
}
group(Hours)
{
    Caption = 'Hours';

    field("Hours Total"; Rec."Hours Total")
    {
        ApplicationArea = All;
        Importance = Promoted;
        ToolTip = 'Specifies the value of the Hours Total field. ';
    }
    field("Hours Lectures"; Rec."Hours Lectures")
    {
        ApplicationArea = All;
        ToolTip = 'Specifies the value of the Hours Lectures
field. ';
    }
    field("Hours Practice"; Rec."Hours Practice")
    {
        ApplicationArea = All;
        ToolTip = 'Specifies the value of the Hours Practice
field. ';
    }
    field("Hours_Laboratory"; Rec."Hours Laboratory")
    {
        ApplicationArea = All;
        ToolTip = 'Specifies the value of the Hours Laboratory
field. ';
    }
}
group(SubjectDescription)
{
    Caption = 'Description';

    field("Description"; Rec."Description")

```

```

        {
            ApplicationArea = All;
            MultiLine = true;
            ToolTip = 'Specifies the value of the Description field.';
        }
    }
}
area(FactBoxes)
{
    part(DocumentsFactBox; "Document Attachment Factbox")
    {
        ApplicationArea = All;
        Caption = 'Documents';
        SubPageLink = "Table ID" = const(Database::"UNV Subject"),
            "No." = field("No.");
    }
    systempart(Links; Links)
    {
        ApplicationArea = RecordLinks;
    }
    systempart(Notes; Notes)
    {
        ApplicationArea = Notes;
    }
}
}

actions
{
    area(Navigation)
    {
        action(DepartmentCard)
        {
            ApplicationArea = All;
            Caption = 'Department Card';
            Image = Card;
            RunObject = Page "UNV Department Card";
            RunPageLink = "No." = field(Department);
            Promoted = true;
            PromotedCategory = Process;
            ToolTip = 'Executes the Department Card action.';
        }
        action(TeacherCard)
        {
            ApplicationArea = All;
            Caption = 'Teacher Card';
            Image = Card;
            RunObject = Page "UNV Academic Employee Card";
            RunPageLink = "No." = field("Responsible Teacher");
            Promoted = true;
            PromotedCategory = Process;
            ToolTip = 'Executes the Teacher Card action.';
        }
        action(GradeEntries)
        {
            ApplicationArea = All;
            Caption = 'Grade Entries';
            Image = List;
            RunObject = Page "UNV Subject Grade Entries";
            RunPageLink = "Subject No." = field("No.");
            Promoted = true;
            PromotedCategory = Process;
            ToolTip = 'Executes the Grade Entries action.';
        }
    }
}
area(Processing)

```

```

    {
        action(AttachDocument)
        {
            ApplicationArea = All;
            Caption = 'Attach Document';
            Image = Attach;
            Promoted = true;
            PromotedCategory = Process;
            ToolTip = 'Executes the Attach Document action.';

            trigger OnAction()
            var
                DocumentAttachmentDetails: Page "Document Attachment
Details";
                RecRef: RecordRef;
            begin
                RecRef.GetTable(Rec);
                DocumentAttachmentDetails.OpenForRecRef(RecRef);
                DocumentAttachmentDetails.RunModal();
            end;
        }
        action(ActivateDeactivate)
        {
            ApplicationArea = All;
            Caption = 'Activate/Deactivate';
            Image = ChangeStatus;
            Promoted = true;
            PromotedCategory = Process;
            ToolTip = 'Executes the Activate/Deactivate action.';

            trigger OnAction()
            begin
                Rec."Is Active" := not Rec."Is Active";
                Rec.Modify(true);
                CurrPage.Update();
            end;
        }
    }
}

```

page 50101 "UNV Faculty Card"

```

{
    ApplicationArea = All;
    Caption = 'Faculty Card';
    PageType = Card;
    SourceTable = "UNV Faculty";
    UsageCategory = Documents;

    layout
    {
        area(content)
        {
            group(General)
            {
                Caption = 'General';

                field("No."; Rec."No.")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Faculty Code field.';
                }
                field(Abbreviation; Rec.Abbreviation)
                {
                    ApplicationArea = All;
                }
            }
        }
    }
}

```

```

        Tooltip = 'Specifies the value of the Abbreviation field.';
    }
    field(Name; Rec.Name)
    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the Name field.';
    }
    field("No_of_Departments"; Rec."No. of Departments")
    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the No. of Employee
field. ';
    }
    field("No_of_Employees"; Rec."No. of Employees")
    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the No. of Employee
field. ';
    }
    field("No_of_Students"; Rec."No. of Students")
    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the No. of Students
field. ';
    }
}
group(DeanInfo)
{
    Caption = 'Dean';

    field(Dean; Rec.Dean)
    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the Dean field.';

        trigger OnValidate()
        begin
            CurrPage.Update();
        end;
    }
    field("Dean Name"; Rec."Dean Name")
    {
        ApplicationArea = All;
        Editable = false;
        Tooltip = 'Specifies the value of the Dean Name field.';
    }
}
part(Departments; "UNV Faculty Departments")
{
    ApplicationArea = All;
    Caption = 'Faculty Departments';
    SubPageLink = "Faculty" = field("No.");
}
}
area(FactBoxes)
{
    part(DocumentsFactBox; "Document Attachment Factbox")
    {
        ApplicationArea = All;
        Caption = 'Documents';
        SubPageLink = "Table ID" = const(Database::"UNV Faculty"),
            "No." = field("No.");
    }
}
systempart(Links; Links)
{
    ApplicationArea = RecordLinks;
}

```

```

    }
    systempart(Notes; Notes)
    {
        ApplicationArea = Notes;
    }
}
actions
{
    area(Navigation)
    {
        action(DepartmentsList)
        {
            ApplicationArea = All;
            Caption = 'Departments List';
            Image = List;
            RunObject = Page "UNV Departments List";
            RunPageLink = "Faculty" = field("No.");
            Promoted = true;
            PromotedCategory = Process;
            ToolTip = 'Executes the Departments List action.';
        }
        action(StudentsList)
        {
            ApplicationArea = All;
            Caption = 'Students List';
            Image = List;
            RunObject = Page "UNV Students List";
            RunPageLink = "Faculty" = field("No.");
            Promoted = true;
            PromotedCategory = Process;
            ToolTip = 'Executes the Students List action.';
        }
    }
    area(Processing)
    {
        action(AttachDocument)
        {
            ApplicationArea = All;
            Caption = 'Attach Document';
            Image = Attach;
            Promoted = true;
            PromotedCategory = Process;
            ToolTip = 'Executes the Attach Document action.';

            trigger OnAction()
            var
                DocumentAttachmentDetails: Page "Document Attachment
Details";
                RecRef: RecordRef;
            begin
                RecRef.GetTable(Rec);
                DocumentAttachmentDetails.OpenForRecRef(RecRef);
                DocumentAttachmentDetails.RunModal();
            end;
        }
    }
}

page 50122 "UNV Activities Cue"
{
    PageType = CardPart;
    Caption = 'Activities';
    SourceTable = "UNV Activity Cue";
}

```

```

RefreshOnActivate = true;

layout
{
    area(content)
    {
        cuegroup(Students)
        {
            Caption = 'Students';

            field("Total Students"; Rec."Total Students")
            {
                ApplicationArea = All;
                DrillDownPageId = "UNV Students List";
                ToolTip = 'Shows the total number of students';
            }
        }

        cuegroup(Faculties)
        {
            Caption = 'Academic Structure';

            field("Total Faculties"; Rec."Total Faculties")
            {
                ApplicationArea = All;
                DrillDownPageId = "UNV Faculties List";
                ToolTip = 'Shows the total number of faculties';
            }

            field("Total Departments"; Rec."Total Departments")
            {
                ApplicationArea = All;
                DrillDownPageId = "UNV Departments List";
                ToolTip = 'Shows the total number of departments';
            }

            field("Total Groups"; Rec."Total Groups")
            {
                ApplicationArea = All;
                DrillDownPageId = "UNV Groups List";
                ToolTip = 'Shows the total number of student groups';
            }
        }

        cuegroup(Academics)
        {
            Caption = 'Academic Activities';

            field("Total Subjects"; Rec."Total Subjects")
            {
                ApplicationArea = All;
                DrillDownPageId = "UNV Subjects List";
                ToolTip = 'Shows the total number of subjects';
            }
            field("Total Employees"; Rec."Total Employees")
            {
                ApplicationArea = All;
                DrillDownPageId = "UNV Academic Employees List";
                ToolTip = 'Shows the total number of academic employees';
            }
        }
    }
}

page 50109 "UNV Students List"
{
    ApplicationArea = All;

```

```

Caption = 'Students List';
PageType = List;
SourceTable = "UNV Student";
CardPageID = "UNV Student Card";
Editable = false;
UsageCategory = Lists;

layout
{
    area(content)
    {
        repeater(Students)
        {
            field("No."; Rec."No.")
            {
                ApplicationArea = All;
                Tooltip = 'Specifies the value of the No. field.';
            }
            field("First Name"; Rec."First Name")
            {
                ApplicationArea = All;
                Tooltip = 'Specifies the value of the First Name field.';
            }
            field("Middle Name"; Rec."Middle Name")
            {
                ApplicationArea = All;
                Tooltip = 'Specifies the value of the Middle Name field.';
            }
            field("Last Name"; Rec."Last Name")
            {
                ApplicationArea = All;
                Tooltip = 'Specifies the value of the Last Name field.';
            }
            field("Birth Day"; Rec."Birth Day")
            {
                ApplicationArea = All;
                Tooltip = 'Specifies the value of the Birth Day field.';
            }
            field(Group; Rec.Group)
            {
                ApplicationArea = All;
                Tooltip = 'Specifies the value of the Group field.';
            }
            field("Group Name"; Rec."Group Name")
            {
                ApplicationArea = All;
                Tooltip = 'Specifies the value of the Group Name field.';
            }
            field(Faculty; Rec.Faculty)
            {
                ApplicationArea = All;
                Tooltip = 'Specifies the value of the Faculty field.';
            }
            field(Department; Rec.Department)
            {
                ApplicationArea = All;
                Tooltip = 'Specifies the value of the Department field.';
            }
            field("Education Form"; Rec."Education Form")
            {
                ApplicationArea = All;
                Tooltip = 'Specifies the value of the Education Form
field.';
            }
            field("Education Degree"; Rec."Education Degree")
            {

```

```

        ApplicationArea = All;
        Tooltip = 'Specifies the value of the Education Degree
field. ';
    }
}
}
area(FactBoxes)
{
    part(StudentPicture; "UNV Student Image Factbox")
    {
        ApplicationArea = All;
        Caption = 'Photo';
        SubPageLink = "No." = field("No.");
    }
}
}
actions
{
    area(Navigation)
    {
        action(OpenCard)
        {
            ApplicationArea = All;
            Caption = 'Student Card';
            Image = Card;
            RunObject = Page "UNV Student Card";
            RunPageLink = "No." = field("No.");
            Promoted = true;
            PromotedCategory = Process;
            PromotedIsBig = true;
            ShortcutKey = 'Shift+F7';
            Tooltip = 'Executes the Student Card action.';
        }
        action(OpenGroup)
        {
            ApplicationArea = All;
            Caption = 'Group Card';
            Image = Card;
            RunObject = Page "UNV Group Card";
            RunPageLink = "No." = field(Group);
            Promoted = true;
            PromotedCategory = Process;
            Tooltip = 'Executes the Group Card action.';
        }
    }
}
}
views
{
    view(ByFaculty)
    {
        Caption = 'By Faculty';
        Filters = where(Faculty = filter(<> '));
    }
}
}

```

```

page 50107 "UNV Student Image Factbox"
{
    Caption = 'Image Factbox';
    DeleteAllowed = false;
    InsertAllowed = false;
    LinksAllowed = false;
    PageType = CardPart;
}

```

```

SourceTable = "UNV Student";

layout
{
    area(content)
    {
        group(Control2)
        {
            ShowCaption = false;
            field(Image; Rec.Image)
            {
                ApplicationArea = All;
                ShowCaption = false;
            }
        }
    }
}

actions
{
    area(processing)
    {
        action("Import Image")
        {
            ApplicationArea = All;
            Caption = 'Import';
            Image = Import;
            ToolTip = 'Executes the Import action.';

            trigger OnAction()
            begin
                ImportImage();
            end;
        }
        action("Delete Image")
        {
            ApplicationArea = All;
            Caption = 'Delete';
            Image = Delete;
            ToolTip = 'Executes the Delete action.';

            trigger OnAction()
            begin
                DeleteImage();
            end;
        }
    }
}

local procedure ImportImage()
var
    ImageInStream: InStream;
    FromFileName: Text;
    OverrideImageQst: Label 'Current photo will be changed. Continue?';
begin
    if Rec.Image.HasValue then
        if not Confirm(OverrideImageQst) then
            exit;

            if UploadIntoStream('Import image', '', 'All files (*.*)|*.*',
FromFileName, ImageInStream) then begin
                Clear(Rec.Image);
                Rec.Image.ImportStream(ImageInStream, FromFileName);
                Rec.Modify(true);
            end;
end;
end;

```

```

local procedure DeleteImage()
var
    DeleteImageQst: Label 'Are you sure you want to delete the photo?';
begin
    if not Rec.Image.HasValue then
        exit;

    if Confirm(DeleteImageQst) then begin
        Clear(Rec.Image);
        Rec.Modify(true);
    end;
end;
}

page 50100 "UNV Student Card"
{
    ApplicationArea = All;
    Caption = 'Student Card';
    PageType = Card;
    UsageCategory = None;
    SourceTable = "UNV Student";

    layout
    {
        area(Content)
        {
            group(General)
            {
                Caption = 'General';

                field("No."; Rec."No.")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the No. field.';
                }
                field("First Name"; Rec."First Name")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the First Name field.';
                }
                field("Middle Name"; Rec."Middle Name")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Middle Name field.';
                }
                field("Last Name"; Rec."Last Name")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Last Name field.';
                }
                field("Birth Day"; Rec."Birth Day")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Birth Day field.';
                }
                field("Education Form"; Rec."Education Form")
                {
                    ApplicationArea = All;
                    ToolTip = 'Specifies the value of the Education Form
field.';
                }
                field("Education Degree"; Rec."Education Degree")
                {
                    ApplicationArea = All;

```

```

        Tooltip = 'Specifies the value of the Education Degree
field. ';
    }
}
group(FacultyInfo)
{
    Caption = 'Faculty';

    field(Faculty; Rec.Faculty)
    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the Faculty field.';

        trigger OnValidate()
        begin
            CurrPage.Update();
        end;
    }
    field("Faculty Abbreviation"; Rec."Faculty Abbreviation")
    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the Faculty Abbreviation
field. ';
    }
    field("Faculty Name"; Rec."Faculty Name")
    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the Faculty Name field.';
    }
}
group(DepartmentInfo)
{
    Caption = 'Department';

    field(Department; Rec.Department)
    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the Department field.';

        trigger OnValidate()
        begin
            CurrPage.Update();
        end;
    }
    field("Department Name"; Rec."Department Name")
    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the Department Name
field. ';
    }
}
group(GroupInfo)
{
    Caption = 'Group';

    field(Group; Rec.Group)
    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the Group field.';

        trigger OnValidate()
        begin
            CurrPage.Update();
        end;
    }
}

```

```

    }
    field("Group Name"; Rec."Group Name")
    {
        ApplicationArea = All;
        Tooltip = 'Specifies the value of the Group Name field.';
    }
}
area(FactBoxes)
{
    part(StudentPicture; "UNV Student Image Factbox")
    {
        ApplicationArea = All;
        Caption = 'Photo';
        SubPageLink = "No." = field("No.");
    }

    part(DocumentsFactBox; "Document Attachment Factbox")
    {
        ApplicationArea = All;
        Caption = 'Documents';
        SubPageLink = "Table ID" = const(Database::"UNV Student"),
            "No." = field("No.");
    }

    systempart(Links; Links)
    {
        ApplicationArea = RecordLinks;
    }

    systempart(Notes; Notes)
    {
        ApplicationArea = Notes;
    }
}
}
}

```

page 50106 "UNV Headline RC"

```

{
    PageType = HeadlinePart;
    Caption = 'Welcome to Academic Management';

    layout
    {
        area(content)
        {
            group(GreetingHeadline)
            {
                Visible = true;
                ShowCaption = false;

                field(Greeting; GetGreeting())
                {
                    ApplicationArea = All;
                    Caption = 'Greeting';
                    Editable = false;
                }

                field(UserGreeting; GetUserGreeting())
                {
                    ApplicationArea = All;
                    Caption = 'User Greeting';
                    Editable = false;
                }
            }
        }
    }
}

```

```

        field(CurrentSemester; GetCurrentSemester())
        {
            ApplicationArea = All;
            Caption = 'Current Semester';
            Editable = false;
        }
    }
}

local procedure GetGreeting(): Text
var
    Hour: Integer;
begin
    Hour := DT2Time(CurrentDateTime).Hour;

    if (Hour < 12) then
        exit('Good morning!');
    if (Hour < 18) then
        exit('Good afternoon!');
    exit('Good evening!');
end;

local procedure GetUserGreeting(): Text
var
    User: Record User;
    UserName: Text;
begin
    if User.Get(UserSecurityId()) then
        UserName := User."Full Name"
    else
        UserName := 'Administrator';

    exit('Welcome back, ' + UserName + '!');
end;

local procedure GetCurrentSemester(): Text
var
    Month: Integer;
begin
    Month := Date2DMY(Today, 2);

    if (Month >= 9) or (Month <= 1) then
        exit('Current Semester: Fall Semester');
    if (Month >= 2) and (Month <= 6) then
        exit('Current Semester: Spring Semester');
    exit('Current Semester: Summer Break');
end;
}

```

page 50108 "UNV Group Students"

```

{
    ApplicationArea = All;
    Caption = 'Group Students';
    PageType = ListPart;
    SourceTable = "UNV Student";
    CardPageID = "UNV Student Card";
    Editable = false;

    layout
    {
        area(content)
        {
            repeater(Students)
        }
    }
}

```

```

    {
        field("No."; Rec."No.")
        {
            ApplicationArea = All;
            Tooltip = 'Specifies the value of the No. field.';
        }
        field("First Name"; Rec."First Name")
        {
            ApplicationArea = All;
            Tooltip = 'Specifies the value of the First Name field.';
        }
        field("Last Name"; Rec."Last Name")
        {
            ApplicationArea = All;
            Tooltip = 'Specifies the value of the Last Name field.';
        }
        field("Education Form"; Rec."Education Form")
        {
            ApplicationArea = All;
            Tooltip = 'Specifies the value of the Education Form
field. ';
        }
        field("Education Degree"; Rec."Education Degree")
        {
            ApplicationArea = All;
            Tooltip = 'Specifies the value of the Education Degree
field. ';
        }
    }
}

actions
{
    area(Processing)
    {
        action(OpenCard)
        {
            ApplicationArea = All;
            Caption = 'Student Card';
            Image = Card;
            RunObject = Page "UNV Student Card";
            RunPageLink = "No." = field("No.");
            ShortcutKey = 'Shift+F7';
            Tooltip = 'Executes the Student Card action.';
        }
    }
}

```

page 50115 "UNV Groups List"

```

{
    ApplicationArea = All;
    Caption = 'Groups List';
    PageType = List;
    SourceTable = "UNV Group";
    CardPageID = "UNV Group Card";
    Editable = false;
    UsageCategory = Lists;

    layout
    {
        area(content)
        {
            repeater(Groups)

```

```

    {
        field("No."; Rec."No.")
        {
            ApplicationArea = All;
            ToolTip = 'Specifies the value of the No. field.';
        }
        field(Name; Rec.Name)
        {
            ApplicationArea = All;
            ToolTip = 'Specifies the value of the Name field.';
        }
        field(Abbreviation; Rec.Abbreviation)
        {
            ApplicationArea = All;
            ToolTip = 'Specifies the value of the Abbreviation field.';
        }
        field(Department; Rec.Department)
        {
            ApplicationArea = All;
            ToolTip = 'Specifies the value of the Department field.';
        }
        field("Department Name"; Rec."Department Name")
        {
            ApplicationArea = All;
            ToolTip = 'Specifies the value of the Department Name
field.';
        }
        field(Faculty; Rec.Faculty)
        {
            ApplicationArea = All;
            ToolTip = 'Specifies the value of the Faculty field.';
        }
        field("Faculty Name"; Rec."Faculty Name")
        {
            ApplicationArea = All;
            ToolTip = 'Specifies the value of the Faculty Name field.';
        }
        field(Curator; Rec.Curator)
        {
            ApplicationArea = All;
            ToolTip = 'Specifies the value of the Curator field.';
        }
        field("Curator Name"; Rec."Curator Name")
        {
            ApplicationArea = All;
            ToolTip = 'Specifies the value of the Curator Name field.';
        }
        field("No. of Students"; Rec."No. of Students")
        {
            ApplicationArea = All;
            ToolTip = 'Specifies the value of the No. of Students
field.';
        }
    }
}

actions
{
    area(Navigation)
    {
        action(OpenCard)
        {
            ApplicationArea = All;
            Caption = 'Open Card';
            Image = Card;
        }
    }
}

```

```

        RunObject = Page "UNV Group Card";
        RunPageLink = "No." = field("No.");
        Promoted = true;
        PromotedCategory = Process;
        PromotedIsBig = true;
        ShortcutKey = 'Shift+F7';
        ToolTip = 'Executes the Open Card action.';
    }
    action(StudentsList)
    {
        ApplicationArea = All;
        Caption = 'Students List';
        Image = List;
        RunObject = Page "UNV Students List";
        RunPageLink = "Group" = field("No.");
        Promoted = true;
        PromotedCategory = Process;
        ToolTip = 'Executes the Students List action.';
    }
    action(DepartmentCard)
    {
        ApplicationArea = All;
        Caption = 'Department Card';
        Image = Card;
        RunObject = Page "UNV Department Card";
        RunPageLink = "No." = field(Department);
        Promoted = true;
        PromotedCategory = Process;
        ToolTip = 'Executes the Department Card action.';
    }
    action(FacultyCard)
    {
        ApplicationArea = All;
        Caption = 'Faculty Card';
        Image = Card;
        RunObject = Page "UNV Faculty Card";
        RunPageLink = "No." = field(Faculty);
        Promoted = true;
        PromotedCategory = Process;
        ToolTip = 'Executes the Faculty Card action.';
    }
}
area(Processing)
{
    action(AttachDocument)
    {
        ApplicationArea = All;
        Caption = 'Attach Document';
        Image = Attach;
        ToolTip = 'Executes the Attach Document action.';

        trigger OnAction()
        var
            DocumentAttachmentDetails: Page "Document Attachment
Details";
            RecRef: RecordRef;
        begin
            RecRef.GetTable(Rec);
            DocumentAttachmentDetails.OpenForRecRef(RecRef);
            DocumentAttachmentDetails.RunModal();
        end;
    }
}
}

```

page 50117 "KHNU Employee Image Factbox"

```
{
  Caption = 'Employee Image';
  PageType = CardPart;
  SourceTable = "KHNU Academic Employee";

  layout
  {
    area(content)
    {
      group(General)
      {
        ShowCaption = false;

        field(Image; Rec.Image)
        {
          ApplicationArea = All;
          ToolTip = 'Specifies the value of the Image field.';
        }
      }
    }
  }

  actions
  {
    area(processing)
    {
      action(ImportImage)
      {
        ApplicationArea = All;
        Caption = 'Import';
        Image = Import;
        ToolTip = 'Executes the Import action.';
        trigger OnAction()
        var
          ImageInStream: InStream;
          FromFileName: Text;
        begin
          if UploadIntoStream('Choose photo', '', 'Image
(*.jpg;*.jpeg;*.png)|*.jpg;*.jpeg;*.png', FromFileName, ImageInStream) then
begin
          Clear(Rec.Image);
          Rec.Image.ImportStream(ImageInStream, FromFileName);
          Rec.Modify(true);
        end;
      end;
    }
    action>DeleteImage)
    {
      ApplicationArea = All;
      Caption = 'Delete';
      Image = Delete;
      ToolTip = 'Executes the Delete action.';

      trigger OnAction()
      begin
        if Confirm('Are you sure you want to delete the photo?')
then begin
          Clear(Rec.Image);
          Rec.Modify(true);
        end;
      end;
    }
  }
}
}
```

ДОДАТОК Б
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ СЛАЙДИ

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Кафедра інженерії програмного забезпечення

1

Кваліфікаційна робота: «Система планування ресурсів навчального закладу на базі Microsoft Dynamics 365»

Виконав: студент IV курсу, групи ІПЗ-21-1 Кваснюк Д.І.
Керівник: канд. тех. наук, доцент Яшина О.М.




Рисунок Б.1 – «Слайд: Титульна сторінка»

2

Актуальність теми

{ }

Виявлені наявні відкриті проблеми предметної області:

- недостатнє делегування управлінських процесів;
- відсутність автоматизованих інтегрованих рішень для ефективної обробки великих обсягів даних
- недостатня ефективність використання наявних ресурсів.




Рисунок Б.2 – «Слайд: Актуальність теми»

3

Мета та завдання

Метою роботи є розробка та впровадження системи планування ресурсів навчального закладу на базі платформи Microsoft Dynamics 365, яка забезпечить автоматизацію та оптимізацію процесів управління освітньою діяльністю.

Аналіз предметної області та існуючих рішень	Дослідити специфіку функціонування навчальних закладів та їхні потреби в управлінні ресурсами. Проаналізувати існуючі системи планування ресурсів.
Проектування архітектури системи	Розробити концептуальну модель системи планування ресурсів. Спроекувати структуру бази даних. Розробити архітектуру програмного забезпечення.
Розробка функціональних модулів системи	Розробка бізнес-процесів та робочих потоків. Створення користувацьких інтерфейсів. Налаштування системи безпеки та прав доступу.
Тестування системи	Розробка плану тестування. Проведення функціонального тестування. Аналіз результатів тестування та усунення виявлених недоліків.

Рисунок Б.3 – «Слайд: Мета та завдання»

4

Змістовий аналіз предметної області, її структурних та функціональних особливостей

Фундаментальна робота систем планування ресурсів базується на маніпулюванні активами та їх безпосереднім залученням у процеси підприємства.

Основні аспекти, які підлягають автоматизації. :

- формування звітності,
- управління запасами
- планування ресурсів.

Основні категорії ресурсів:

- матеріальні,
- людські
- фінансові
- інформаційні.

Рисунок Б.4 – «Слайд: Змістовий аналіз предметної області, її структурних та функціональних особливостей»

Аналіз наявного програмно-технічного забезпечення 5

Критерій	SAP 	Blackbaud 	Elucian Banner 	Dynamics 365 
Функціональна повнота	Висока	Середня	Висока	Висока
Гнучкість налаштування	Середня	Середня	Середня	Висока
Інтеграційні можливості	Високі	Середні	Високі	Високі
Вартість впровадження	Дуже висока	Висока	Висока	Середня
Складність впровадження	Висока	Середня	Висока	Середня
Локалізація для України	Часткова	Відсутня	Відсутня	Наявна
Хмарне розгортання	Доступне	Основний варіант	Доступне	Основний варіант
Аналітичні можливості	Високі	Середні	Середні	Високі
Масштабованість	Висока	Середня	Висока	Висока
Мобільний доступ	Обмежений	Наявний	Обмежений	Повний

Рисунок Б.5 – «Слайд: Аналіз наявного програмно-технічного забезпечення»

Визначення функціональних та нефункціональних вимог до ПЗ 6

Функціональні вимоги	Нефункціональні вимоги
Управління навчальними планами	Інтеграція з іншими системами
Аналіз використання ресурсів	Імпорт/експорт у стандартних форматах
Формування структури дисциплін	Візуалізація аналітики
Ведення бази викладачів і співробітників	Створення користувацьких звітів
Формування стандартних звітів	Експорт у PDF, Excel, Word
Створення та редагування дисциплін	
Автоматизоване складання рейтингу	
Ведення бази даних студентів	
Зберігання та робота із документами	

Рисунок Б.6 – «Слайд: Визначення функціональних та нефункціональних вимог до ПЗ»

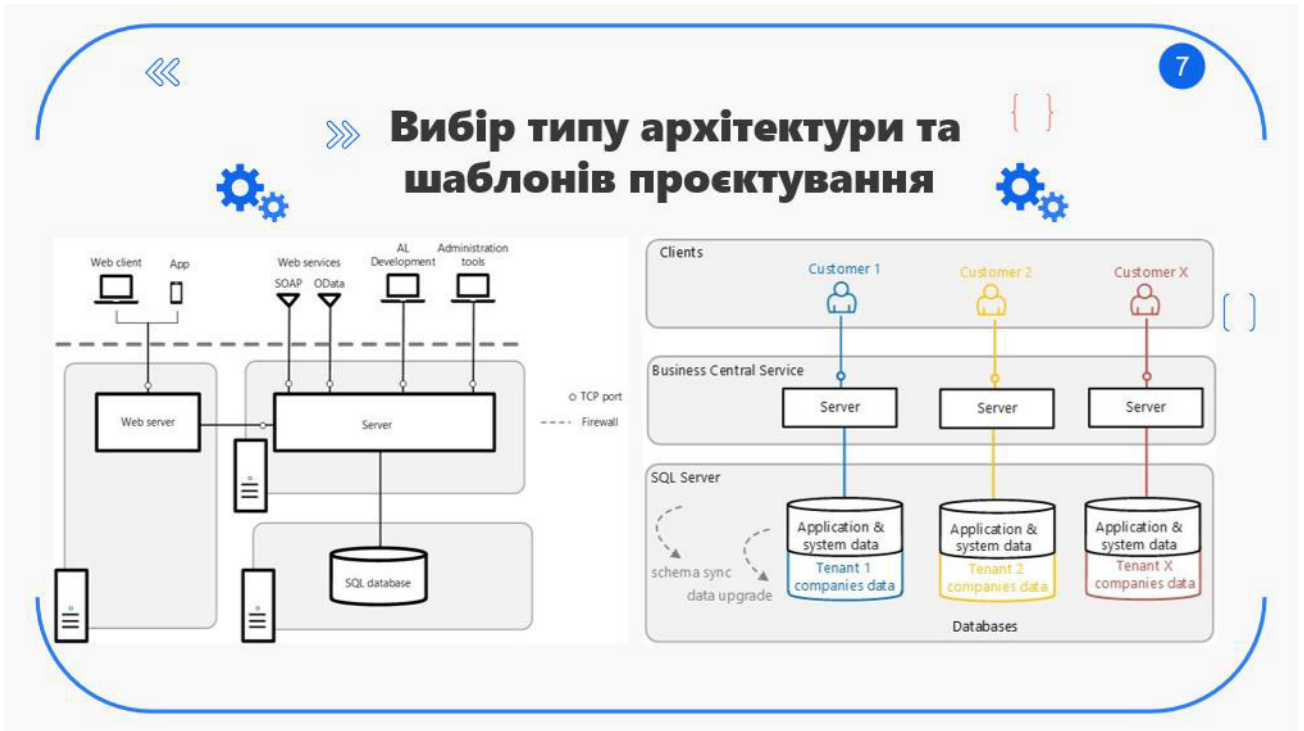


Рисунок Б.7 – «Слайд: Вибір типу архітектури та шаблонів проєктування»

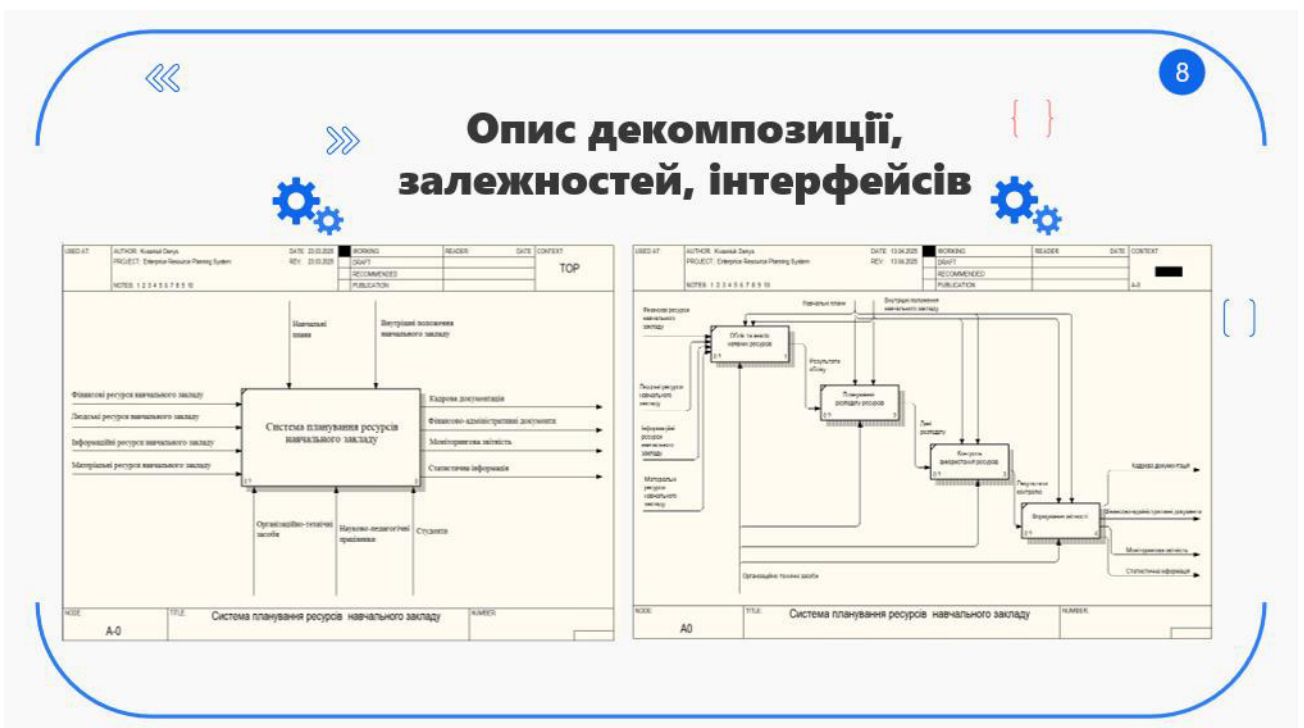


Рисунок Б.8 – «Слайд: Опис декомпозиції, залежностей, інтерфейсів»



Рисунок В.9 – «Слайд: Проектування модулів і даних»

Аналіз та вибір технологій

10

AL

AL

Мова програмування для платформи Microsoft Dynamics 365

Microsoft Dynamics 365

Платформа для розробки та розгортання застосунку

Microsoft SQL Server MS SQL Server

Система управління базами даних

Рисунок Б.10 – «Слайд: Аналіз та вибір технологій»

Вимоги до технічного та програмного забезпечення

11

Категорія	Вимога
Веб-браузери	Microsoft Edge (останні версії) Google Chrome (останні версії) Mozilla Firefox (останні версії) Safari (останні версії для macOS)
Операційна система (клієнт)	Windows 11 Windows 10 (версія 21H2 або новіша) macOS (останні дві версії)
Операційна система (сервер)	Windows Server 2022 Windows Server 2019
Процесор	1.6 ГГц або швидший, мінімум 2 ядра (рекомендовано 4 ядра або більше)
Оперативна пам'ять	Мінімум 4 ГБ (рекомендовано 8 ГБ або більше)
Вільне місце на диску	Мінімум 1 ГБ для клієнта Мінімум 20 ГБ для сервера
Роздільна здатність екрану	Мінімум 1366 x 768 (рекомендовано 1920 x 1080 або вище)
Мобільні пристрої	iOS 15.0 або новіше Android 9.0 або новіше
Бази даних	Microsoft SQL Server 2022 Microsoft SQL Server 2019 Microsoft SQL Server 2017 Azure SQL Database
Додаткові компоненти	Microsoft .NET Framework 4.8 або новіше Microsoft Edge WebView2 Runtime Microsoft Visual C++ Redistributable для Visual Studio 2019
Інтернет-з'єднання	Стабільне підключення (рекомендовано 10 Мбіт/с або швидше)
Інструменти розробки	Visual Studio Code з AL Language Extension Git для контролю версій Docker Desktop (для локальної розробки)

Рисунок Б.11 – «Слайд: Вимоги до технічного та програмного забезпечення»

Реалізація модулів

Приклад інтерфейсу однієї із сторінок програми на портативному комп'ютері

Work Date: 01/03/2027
⌕ + ⌵
Save

KHNU Student Card

General

Details Attachments (0)

Photo

Faculty

Department

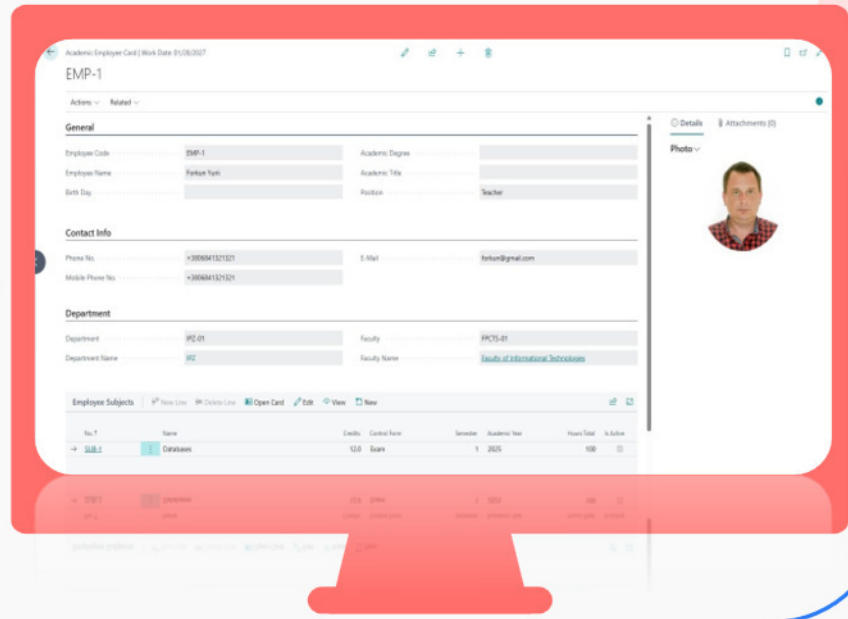
Group

12

Рисунок Б.12 – «Слайд: Реалізація модулів»

Реалізація модулів

Приклад інтерфейсу однієї із сторінок програми на портативному комп'ютері



13

Рисунок Б.13 – «Слайд: Реалізація модулів»

Тестування ПЗ

The screenshot shows a test runner interface with a table of test results. The table has columns for Line Type, Count/ID, Name, Run, Result, Error Message, and Duration. All tests listed are 'Successful'.

Line Type	Count/ID	Name	Run	Result	Error Message	Duration
Codeunit	50100	KRMU Subject Test	[Success]	Success	-	150 milliseconds
Function	50100	TestCreateSubject	[Success]	Success	-	20 milliseconds
Function	50100	TestDeleteSubject	[Success]	Success	-	16 milliseconds
Function	50100	TestCalculateHoursTotal	[Success]	Success	-	20 milliseconds
Codeunit	50101	KRMU Faculty Test	[Success]	Success	-	17 milliseconds
Function	50101	TestCreateFaculty	[Success]	Success	-	50 milliseconds
Function	50101	TestDeleteFaculty	[Success]	Success	-	27 milliseconds
Codeunit	50102	KRMU DepartmentTest	[Success]	Success	-	21 milliseconds
Function	50102	TestCreateDepartment	[Success]	Success	-	157 milliseconds
Function	50102	TestDeleteDepartment	[Success]	Success	-	27 milliseconds
Codeunit	50103	KRMU Department Test	[Success]	Success	-	14 milliseconds
Function	50103	TestCreateDepartment	[Success]	Success	-	40 milliseconds
Function	50103	TestDeleteDepartment	[Success]	Success	-	17 milliseconds
Function	50103	TestDepartmentFlowFields	[Success]	Success	-	16 milliseconds

Всі тести **успішно** пройдені, що свідчить про коректну роботу основних функцій системи. Середній час виконання тестів становить приблизно **24 мс**, що є хорошим показником продуктивності. Найдовше виконується тест TestStudentFlowFields (**40 мс**), що пов'язано з додатковими обчисленнями для FlowField полів.

Рисунок Б.14 – «Слайд: Тестування ПЗ»

Завдання	Результати виконання
Аналіз предметної області та існуючих рішень	Проведено дослідження специфіки функціонування навчальних закладів та їхніх потреб в управлінні ресурсами. Виконано аналіз існуючих систем планування ресурсів. Визначено ключові вимоги до системи на основі потреб навчальних закладів.
Проектування архітектури системи	Розроблено концептуальну модель системи планування ресурсів. Спроектовано структуру бази даних з основними. Розроблено архітектуру програмного забезпечення на базі Microsoft Dynamics 365 Business Central.
Розробка функціональних модулів системи	Створено основні модулі системи для управління факультетами, кафедрами, предметами, студентами та оцінками. Розроблено користувацький інтерфейс для роботи з даними. Налаштовано систему безпеки та права доступу відповідно до ролей користувачів.
Тестування системи	Розроблено та реалізовано план тестування. Проведено функціональне тестування всіх модулів. Виконано аналіз результатів тестування, який підтвердив коректну роботу системи. Середній час виконання тестів становить приблизно 24 мс, що свідчить про хорошу продуктивність системи.

Висновки

Розроблена система планування ресурсів для навчальних закладів на базі Microsoft Dynamics 365 Business Central **успішно** реалізує всі поставлені завдання. Система забезпечує ефективне управління основними ресурсами навчального закладу та готова до впровадження, та може бути масштабована відповідно до зростаючих потреб навчального закладу.

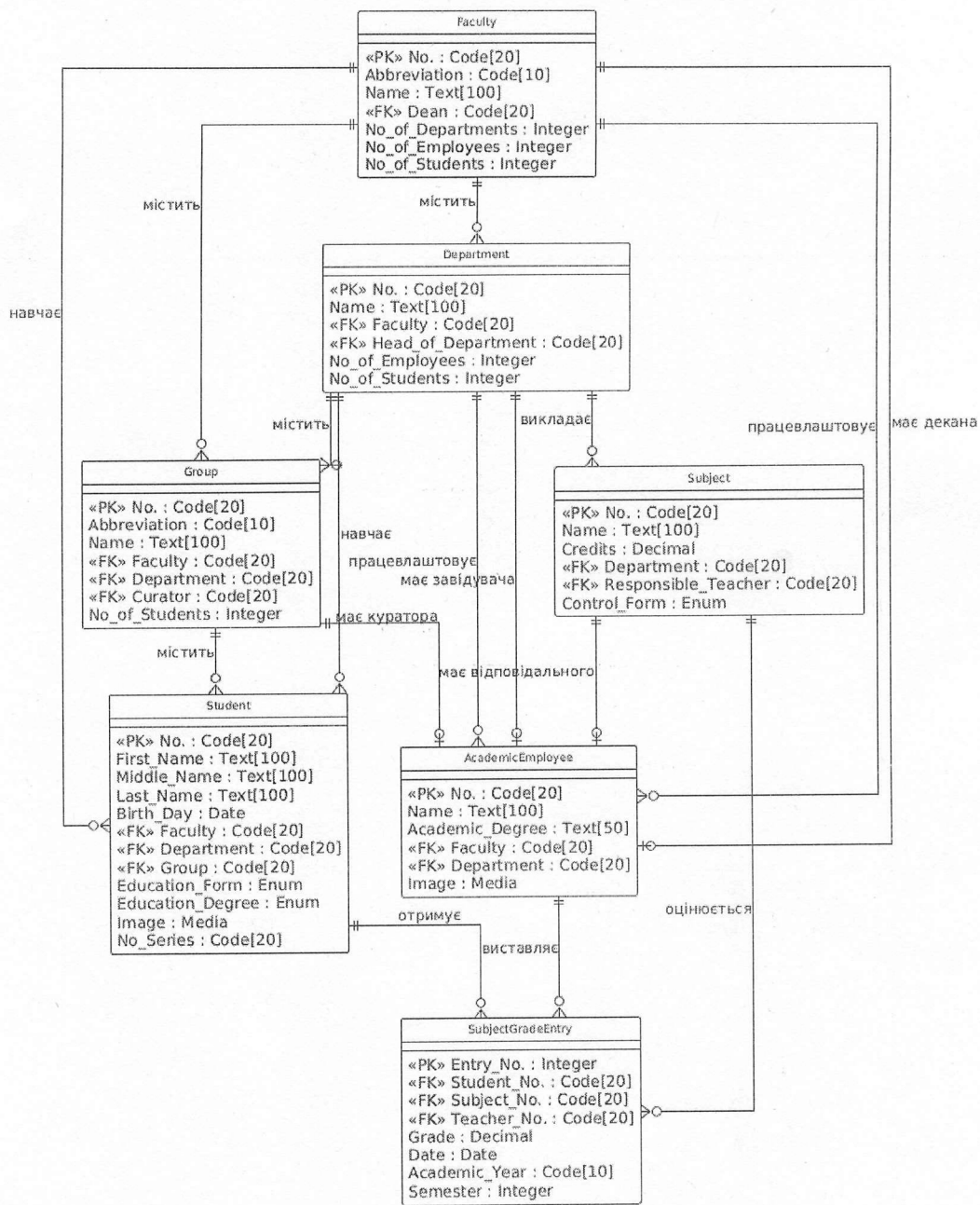
Рисунок Б.15 – «Слайд: Висновки»

Дякую за увагу!

The illustration shows a hand in a red glove touching a screen. The screen displays a progress indicator with a blue circle and a percentage of 100. To the right, there are several code snippets in a dark box, including: `((({{>>}}))<<)`, `((>>0|□□□))`, `((: 00 - =>>))`, `{ (<1 00 1 000 >>)}`, `((: 0)'><)`, `<01 001> +100 0>`, `((: 0)'><)`, and `{ (<1 00 1 000 >>)}`. The background features decorative blue and red shapes and navigation icons.

Рисунок Б.16 – «Слайд: Дякую за увагу»

ГРАФІЧНІ МАТЕРІАЛИ



КВРІПЗ.2101075.01.06.E8

Система планування ресурсів
навчального закладу на базі
Microsoft Dynamics 365

Логічна модель бази даних

Літ. Маса. Масштаб

Змн	Арк.	№ докум.	Підпис	Дата
Розробив		Квасюк Д.І.	<i>[Signature]</i>	02.06
Керівник		Яшина О.М.	<i>[Signature]</i>	04.06
Н. Контр.		Форкун Ю.В.	<i>[Signature]</i>	04.06
Зав. каф.		Бедратюк Л.П.	<i>[Signature]</i>	05.06

Аркуш 1 | Аркушів 3

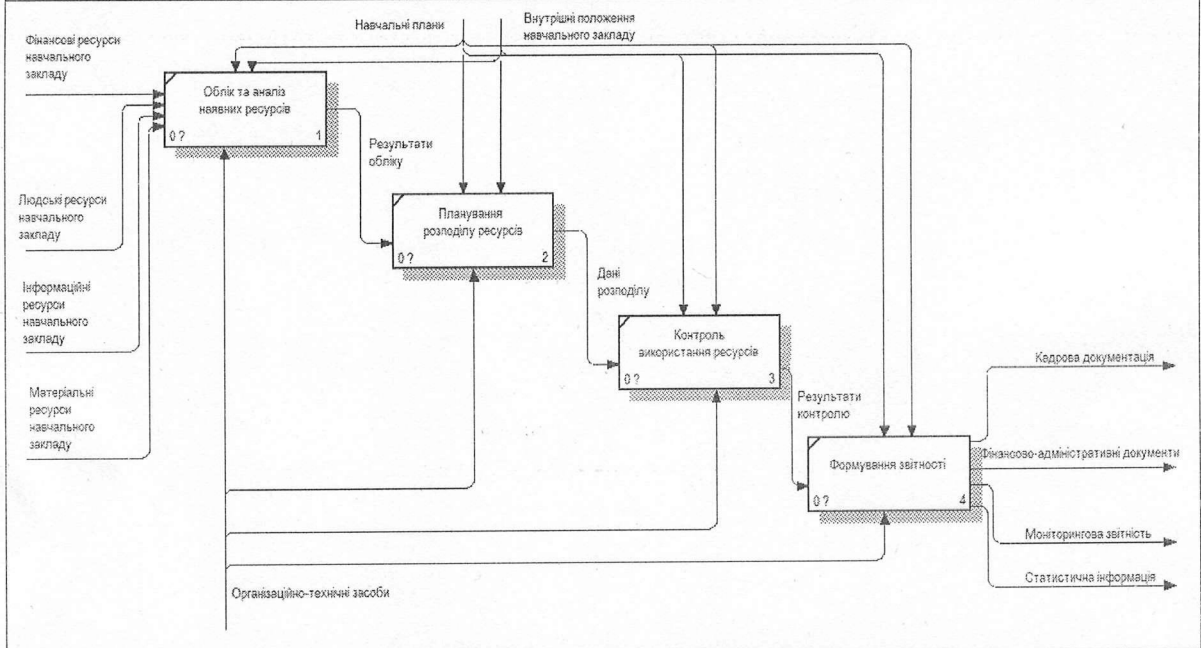
ХНУ, ІПЗ-21-1

USED AT:	AUTHOR: Kvasniuk Denys PROJECT: Enterprise Resource Planning System NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 23.03.2025 REV: 23.03.2025	WORKING DRAFT RECOMMENDED PUBLICATION	READER	DATE	CONTEXT: TOP
----------	---	-------------------------------------	--	--------	------	-----------------

NODE:	TITLE: Система планування ресурсів навчального закладу	NUMBER:
A-0		

					КВРІПЗ.2101075.01.06.E8		
					Система планування ресурсів навчального закладу на базі Microsoft Dynamics 365		
					Контекстна діаграма		
Змн	Арк.	№ докум.	Підпис	Дата	Лім.	Маса.	Масштаб
Розробив		Кваснюк Д.І.	<i>[Signature]</i>	04.06			
Керівник		Яцига О.М.	<i>[Signature]</i>	04.06			
					Аркуш 2		Аркушів 3
					ХНУ, ІПЗ-21-1		
Н. Контр.		Форкун Ю.В.	<i>[Signature]</i>	04.06			
Зав. каф.		Бедратюк Л.П.	<i>[Signature]</i>	05.06			

USED AT:	AUTHOR: Kvasniuk Denys	DATE: 13.04.2025	WORKING	READER	DATE	CONTEXT:
	PROJECT: Enterprise Resource Planning System	REV: 13.04.2025	DRAFT			
	NOTES: 1 2 3 4 5 6 7 8 9 10		RECOMMENDED			
			PUBLICATION			A-0



NOOE:	TITLE: Система планування ресурсів навчального закладу	NUMBER:
A0		

					КВРІПЗ.2101075.01.06.E8		
					Система планування ресурсів навчального закладу на базі Microsoft Dynamics 365		
					Діаграма декомпозиції		
Змн.	Арк.	№ докум.	Підпис	Дата	Лім.	Маса.	Масштаб
Розробив		Кваснюк Д.І.	<i>[Signature]</i>	04.06			
Керівник		Яшина О.М.	<i>[Signature]</i>	04.06			
					Аркуш 3		Аркушів 3
					ХНУ, ІПЗ-21-1		
Н. Контр.		Форкун Ю.В.	<i>[Signature]</i>	04.06			
Зав. каф.		Бедратюк Л.П.	<i>[Signature]</i>	05.06			

СУПРОВІДНІ ДОКУМЕНТИ

Завідувачу кафедри інженерії програмного
забезпечення проф. Леоніду БЕДРАТЮКУ
здобувача вищої освіти
Кваснюка Дениса Івановича
факультет ІТ, ІV курс, група ІІЗ-21-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності в Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії Хмельницького національного університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-обчислювального комплексу StrikePlagiarism та/або програмно-технічного засобу AntiPlagiarism і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення текстових збігів у роботах.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

04.06.2025
дата


підпис

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 1.0%

Dictionary check: en_US, ru_RU, ua_UA. **Errors in the documents: 8%**

ID: 242710 Title: БКР Система планування ресурсів навчального закладу на базі Microsoft Dynamics 365 Added in a DB: 2025-06-02 Authors: КВАСНЮК Денис Heads: Оксана ЯШИНА канд. тех. наук, доцент Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	91176	1395	2862 (3%)	44 (3%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Кваснюк Денис

Співавтор:

Назва: БКР Система планування ресурсів навчального закладу на базі Microsoft Dynamics 365

Науковий керівник:

Підрозділ: Кафедра інженерії програмного забезпечення

Коефіцієнт подібності 1: 1.7%

Коефіцієнт подібності 2: 0.2%

Мікропробіли: 0

Заміна букв: 2

Інтервали: 0

Блі знаки: 0

Дата створення звіту: 2025-06-01 16:06:37.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1. Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

Дата

01.06.2025

експерт



РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «Бакалавр»

Дипломник Кваснюк Денис Іванович

Тема Система планування ресурсів навчального закладу на базі Microsoft Dynamics 365

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 кількість сторінок записки 80

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі розглядається розробка інтегрованої системи планування та управління ресурсами для навчальних закладів на базі Microsoft Dynamics 365. Проведено комплексний аналіз предметної області, що включав дослідження існуючих програмних рішень та формування функціональних і нефункціональних вимог. Обґрунтовано вибір архітектурних та технологічних рішень для проектування системи, зокрема її логічної моделі бази даних та програмних модулів. Реалізовано функціональну систему, що охоплює розробку табличних об'єктів даних, інтерфейсу користувача та розгортання. Проведене тестування підтвердило коректну роботу програмного забезпечення.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням всіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У кваліфікаційній роботі розглядається створення системи планування ресурсів для навчального закладу на базі Microsoft Dynamics 365. У вступі обґрунтовано актуальність теми, визначено мету та завдання дипломного проектування. У першому розділі проведено всебічний аналіз предметної області, що включав виявлення ключових проблем та ретельне вивчення наявних програмно-технічних рішень, що дозволило обґрунтувати вибір платформи та сформулювати функціональні та нефункціональні вимоги. У другому розділі спроектовано архітектуру та структуру системи з обґрунтуванням вибору моделі розгортання SaaS, а також розроблено логічну модель бази даних та деталізовано програмні модулі. У третьому розділі реалізовано ключові компоненти системи та здійснено розгортання системи. Також було проведено комплексне тестування, яке підтвердило працездатність системи та відповідність вимогам. Результатом роботи стало створення сучасного, надійного та інтуїтивно зрозумілого програмного середовища, яке оптимізує процеси планування ресурсів.

4. Позитивні сторони роботи Кваліфікаційна робота є актуальною, оскільки ефективне планування та управління ресурсами є критично важливим для сучасних навчальних закладів, а розробка інтегрованої системи на базі Microsoft Dynamics 365 пропонує сучасне та потужне рішення для цих завдань. Робота вирізняється всебічним аналізом предметної області та детальним порівнянням існуючих аналогів, що обґрунтовує вибір платформи. Заслуговує на високу оцінку продумане архітектурне проектування, що включає вибір оптимальної моделі розгортання SaaS, розробку логічної моделі бази даних, а також якісну програмну реалізацію з використанням сучасних інструментів.

5. Негативні сторони роботи не реалізує інтеграцію системи з наявними рішеннями навчального закладу.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення та пояснювальна записка відповідають встановленим стандартам. Представлені діаграми та рисунки доповнюють тему роботи. Пояснювальна записка оформлена згідно з усіма чинними вимогами.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує на позитивну оцінку. Матеріал пояснювальної записки викладений структуровано, послідовно та зрозуміло, що дозволяє легко засвоїти тему проєктування. Графічний матеріал наочно демонструє деталі проєктування системи.

8. Інші зауваження _____

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі, відповідає поставленій задачі та заслуговує на оцінку «відмінно».

РЕЦЕНЗЕНТ Висеню Сергій Михайлович, доктор технічних наук, професор, заступник декана факультету інформатичної технології з комп'ютерної та мультимедійної роботи

“ 4 ” червня

2025 р.


(підпис)

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Система планування ресурсів навчального закладу на базі Microsoft Dynamics 365

Автор Кваснюк Денис Іванович

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

Рівень вищої освіти Перший (бакалаврський)

Спеціальність 121 «Інженерія програмного забезпечення»

Науковий керівник: Яшина Оксана Миколаївна, канд. тех. наук, доцент

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальнозживаних обов'язкових словосполучень у стандартних бланках (титулка, завдання, анотація, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій та у переліку джерел посилання;

2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/ схожості, складає 1,7%, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

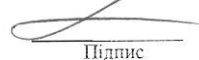
Дата 5.06.2025

Завідувач кафедри


Підпис

Леонід БЕДРАТЮК
Ім'я, ПРІЗВИЩЕ

Гарант освітньої програми


Підпис

Леонід БЕДРАТЮК
Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи


Підпис

Оксана ЯШИНА
Ім'я, ПРІЗВИЩЕ