

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Архітектура інформаційної системи освітнього процесу ЗВО»

Студента 2м курсу, 2 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
освітньої програми
«Інженерія програмного
забезпечення»

підпис студента

Шестак Ярослава
Івановича

Науковий керівник
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Харченко Олександр
Анатолійович

Гарант освітньої програми
кандидат педагогічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис гаранта

Котенко Наталія
Олексіївна

Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«16» листопада 2022 р.

Завдання

на випускний кваліфікаційний проєкт студентів

Шестаку Ярославу Івановичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Архітектура інформаційної системи освітнього процесу ЗВО»

Затверджена наказом ректора від «9» грудня 2022 р. № 3939

2. Строк здачі студентом закінченого проєкту 25 листопада 2023

3. Цільова установка та вихідні дані до проєкту

Метою є дослідження архітектури інформаційних систем ЗВО, особливостей їх використання й ефективності функціонування, та розробка програмно-апаратної архітектури інформаційної системи освітнього процесу ЗВО.

Об'єкт дослідження інформаційна система ЗВО.

Предмет дослідження є розробка моделі архітектури інформаційної системи освітнього процесу ЗВО, яка забезпечить ефективне функціонування та взаємодію різних компонентів інформаційної системи освітнього процесу ЗВО, покращить якість освіти та забезпечить ефективне управління навчальним процесом.

4. Консультанти проєкту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проєкту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. СУЧАСНІ ВИКЛИКИ ТА ТЕНДЕНЦІЇ В ОСВІТНЬОМУ СЕРЕДОВИЩІ

1.1. Аналіз освітнього процесу ЗВО

1.2. Аналіз попередніх досліджень

1.3. Функціональні характеристики інформаційної системи ЗВО

1.4. Висновки до розділу 1

РОЗДІЛ 2. ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОЇ АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Особливості проектування програмно-апаратної архітектури

2.2. Архітектура класів програмного додатку

2.3. Моделювання архітектури діаграми діяльності програмного додатку

2.4. Архітектура моделі компонентів програмного додатку

2.5. Проектування бази даних програмного додатку

2.6. Проектування архітектури апаратного забезпечення інформаційної системи

2.7. Обґрунтування вибору програмної платформи реалізації програмного додатку

2.8. Висновки до розділу 2

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ДОДАТКУ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Попередні приготування

3.3. Створення таблиць

3.4. Налаштування форм

3.5. Налаштування подань

3.6. Створення додатку на основі моделі

3.7. Налаштування ролей доступу

3.8. Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ТЕХНІЧНЕ ЗАВДАННЯ

ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

АНОТАЦІЯ

У ході дослідження проаналізовано сучасні виклики та тенденції в освітньому середовищі, описані функціональні характеристики інформаційної системи закладу вищої освіти. Спроектовано програмно-апаратну архітектуру інформаційної системи, яка складається з моделі архітектури класів програмного додатку, моделювання архітектури діаграми діяльності, архітектури моделі компонентів програмного додатку, концептуальної, логічної, фізичної моделей бази даних, архітектури апаратного забезпечення інформаційної системи.

За допомогою інструментальних засобів PowerApps був розроблений програмний додаток інформаційної системи закладу вищої освіти. Цей додаток повністю відповідає функціональним вимогам, що були визначені завданнями дослідження.

Ключові слова: інформаційна система, програмно-апаратна архітектура, моделювання архітектури, програмний додаток, PowerApps.

ABSTRACT

During the research, contemporary challenges and trends in the educational environment were analyzed, and the functional characteristics of the higher education institution's information system were described. The software-hardware architecture of the information system was designed, comprising the model of application class architecture, activity diagram architecture modeling, application component model architecture, conceptual, logical, and physical database models, as well as hardware architecture of the information system.

Using PowerApps tools, a software application for the higher education institution's information system was developed. This application fully meets the functional requirements set by the research tasks.

Keywords: information system, software-hardware architecture, architecture modeling, software application, PowerApps.



ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 СУЧАСНІ ВИКЛИКИ ТА ТЕНДЕНЦІЇ В ОСВІТНЬОМУ СЕРЕДОВИЩІ	6
1.1. Аналіз освітнього процесу ЗВО.....	6
1.2. Аналіз попередніх досліджень.....	12
1.3. Функціональні характеристики інформаційної системи ЗВО	13
1.4. Висновки до розділу 1	18
РОЗДІЛ 2 ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРATНОЇ АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ	19
2.1. Особливості проектування програмно-апаратної архітектури.....	19
2.2. Модель архітектури класів програмного додатку	21
2.3. Моделювання архітектури діаграми діяльності програмного додатку.....	30
2.4. Архітектура моделі компонентів програмного додатку.....	33
2.5. Проектування бази даних програмного додатку	36
2.6. Проектування архітектури апаратного забезпечення інформаційної системи	42
2.7. Обґрунтування вибору програмної платформи реалізації програмного додатку	49
2.8. Висновки до розділу 2	58
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ДОДАТКУ ІНФОРМАЦІЙНОЇ СИСТЕМИ	60
3.1. Попередні приготування	60
3.2. Створення таблиць	63
3.3. Налаштування форм	66
3.4. Налаштування подань	68
3.5. Створення додатку на основі моделі.....	70
3.6. Налаштування ролей доступу	72
3.7. Висновок до розділу 3	76
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	79
ТЕХНІЧНЕ ЗАВДАННЯ	82
ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ	85

<i>ДТЕУ 121 023-19.МР</i>				
Зм.	Аркуш	№ докум.	Підпис	Дата
Зав. каф.		Криворучко О.В.		
Керівник		Харченко О.А.		
Гарант		Котенко Н.О.		
Розробив		Шестак Я.І.		
Архітектура інформаційної системи освітнього процесу ЗВО			3	2
Зміст			Факультет інформаційних технологій 2м курс, 23 група	

ВСТУП

Актуальність. Впродовж 50 років інформатизувалися заклади вищої освіти, створювалися та впроваджувалися різні інформаційні системи, які виконували певні конкретні задачі. Надалі вони модифікувалися, трансформувалися, переходили з одних платформ на інші. Наразі практично всі освітні процеси охоплені окремими інформаційними системами у єдиному освітньому просторі, мають свої задачі та є потреба у їх детальному аналізі, оптимізації та уніфікації.

Сучасний освітній процес виходить за межі традиційних класних кімнат і стає все більш інтегрованим, глобальним та технологічним. За останні роки відбуваються значні зміни в освітній сфері, які стимулюють впровадження інформаційних систем для покращення ефективності та якості освіти.

Одним із основних факторів, що підвищують актуальність даної теми, є стрімкий розвиток інформаційних технологій та зростання їх впливу на всі сфери життя, включаючи освіту. Використання сучасних інформаційних технологій може суттєво поліпшити якість освіти, забезпечити доступ до нових навчальних ресурсів, сприяти індивідуалізації навчання та підвищенню мотивації студентів.

Крім того, в контексті сучасних викликів, таких як воєнні дії, пандемія COVID-19, дистанційне навчання та гнучкі форми навчання стають все більш поширеними. Використання інформаційних систем у таких ситуаціях стає надзвичайно важливим для забезпечення неперервності навчального процесу та забезпечення якості освіти.

Отже, дослідження архітектури інформаційної системи освітнього процесу ЗВО має велику актуальність у зв'язку з необхідністю впровадження

					<i>ДТЕУ 121 023-19.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.				Архітектура інформаційної системи освітнього процесу ЗВО	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник	Харченко О.А.					<i>В</i>	3	82
Гарант	Котенко Н.О.					Факультет інформаційних технологій 2м курс, 23 група		
Розробив	Шестак Я.І.							
					<i>Вступ</i>			

сучасних інформаційних технологій, покращенням ефективності та якості освіти, а також забезпеченням гнучкості та неперервності освітнього процесу.

Метою є дослідження архітектури інформаційних систем ЗВО, особливостей їх використання й ефективності функціонування, та розробка програмно-апаратної архітектури інформаційної системи освітнього процесу ЗВО.

Об'єктом дослідження – інформаційна система ЗВО.

Предмет дослідження є розробка моделі архітектури інформаційної системи освітнього процесу ЗВО, яка забезпечить ефективне функціонування та взаємодію різних компонентів інформаційної системи освітнього процесу ЗВО, покращить якість освіти та забезпечить ефективне управління навчальним процесом.

Завдання дослідження:

- проаналізувати сучасні виклики та тенденції освітнього середовища;
- описати функціональні характеристики інформаційної системи закладу вищої освіти;
- проаналізувати особливості проектування програмно-апаратної архітектури інформаційної системи ЗВО;
- змодельовати архітектуру класів архітектури, моделей компонентів програмного додатку, що визначатиме структуру програми, розподілення функцій між класами та їх взаємозв'язки;
- запроєктувати базу даних програмного додатку, визначаючи структуру та зв'язки між даними, для забезпечення ефективного зберігання та обробки інформації в системі освітнього процесу;
- обґрунтувати вибір програмної платформи для реалізації програмного додатку;

						Аркуш
					ДТЕУ 121 023-19.МР	4
Зм.	Аркуш	№ докум	Підпис	Дата		

- розробка програмного додатку.

Наукова новизна дослідження полягає у проектуванні архітектури моделей інформаційної системи, які спеціально адаптовані до потреб вищих закладів освіти та враховують актуальні виклики і потреби сучасного освітнього середовища.

Методи дослідження включали в себе аналіз літературних джерел, вивчення попередніх досліджень у сфері освітнього процесу та інформаційних систем вищої освіти, проектування програмно-апаратної архітектури, розробку програмного забезпечення, тестування та апробацію в реальних умовах.



									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР				5

РОЗДІЛ 1

СУЧАСНІ ВИКЛИКИ ТА ТЕНДЕНЦІЇ В ОСВІТНЬОМУ СЕРЕДОВИЩІ

1.1. Аналіз освітнього процесу ЗВО

Для аналізу освітнього процесу вищого закладу освіти (ЗВО) розглянемо безпосередній освітній процес ЗВО, а саме основні етапи та складові частини освітнього процесу. Освітній процес вищого закладу освіти (ЗВО) складається з ряду етапів та складових частин, які впливають на процес навчання та його результативність. Розглянемо кожен з них детальніше.

Підготовчий етап передбачає організаційні та адміністративні процеси, пов'язані з прийняттям здобувачів освіти, формуванням навчальних груп та розкладом занять. Наприклад, це може включати процедури заяв та вступу, розподіл студентів за спеціальностями та групами, планування навчального семестру тощо.

Навчальний етап є центральним у освітньому процесі, включає проведення лекцій, практичних занять, лабораторних робіт, семінарських занять тощо. Наприклад, студенти можуть відвідувати лекції, де викладач передає теоретичні знання та концепції, а після цього брати участь у практичних заняттях, де вони застосовують отримані знання на практиці.

Оцінювальний етап передбачає оцінку знань та навичок студентів. Це можуть бути письмові або практичні іспити, контрольні роботи, заліки, курсові проекти, дипломні роботи тощо. Наприклад, студенти можуть скласти письмові іспити або представляти результати своїх досліджень у формі курсового або дипломного проекту.

					<i>ДТЕУ 121 023-19.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.	Криворучко О.В.				Архітектура інформаційної системи освітнього процесу ЗВО	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник	Харченко О.А.					Р1	6	82
Гарант	Котенко Н.О.				<i>Сучасні виклики та тенденції в освітньому середовищі</i>	Факультет інформаційних технологій 2м курс, 23 група		
Розробив	Шестак Я.І.							

Сучасне освітнє середовище зазнає впливу різноманітних викликів та тенденцій, що впливають на освітній процес вищих навчальних закладів.

За останні роки інформаційні технології значно змінили підхід до навчання та викладання. Використання комп'ютерів, планшетів, онлайн-ресурсів та спеціалізованого програмного забезпечення дозволяє зробити навчання більш доступним, інтерактивним та ефективним.

Сучасні студенти мають різні потреби та можливості, тому гнучкість навчання є важливим аспектом. Це може включати варіативний вибір предметів, індивідуалізоване навчання, дистанційні форми навчання та можливість самостійно складати навчальний розклад.

Активне навчання та практична спрямованість освітнього процесу. Сучасні вимоги ринку праці вимагають випускників з практичними навичками. Тому в освітньому процесі все більше акцентується на практичних заняттях, проектній роботі, стажуваннях у компаніях та співпраці з роботодавцями.

В сучасному освітньому середовищі все більше акцентується на персоналізації навчання. Це означає, що навчальні програми та методики адаптуються до потреб та індивідуальних особливостей кожного студента. Наприклад, застосування індивідуального підходу до кожного студента, врахування його здібностей та інтересів, можливість вибору предметів або спеціалізацій.

У сучасному освітньому середовищі все більше значення надається міждисциплінарності та інтеграції знань. Замість розділеного на окремі дисципліни навчання, студенти виконують проекти або завдання, які охоплюють різні аспекти і напрямки знань. Це допомагає створити більш повне та цілісне розуміння предметної області та розвивати широкий спектр навичок.

					ДТЕУ 121 02з-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		7

Окрім професійних знань, сучасні випускники повинні мати розвинені навички Soft-skills, такі як комунікація, креативність, критичне мислення, проблемне мислення, керування проектами тощо. В освітньому процесі вищих закладів освіти дедалі більше уваги приділяється розвитку цих навичок через використання активних методів навчання, групову роботу, проектну діяльність тощо.

Однією з актуальних тенденцій є посилення контролю та забезпечення якості освіти. Вищі заклади освіти активно впроваджують системи оцінювання, моніторингу та звітності, щоб забезпечити високу якість навчання та відповідність вимогам ринку праці.

Ці приклади відображають сучасні виклики та тенденції в освітньому середовищі, які впливають на організацію та зміст освітнього процесу вищого навчального закладу. Розуміння цих викликів є важливим для розробки архітектури інформаційної системи, яка відповідатиме потребам сучасного освітнього процесу.

Сучасне освітнє середовище стикається з різноманітними викликами та тенденціями, що впливають на спосіб навчання, викладання та організацію освітнього процесу. Для вирішення сучасних викликів та тенденцій в освітньому середовищі, апаратно-програмно-комунікаційне рішення може бути розроблено з урахуванням наступних аспектів. Розглянемо кілька прикладів сучасних викликів та тенденцій в освітньому середовищі.

Інтеграція технологій у навчання. Завдяки швидкому розвитку технологій, освітнє середовище відкриває нові можливості для використання цифрових інструментів та ресурсів. Використання інтерактивних дошок, відеоуроків, онлайн-платформ та мобільних додатків дозволяє зробити навчання більш доступним, цікавим та ефективним. Для впровадження цифрових інструментів у освітній процес необхідна апаратна база, що

					ДТЕУ 121 02з-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

підтримує потрібні технології. Наприклад, використання віртуальної реальності в освіті дозволяє студентам отримати практичний досвід безпосередньо у віртуальній обстановці. Використання інтерактивних дошок потребує встановлення відповідного обладнання в аудиторіях. Крім того, необхідно забезпечити належну мережеву інфраструктуру для підключення до різних цифрових ресурсів та платформ.

Розвиток е-лернінгу та дистанційного навчання. Зростання популярності дистанційного навчання та е-лернінгу є однією з найважливіших тенденцій в освітньому середовищі. Для реалізації дистанційного навчання необхідно обрати відповідну платформу, яка забезпечить можливість завантаження навчальних матеріалів, проведення онлайн-курсів та взаємодії між викладачами та студентами. Це дозволяє студентам отримувати доступ до навчальних матеріалів та занять з будь-якого місця та у зручний для них час. Прикладами таких платформ є Moodle, Blackboard, Microsoft Teams, Google Classroom тощо. Для цього також потрібне апаратне забезпечення, яке забезпечить стабільну роботу платформи та підтримку багатьох користувачів одночасно. Наприклад, використання платформи Moodle дозволяє університетам створювати онлайн-курси з різних предметів та проводити віртуальні заняття для студентів.

Розвиток активного навчання. Традиційний пасивний підхід до навчання поступово замінюється активним навчанням, де студенти беруть активну участь у власному навчанні та розвитку. Групові проекти, практичні завдання, рольові ігри, дискусії та інтерактивні методи навчання стимулюють самостійність, критичне мислення та співпрацю студентів. Наприклад, метод проблемного навчання дозволяє студентам активно вирішувати реальні проблеми та задачі, що стимулює їх творчість та аналітичні навички.

					<i>ДТЕУ 121 02з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		9

Забезпечення доступу до інформації. У зв'язку з розширенням джерел інформації, важливо мати систему, яка забезпечить швидкий та зручний доступ до цифрових ресурсів. Наприклад, це може бути використання хмарних технологій для збереження та розподілу навчальних матеріалів, що дозволить студентам та викладачам отримувати доступ до них з будь-якого пристрою та місця.

Розвиток компетентнісного навчання. Компетентнісний підхід в освіті передбачає формування у студентів певних навичок, знань та вмінь, які дозволять їм успішно функціонувати у сучасному суспільстві. Наприклад, розвиток комунікативних навичок, критичного мислення, творчості та проектного менеджменту є важливими компетенціями, які сприяють успішному професійному розвитку студентів.

Відеоконференції та співпраця в режимі реального часу, оскільки дистанційне навчання стає все більш популярним, важливо мати можливість проводити відеоконференції та співпрацювати в режимі реального часу. Інструменти, такі як Zoom, Microsoft Teams, Google Meet, дозволяють студентам та викладачам взаємодіяти, обговорювати матеріали, задавати запитання та проводити заняття в онлайн-форматі.

Моніторинг та оцінювання. Важливим аспектом є система моніторингу та оцінювання освітнього процесу. Застосування спеціалізованих програмних засобів дозволяє викладачам вести електронний журнал, створювати онлайн-тести, оцінювати студентів та надавати зворотний зв'язок. Наприклад, системи керування навчанням (LMS) дозволяють виконувати ці функції.

Ці апаратно-програмно-комунікаційні рішення сприяють впровадженню сучасних викликів та тенденцій в освітньому середовищі, забезпечуючи зручний доступ до навчальних матеріалів, співпрацю та комунікацію в режимі реального часу, ефективне моніторингове та

					<i>ДТЕУ 121 02з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		10

оцінювальне середовище. Враховуючи специфіку освітнього процесу вищого навчального закладу, таке рішення допоможе забезпечити якісне та ефективне навчання студентів.

Переваги впровадження нової архітектури інформаційної системи:

1. Впровадження нової архітектури може призвести до покращення продуктивності інформаційної системи, зменшення часу відгуку та оптимізації роботи процесів. Наприклад, перехід від старої серверної архітектури до розподіленої обчислювальної системи може забезпечити більш швидкий доступ до даних та підвищення продуктивності виконання завдань.

2. Нова архітектура може забезпечити збільшення масштабованості системи, що дозволить впоратися зі зростаючою кількістю користувачів та обсягом даних. Наприклад, перехід до хмарних технологій може дозволити легко розширювати обчислювальні ресурси в залежності від потреб системи.

3. Дана архітектура може привести до підвищення надійності та безпеки інформаційної системи. Наприклад, використання резервних систем забезпечить неперервну роботу системи навіть у випадку відмови окремих компонентів. Крім того, впровадження сучасних методів шифрування та автентифікації може забезпечити високий рівень безпеки даних.

4. Архітектура може допомогти зменшити витрати на обслуговування та розвиток інформаційної системи. Наприклад, використання віртуалізації дозволяє економити фізичні сервери та енергію, а використання відкритих джерел може знизити витрати на ліцензування програмного забезпечення.

Покращений доступ та зручність використання, що може сприяти поліпшенню доступу до інформації та забезпеченню зручного інтерфейсу для користувачів. Наприклад, розробка мобільних додатків або веб-інтерфейсів може дати змогу студентам та викладачам отримувати доступ до навчальних матеріалів з будь-якого пристрою та в будь-який час.

					ДТЕУ 121 02з-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		11

Ці приклади показують, що впровадження нової архітектури інформаційної системи може мати численні потенційні переваги для освітнього середовища. Застосування сучасних технологій та розробка ефективної архітектури дозволять покращити продуктивність, масштабованість, надійність, безпеку, знизити витрати та полегшити доступ до інформації.

1.2. Аналіз попередніх досліджень

Дослідженню особливостей використання архітектури інформаційної системи освітнього процесу ЗВО вивчались у попередніх дослідженнях. Розглянемо основні публікації.

Автор Іванова О.М. в статті: «Проектування інформаційних систем в освітніх закладах», опублікованої в журналі «Наукові праці вищого навчального закладу» (рік публікації: 2020) провела дослідження щодо методологій, підходів та технологій проектування інформаційних систем для освітніх закладів. Виявлено різноманітні архітектурні концепції та моделі, які використовуються для покращення ефективності та якості освітнього процесу.

В статті «Інтеграція систем управління навчальними програмами та електронних бібліотек в освітній інформаційній системі», авторів Петрова В.С., Сидоренка Н.А.. в дослідженні зосереджувалися на проблемах інтеграції різних інформаційних систем, таких як системи електронного навчання, системи управління навчальними програмами та системи електронних бібліотек. Розглядалися різні підходи до створення єдиної інтегрованої інформаційної системи, що сприяє поліпшенню комунікації та обміну даними між різними компонентами освітнього процесу.

					ДТЕУ 121 02з-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		12

Вчена Семенова Л.В. в статті «Використання хмарних технологій у системах освіти» акцентувала увагу на дослідженні та впровадженні сучасних технологій у системи освіти, таких як хмарні обчислення, інтернет речей (IoT), штучний інтелект (AI) та аналітика даних. Виявлено переваги та виклики використання цих технологій для покращення процесів навчання, адаптації програм та підтримки індивідуалізації навчання.

В статті «Забезпечення безпеки інформаційних систем освітнього процесу», авторів Коваленко М.І., Ткаченко О.П., зверталась увага на важливість захисту даних, конфіденційності і безпеки інформаційних систем освітнього процесу. Аналізувалися методи і технології, які допомагають забезпечити захист важливих даних, виявлення та запобігання кібератакам та іншим загрозам інформаційної безпеки.

Враховуючи попередні дослідження, можна зазначити, що тема «Архітектура інформаційної системи освітнього процесу ЗВО» має актуальність у сфері освіти. Проте, існує потреба у подальшому дослідженні та розробці оптимальних архітектурних рішень, зокрема у контексті використання сучасних технологій та забезпечення безпеки інформації в системі освіти.

1.3. Функціональні характеристики інформаційної системи ЗВО

Інформаційна система ЗВО має багато складових, які постійно удосконалюються, розширюються їх можливості та трансформуються в залежності від потреб суспільства. В процесі цифровізації держави всі процеси автоматизуються та переводяться у цифровий формат, інформація

					ДТЕУ 121 02з-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		13

переноситься у ієрархічні та розподілені бази даних. Для ефективного використання користувачами (викладачами, адміністраторами та здобувачами вищої освіти) інформації, яка вже накопичена, зберігається і надалі акумулюється, необхідно правильно розподілити ресурси інформаційних систем. Тому, за необхідності, слід побудувати зрозумілі, відкриті і доступні інтерфейси обміну інформацією за допомогою API (спеціальні проміжні таблиці (view), способи підключень та організація прав доступу, читання, коригування чи знищення інформації). Важливим є організація кібернетичного захисту окремих частин та в цілому інформаційної системи освітнього процесу. Архітектура інформаційної системи має передбачати способи комунікації з зовнішніми інформаційними системами смарт-міста, що дозволить полегшити отримання коректної, вивіреної інформації з її юридичними підтвердженнями, завіреної електронними підписами тощо [2].

Автором запропонована архітектура інформаційної системи освітнього процесу ЗВО (зазначена на рис 1.1). Вона містить ряд окремих автономних інформаційних систем:

- Система освітньої діяльності ЗВО – це інформаційна система, яка виконує функції обміну інформації між всіма користувачами інфраструктури ЗВО, визначає правила та типи користувачів, в залежності від статусу: викладач, адміністратор чи здобувач – отримує певну структуровану інформацію в залежності від запиту, потреби: розклад, навантаження, тип заняття, аудиторія з певними матеріально-технічними засобами навчання, організація вибору індивідуальної траєкторії навчання, організацію заліків, іспитів, консультацій, ознайомлення з результатами сесії, формування результуючих даних у розрізі студента, групи, груп кафедри, груп факультету, потоку чи в цілому всіх студентів ЗВО [5]. Аналіз результатів успішності, кількісних

					ДТЕУ 121 02з-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		14

та якісних характеристик відвідувань занять. Система включає дані щодо переведення здобувача освіти, закінчення навчання чи відрахування за неуспішність із доведенням інформації керівництву закладу засобами інформаційної системи.

- Система дистанційного навчання ЗВО – це інформаційна система, яка організовує сам процес навчання здобувачів вищої освіти в якій вони мають змогу отримувати необхідну для освіти інформацію, консультації викладачів, спілкуватись з колегами по навчанню в зручний для себе час. В системі дистанційного навчання розміщені всі необхідні методичні матеріали, електронні підручники, презентації, програми, робочі програми, методичні рекомендації для самостійної роботи здобувача вищої освіти.
- Конференц-системи для змішаного навчання, VR-системи – використовуються для розширення можливостей надання освітніх послуг та охоплення великої кількості слухачів, здобувачів та учасників конференцій і інших наукових заходів, які дають можливість очного і дистанційного навчання, чи демонстрування презентацій. Також широко почали використовувати в освітньому процесі технології віртуальної реальності VR-системи, у яких є можливість досліджувати певні процеси у віртуальному просторі.
- Система фінансово-економічної діяльності ЗВО – дає можливість у цифровому вигляді вести повний фінансовий облік матеріальних цінностей, облік всіх фінансових операцій, контролювати інформацію про фінансовий стан тощо. Система фінансово-економічної діяльності надає повну інформацію щодо прийняття управлінських рішень.
- Система кібернетичної безпеки – це система, яка забезпечує безпеку всіх інформаційних систем ЗВО, відповідно до побудованої архітектури [4].

					<i>ДТЕУ 121 02з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		15

- Системи керування серверних ресурсів – в залежності від потреби здобувачів вищої освіти, викладачів організовується розподіл ресурсів серверного обладнання, з урахуванням побудови постійного дублювання інформації, використання хмарних ресурсів – підписок Office 365(компонентів), скарб-освіти тощо – адміністратори в залежності від політики розмежування прав надають доступ до даних ресурсів.
- Система керування мережею, VPN ЗВО – для забезпечення ефективної роботи у комп’ютерній мережі, як інформаційній системі постійно ведеться моніторинг її стану, вживаються політики безпеки на різних рівнях на комутаційному керованому обладнанні за допомогою мережного програмного забезпечення. В період воєнного стану заборонено використовувати відкриті способи доступу до ресурсів елементів мережі, тільки з використанням VPN з генерацією персонального ключа користувача, для автентифікації та не допущення організаційного порушення захисту, в залежності від активності зовнішніх користувачів система моніторингу автоматично може заблокувати для забезпечення безпеки архітектури інформаційної системи ЗВО.

					<i>ДТЕУ 121 02з-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		16

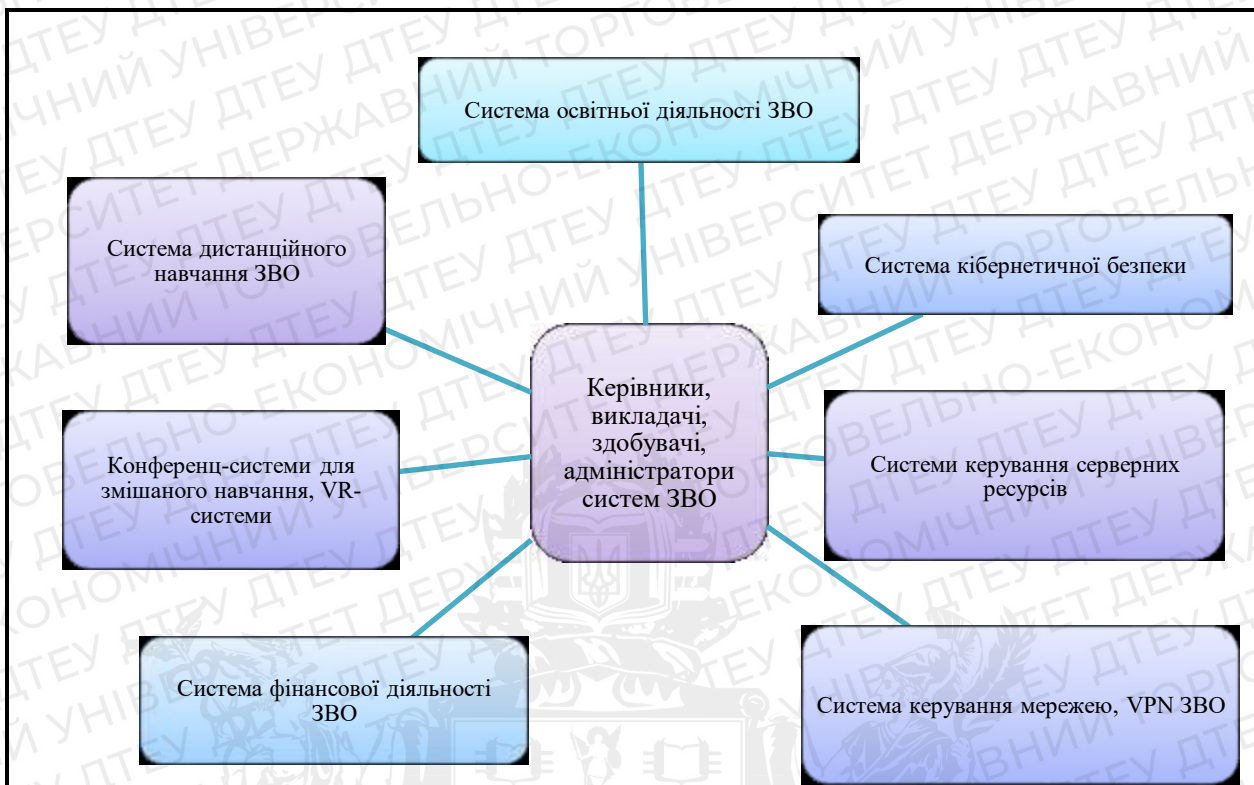


Рис 1.1. Загальна архітектура інформаційної системи освітнього процесу ЗВО

Така архітектура, на нашу думку, дасть можливість повного контролю інформації, можливість відповідної побудови захисту, що наразі є актуальним в період воєнного стану в Україні. Також вона дасть можливість правильно побудувати та ефективно використати інформаційні ресурси ЗВО, дозволить оптимізувати потоки інформації, підвищить надійність такої системи [3]. В процесі експлуатації це дасть можливість у повній мірі використати гібридні інформаційні системи, поєднувати роботу у локальній мережі та використати хмарні технології, а при потребі безпечно працювати віддалено у межах інформаційної системи ЗВО. Веб ресурси ЗВО необхідно розподілити на закриті і відкриті, для правильної організації освітнього процесу.

					ДТЕУ 121 02з-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		17

1.4. Висновки до розділу 1

У першому розділі було проведено аналіз сучасних викликів та тенденцій в освітньому середовищі закладу вищої освіти. Швидкий технологічний розвиток, динамічні зміни в суспільстві та поглиблена глобалізація вимагають перегляду та адаптації традиційних підходів до навчання. Виявлено, що навчальний процес складається з різних етапів та компонентів, які взаємодіють для досягнення мети, а саме навчання та підготовці висококваліфікованих фахівців. Однак сучасні виклики, такі як інтеграція технологій, дистанційне навчання, активне навчання та розвиток компетентностей, досягають системи на більш динамічний та інтерактивний режим. Перехід до нової архітектури може призвести до численних переваг, таких як покращення якості та ефективності навчального процесу, оптимізація управління закладом, підвищення зручності користувачів.

Відповідно до цього, розробка і впровадження нової інформаційної системи має потенціал стати вагомим кроком до адаптації освітнього процесу до сучасних вимог та створення сприятливого навчального середовища для всіх учасників. Розуміння цих викликів та тенденцій є важливим для розробки архітектури інформаційної системи, яка відповідатиме потребам сучасного освітнього процесу.

Правильно побудувавши інформаційну систему освітнього процесу ЗВО отримується модель системи, де можна прогнозувати та передбачати стійкість системи. Важливим місцем у даній системі є співвідношення всіх окремих інформаційних систем, які постійно розвиваються, трансформуються, тому можна контролювати процеси підготовки і впровадження оновлень. Архітектура дає можливість побудувати інформаційну систему, яка покращує ефективність управління закладом вищої освіти та безпосередньо освітнім процесом.

					ДТЕУ 121 02з-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		18

РОЗДІЛ 2

ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРATНОЇ АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Особливості проектування програмно-апаратної архітектури

Архітектура системи – це опис взаємозв'язків між компонентами системи, їх функцій та поведінки.

Для проектування архітектури системи освітнього процесу ЗВО необхідно виконати наступні завдання:

1. Визначити склад компонентів програмного додатку інформаційної системи та їх взаємозв'язки (змоделювати архітектуру класів).
2. Визначити алгоритми роботи інформаційної системи та процеси, які вона повинна виконувати.
3. Розробити архітектурні діаграми, що відображають взаємозв'язки та залежності між компонентами системи.
4. Розробити концептуальну, логічну та фізичну моделі бази даних та спосіб її організації.
5. Визначити апаратні технології та програмні інструменти, які будуть використовуватися для розробки та реалізації інформаційної системи.

Проектування архітектури системи дозволяє створити загальне уявлення про будову та роботу системи, що є необхідною передумовою для її розробки та імплементації.

Проектування програмної архітектури інформаційної системи є важливим етапом в розробці системи, де визначається структура, організація та взаємодія компонентів системи для досягнення поставлених цілей.

					<i>ДТЕУ 121 023-19.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.	Криворучко О.В.				Архітектура інформаційної системи освітнього процесу ЗВО	Стадія	Аркуш	Аркушів
Керівник	Харченко О.А.					P2	19	82
Гарант	Котенко Н.О.					Факультет інформаційних технологій		
Розробив	Шестак Я.І.				Проектування програмно-апаратної архітектури інформаційної системи	2м курс, 2з група		

Основною метою проектування архітектури є створення ефективної, масштабованої, надійної та підтримуваної системи.

Важливо зазначити, що проектування програмної архітектури є ітеративним процесом, який може вимагати змін і удосконалень на різних етапах розробки системи. Деталізація кожного пункту може варіюватися залежно від конкретного проекту та його вимог.

Проектування апаратної архітектури інформаційної системи включає в себе визначення фізичної інфраструктури, компонентів апаратного забезпечення та їх організацію для забезпечення працездатності та ефективності системи.

Для цього спочатку характеризують вимоги до апаратної архітектури. На цьому етапі проводиться аналіз вимог до системи, зокрема вимог до продуктивності, надійності, масштабованості, безпеки та інших параметрів. Враховуються такі фактори, як очікувана кількість користувачів, обсяг обробки даних, необхідність резервного копіювання тощо.

Наступним етапом буде вибір апаратного забезпечення. Залежно від вимог, проводиться вибір апаратних компонентів, таких як сервери, роутери, комутатори, мережеве обладнання, сховища даних, резервне копіювання тощо. Враховуються такі параметри, як потужність, швидкість, масштабованість, надійність та сумісність зі забезпеченням програмного забезпечення.

Подальшим кроком буде організація фізичної інфраструктури, а саме визначається розміщення апаратного забезпечення, його зв'язок та конфігурація; враховуються такі фактори, як фізична безпека, доступність, сегментування мережі, використання віртуалізації та інші фактори.

Далі визначається логічна структура мережі, включаючи маршрутизацію, підключення, сегментування, захист мережі, комутацію та інші параметри. Розробляється детальний план мережі, включаючи вибір

						ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			20

мережевих пристроїв, розподіл IP-адрес, налаштування файрвола та інші налаштування мережі.

Останнім кроком буде забезпечення безпеки апаратної архітектури. На цьому етапі враховуються такі фактори, як захист мережі, використання файрволів, шифрування даних, виявлення та запобігання вторгненням, резервне копіювання та відновлення даних тощо. Далі проводиться тестування апаратної архітектури для перевірки його працездатності, надійності, продуктивності та відповідності вимогам. Виконуються навантажувальні тести, тести на витривалість, тести на безпеку та інші тести.

Ці кроки зазвичай виконуються в ітеративному циклі з іншими етапами розробки інформаційної системи, такими як аналіз вимог, проектування програмної архітектури, розробка, тестування та впровадження. Деталізація кожного кроку може варіюватися залежно від конкретного проекту та його вимог.

2.2. Модель архітектури класів програмного додатку

Діаграма класів орієнтована на об'єктно-орієнтовану модель програмування і відображає інформаційну модель у вигляді системи класів, які логічно взаємопов'язані між собою.

Кожен клас має свої методи, які контролюють дії об'єкта класу і свої визначені параметри, що зберігаються у змінних кожного об'єкта чи класу. Опишімо детально кожний клас, що проектується.

Клас «Співробітник» (Employee), містить наступні атрибути:

- ✓ id: int (ідентифікатор співробітника).
- ✓ firstName: string (ім'я співробітника).
- ✓ lastName: string (прізвище співробітника).
- ✓ middleName: string (по батькові співробітника).

						Аркуш
					ДТЕУ 121 023-19.МР	21
Зм.	Аркуш	№ докум	Підпис	Дата		

- ✓ position: string (посада співробітника).
Клас включає в себе такі методи:
 - ✓ GetEmployeeById(id: int): Employee (метод для отримання співробітника за ідентифікатором).
 - ✓ GetAllEmployees(): List<Employee> (метод для отримання списку всіх співробітників).
Клас «Аудиторія» (Audience) включає в себе такі атрибути:
 - ✓ id: int – унікальний ідентифікатор аудиторії.
 - ✓ name: string – назва аудиторії.
 - ✓ purpose: string – призначення аудиторії.
 - ✓ capacity: int – вмістимість аудиторії.
 - ✓ equipment: string[] – перелік обладнання в аудиторії.
 - ✓ location: string - розташування аудиторії.
 - ✓ availability: boolean – прапорець, що вказує на доступність аудиторії.
 - ✓ schedule: Schedule[] – масив розкладу використання аудиторії.
 - ✓ status: string – поточний стан аудиторії.
- Даний клас володіє наступними методами:
- ✓ getID(): int – повертає ідентифікатор аудиторії.
 - ✓ getName(): string – повертає назву аудиторії.
 - ✓ setName(name: string): void – встановлює нову назву аудиторії.
 - ✓ getPurpose(): string – повертає призначення аудиторії.
 - ✓ setPurpose(purpose: string): void – встановлює нове призначення аудиторії.
 - ✓ getCapacity(): int – повертає вмістимість аудиторії.
 - ✓ setCapacity(capacity: int): void – встановлює нову вмістимість аудиторії.
 - ✓ getEquipment(): string[] – повертає масив обладнання в аудиторії.
 - ✓ setEquipment(equipment: string[]): void – встановлює новий масив обладнання в аудиторії.

					ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		22

- ✓ getLocation(): string – повертає розташування аудиторії.
- ✓ setLocation(location: string): void – встановлює нове розташування аудиторії.
- ✓ isAvailable(): boolean – перевіряє доступність аудиторії.
- ✓ setAvailability(availability: boolean): void – встановлює новий статус доступності аудиторії.
- ✓ getSchedule(): Schedule[] – повертає масив розкладу використання аудиторії.
- ✓ addSchedule(schedule: Schedule): void – додає новий об'єкт розкладу використання аудиторії.
- ✓ removeSchedule(schedule: Schedule): void – видаляє об'єкт розкладу використання аудиторії.
- ✓ getStatus(): string – повертає поточний стан аудиторії.
- ✓ setStatus(status: string): void – встановлює новий стан аудиторії.

Клас «Ремонт обладнання» (EquipmentRepair) має наступні атрибути:

- ✓ id: int (ідентифікатор ремонту).
- ✓ equipmentId: int (ідентифікатор обладнання).
- ✓ employeeId: int (ідентифікатор співробітника).
- ✓ problemDescription: string (опис проблеми).
- ✓ repairDescription: string (опис виправлення проблеми).

Даний клас містить такі методи:

- ✓ GetEquipmentRepairById(id: int): EquipmentRepair – метод для отримання ремонту обладнання за ідентифікатором.
- ✓ GetAllEquipmentRepairs(): List<EquipmentRepair> – метод для отримання списку всіх ремонтів обладнання.
- ✓ GetEquipmentRepairsByEquipmentId (equipmentId: int): List<EquipmentRepair> – метод для отримання списку всіх ремонтів обладнання за ідентифікатором обладнання.

					ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		23

✓ `GetEquipmentRepairsByEmployeeI (employeeId: int): List<EquipmentRepair>` – метод для отримання списку всіх ремонтів обладнання за ідентифікатором співробітника.

Клас «Замовлення на ремонт обладнання» (`EquipmentRepairOrders`) містить атрибути:

- ✓ `ApplicationID: int` (ідентифікатор заявки).
- ✓ `EquipmentIdentifier: int` (ідентифікатор обладнання, що потребує ремонту).
- ✓ `ProblemDescription: string` (опис проблеми).
- ✓ `ApplicationStatus: enum` (статус заявки: Нова, В обробці, Виконана).

Даний клас володіє такими методами:

- ✓ `Designer: (Вхідні дані: ApplicationID, EquipmentIdentifier, ProblemDescription)` – ініціалізує новий екземпляр класу «Замовлення на ремонт обладнання» з вказаними атрибутами).
- ✓ `Set Application Status: ApplicationStatus` (значення з переліку) – встановлює статус заявки на вказане значення).
- ✓ `GetApplicationInformation: Вихідні дані: ApplicationID, EquipmentIdentifier, ProblemDescription, ApplicationStatus` – отримує та повертає інформацію про заявку на ремонт, включаючи її ідентифікатор, ідентифікатор обладнання, опис проблеми та статус заявки).
- ✓ `EditTheRequest: (Вхідні дані: UpdatedProblemDescription (строка)` – (змінює опис проблеми заявки на ремонт на наданий оновлений опис).

Ці методи дозволяють здійснювати маніпуляції та отримувати інформацію, пов'язану з заявкою на ремонт. Конструктор ініціалізує новий об'єкт заявки з вказаними атрибутами. Метод `Set Application Status` дозволяє змінювати статус заявки. Метод `GetApplicationInformation` повертає всю необхідну інформацію про заявку. Нарешті, метод `EditTheRequest` дозволяє змінювати опис проблеми в заявці.

						ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			24

Клас «Програмне забезпечення» (SoftwareI) містить такі атрибути:

- ✓ SoftwareID: int (ідентифікатор програмного забезпечення).
- ✓ SoftwareName: string (назва програмного забезпечення).
- ✓ SoftwareVersion: string (версія програмного забезпечення).
- ✓ LicenseType: enum (тип ліцензії: одноразова, річна, постійна).
- ✓ LicensePrice: decimal (ціна ліцензії).

Методи даного класу:

- ✓ Constructor: вхідні дані: SoftwareID, SoftwareName, SoftwareVersion, LicenseType, LicensePrice – ініціалізує новий екземпляр класу «Програмне забезпечення» з вказаними атрибутами.
- ✓ GetSoftwareInformation: вихідні дані: SoftwareID, SoftwareName, SoftwareVersion, LicenseType, LicensePrice – повертає інформацію про програмне забезпечення, включаючи його ідентифікатор, назву, версію, тип ліцензії та ціну ліцензії.

Ці атрибути і методи дозволяють отримувати інформацію про програмне забезпечення. Конструктор ініціалізує новий об'єкт програмного забезпечення з вказаними атрибутами. Метод GetSoftwareInformation повертає всю необхідну інформацію про програмне забезпечення, включаючи його ідентифікатор, назву, версію, тип ліцензії та ціну ліцензії.

Клас «Устаткування» (Equipment) включає в себе наступні атрибути:

- ✓ EquipmentID: int (ідентифікатор устаткування).
- ✓ EquipmentName: string (назва устаткування).
- ✓ EquipmentType: enum (тип устаткування: комп'ютер, принтер, сканер, інше).
- ✓ SerialNumber: string (серійний номер).
- ✓ PurchaseDate: DateTime (дата придбання).
- ✓ Cost: decimal (вартість).

Методи класу:

						ДТЕУ 121 023-19.МР	Аркуш
							25
Зм.	Аркуш	№ докум	Підпис	Дата			

✓ Constructor: вхідні дані: EquipmentID, EquipmentName, EquipmentType, SerialNumber, PurchaseDate, Cost – ініціалізує новий екземпляр класу «Устаткування» з вказаними атрибутами.

✓ GetEquipmentInformation: вихідні дані: EquipmentID, EquipmentName, EquipmentType, SerialNumber, PurchaseDate, Cost – повертає інформацію про устаткування, включаючи його ідентифікатор, назву, тип, серійний номер, дату придбання та вартість.

Вказані атрибути і методи дозволяють отримувати інформацію про устаткування. Конструктор ініціалізує новий об'єкт устаткування з вказаними атрибутами. Метод GetEquipmentInformation повертає всю необхідну інформацію про устаткування, включаючи його ідентифікатор, назву, тип, серійний номер, дату придбання та вартість.

Клас «Завкафедри» (HeadOfDepartment) має такі атрибути:

- ✓ LastName: string (прізвище).
- ✓ FirstName: string (ім'я).
- ✓ Patronymic: string (по батькові).
- ✓ Email: string (електронна адреса).
- ✓ Phone: string (телефон).

Методи даного класу:

- ✓ Constructor: вхідні дані: LastName, FirstName, Patronymic, Email, Phone – ініціалізує новий екземпляр класу «Завкафедри» з вказаними атрибутами.
- ✓ GetHeadInformation: вихідні дані: LastName, FirstName, Patronymic, Email, Phone – повертає інформацію про завідувача кафедри, включаючи його прізвище, ім'я, по батькові, електронну адресу та телефон.

Ці атрибути і методи дозволяють отримувати інформацію про завідувача кафедри. Конструктор ініціалізує новий об'єкт завкафедри з вказаними атрибутами. Метод GetHeadInformation повертає всю необхідну інформацію

						Аркуш
					ДТЕУ 121 023-19.МР	26
Зм.	Аркуш	№ докум	Підпис	Дата		

про завідувача, включаючи його прізвище, ім'я, по батькові, електронну адресу та телефон.

Клас «Мережа» (Network) має такі атрибути:

- ✓ NetworkName (тип: рядок) – ім'я мережі.
- ✓ Devices (тип: список пристроїв) – список пристроїв, підключених до мережі.
- ✓ Connections (тип: список з'єднань) – список з'єднань між пристроями в мережі.

Клас «Мережа» має такі методи:

- ✓ SetNetworkName(name: рядок) – встановлює ім'я мережі.
- ✓ GetNetworkName(): рядок – повертає ім'я мережі.
- ✓ AddDevice(device) – додає пристрій до мережі.
- ✓ RemoveDevice(device) – видаляє пристрій з мережі.
- ✓ ConnectDevices(device1, device2, deviceN) – встановлює з'єднання між двома пристроями.
- ✓ DisconnectDevices(device1, device2, deviceN) – розриває з'єднання між двома пристроями.
- ✓ GetConnectedDevices(device): список пристроїв – повертає список пристроїв, з'єднаних з вказаним пристроєм.
- ✓ GetAllDevices(): список пристроїв – повертає список усіх пристроїв у мережі.
- ✓ GetAllConnections(): список з'єднань – повертає список усіх з'єднань в мережі.

Клас «Відділ ЦІТ « DepartmentInformationTechnologies Center має наступні атрибути:

- ✓ ContactPerson: string (контактна особа - ім'я та прізвище контактної особи ЦІТ).
- ✓ Phone: string (телефон контактної особи ЦІТ).

						Аркуш
					ДТЕУ 121 023-19.МР	27
Зм.	Аркуш	№ докум	Підпис	Дата		

- ✓ Email: string (електронна пошта контактної особи ЦІТ).
- Методи класу:
- ✓ AddRepair: вхідні дані: repairInfo (інформація про ремонт обладнання) :
Додає новий запис про ремонт обладнання в базу даних.
- ✓ DeleteRepair: вхідні дані: repairID (ідентифікатор ремонту) – видаляє запис про ремонт обладнання з бази даних.
- ✓ GetRepairsList: вихідні дані: список записів про ремонт обладнання – повертає список всіх записів про ремонт обладнання з бази даних.
- ✓ ChangeRepairStatus: вхідні дані: repairID (ідентифікатор ремонту), newStatus (новий статус ремонту) – змінює статус ремонту обладнання в базі даних на новий статус.
- ✓ AddSoftware: Вхідні дані: softwareInfo (інформація про нове програмне забезпечення) – новий запис про програмне забезпечення в базу даних.
- ✓ DeleteSoftware: вхідні дані: softwareID (ідентифікатор програмного забезпечення) – видаляє запис про програмне забезпечення з бази даних.
- ✓ GetSoftwareList: вихідні дані: список записів про програмне забезпечення – повертає список всіх записів про програмне забезпечення з бази даних.
- ✓ AddEquipment: Вхідні дані: equipmentInfo (інформація про нове обладнання) – додає новий запис про обладнання в базу даних.
- ✓ DeleteEquipment: вхідні дані: equipmentID (ідентифікатор обладнання) – видаляє запис про обладнання з бази даних.
- ✓ GetEquipmentList: вихідні дані: список записів про обладнання – повертає список всіх записів про обладнання з бази даних.
- ✓ AddUser: вхідні дані: userInfo (інформація про нового користувача) – додає новий запис про користувача в базу даних.
- ✓ DeleteUser: вхідні дані: userID (ідентифікатор користувача) – видаляє запис про користувача з бази даних.

					ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		28

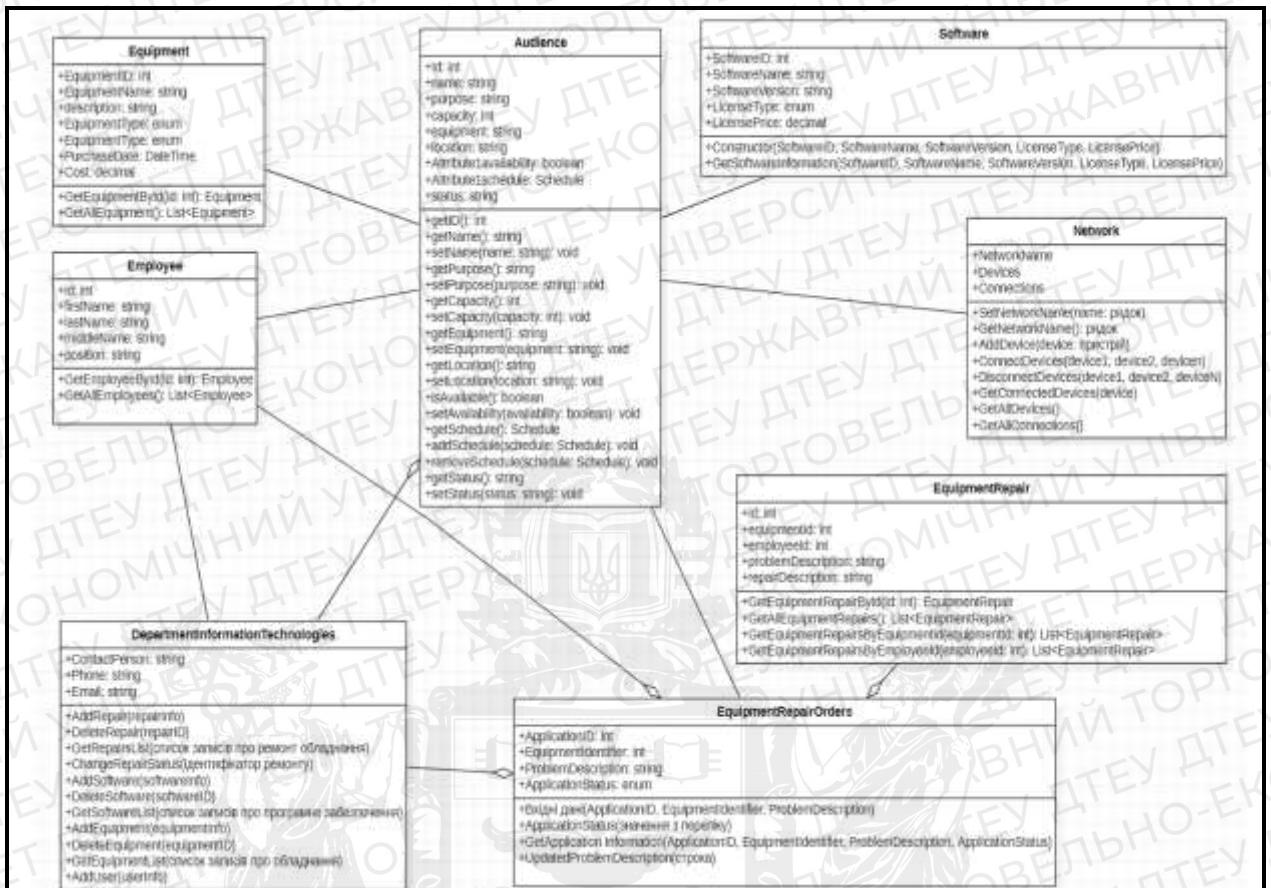


Рис. 2.1. Модель архітектури класів програмного додатку

Джерело: побудовано автором

Взаємозв'язки між класами можуть бути різними в залежності від того, які функції системи повинні бути підтримані та які задачі повинні вирішуватися. Однак, на основі моделі, яку ми створили, можна виділити декілька можливих взаємозв'язків:

Клас «Відділ ЦІТ» може мати агрегаційний взаємозв'язок з класом «Ремонтне замовлення», оскільки ремонтний центр може мати багато ремонтних замовлень, але вони не є його частинами і можуть існувати самостійно.

Клас «Ремонтне замовлення» може мати композиційний взаємозв'язок з класом «Клієнт», оскільки кожне ремонтне замовлення повинно бути пов'язане з конкретним клієнтом і не може існувати без нього.

Клас «Замовлення на ремонт обладнання» може мати агрегаційний взаємозв'язок з класом «Устаткування», оскільки кожне ремонтне замовлення може стосуватися конкретного обладнання, але воно не є його частиною.

Клас «Устаткування» може мати асоціативний взаємозв'язок з класом «Програмне забезпечення», оскільки обладнання може мати багато програмного забезпечення, а програмне забезпечення може використовуватися на різному обладнанні.

Клас «Працівник» може мати агрегаційний взаємозв'язок з класом «Ремонтне замовлення», оскільки кожен технік може мати багато ремонтних замовлень, але вони не є його частинами і можуть існувати самостійно.

Ці взаємозв'язки можуть допомогти в розробці відповідної архітектури системи та визначенні способу взаємодії між класами.

2.3. Моделювання архітектури діаграми діяльності програмного додатку

Для моделювання архітектури діаграми діяльності програмного додатку будемо використовувати UML-модель діяльності, яка включає в себе діаграми, що ілюструють послідовність дій, рішень та активностей, які відбуваються в рамках процесу розробки програмного додатку. Діаграми діяльності можуть містити дії, стани, рішення, вітки виконання, паралельні процеси та зв'язки між ними.

Опис UML-моделі діяльності для відділу Центру інформаційних технологій, як ключового компонента побудови інформаційної системи освітнього процесу, може включати основні елементи:

1. Процес перевірки ідентифікації та авторизації учасників освітнього процесу ЗВО. При успішній ідентифікації, учаснику доступні наступні кроки налаштування та проведення освітнього процесу. При неуспішній

						Аркуш
						30
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР	

авторизації, учаснику заблоковано доступ до програмного забезпечення закладу освіти. Також при успішній авторизації, відбувається процес шифрування користувача (в рамках забезпечення захищеності програмного комплексу ЗВО, та надаються права доступу до певних інформаційних ресурсів, в залежності від посади користувача.

2. Процес встановлення та налаштування програмного та апаратного забезпечення на комп'ютерах установлених в аудиторіях ЗВО. Діаграма діяльності моделює послідовність кроків, включаючи вибір програмного забезпечення, запит на наявність ліцензії, встановлення та тестування програми.
3. Діаграма діяльності процесу технічної підтримки програмно-апаратного комплексу ЗВО, включає в себе керування обладнанням, де показано, які дії виконуються при придбанні, розгортанні та обслуговуванні нового обладнання, відстеження статусу замовлень на ремонт обладнання та програмного забезпечення. Вони можуть відображати можливі стани та переходи між ними в рамках певного процесу
4. Процес обробки замовлень на ремонт обладнання. Діаграма діяльності, що показує процес обробки замовлень, включає в себе кроки, які виконуються від початку фіксації заявки до її вирішення та закриття.
5. Діаграми послідовності для взаємодій між співробітниками ЗВО та процесом технічної підтримки освітнього процесу закладу освіти.

						Аркуш
					ДТЕУ 121 023-19.МР	31
Зм.	Аркуш	№ докум	Підпис	Дата		

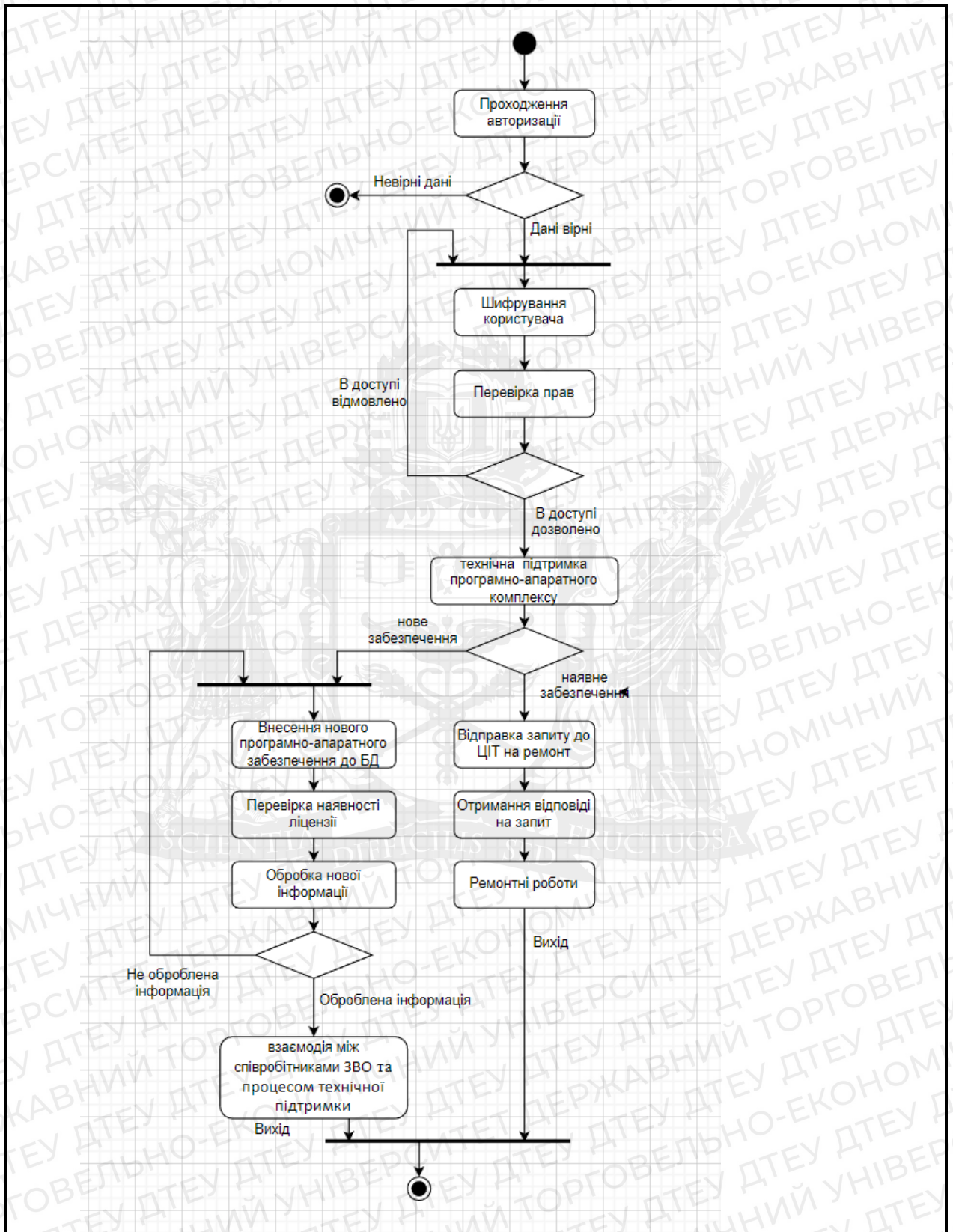


Рис.2.2. Архітектура діаграми діяльності програмного додатку

Джерело: побудовано автором

						Аркуш
					ДТЕУ 121 023-19.МР	32
Зм.	Аркуш	№ докум	Підпис	Дата		

2.4. Архітектура моделі компонентів програного додатку

Для проєктування архітектури моделі компонентів програного додатку було обрано UML-діаграма компонентів. Дана модель представляє інформаційну систему, яка включає в себе різні компоненти і модулі для забезпечення різноманітних функцій. Ця модель також передбачає наявність апаратного та програмного забезпечення, серверів для обробки та збереження даних.

Основний компонент змодельованої системи – центральний сервер, який забезпечує централізоване управління та обробку даних. Дані проходять через модулі модифікації, аналізу та фільтрації для виконання певних операцій з обробки даних.

Змодельована система має можливість шифрування та дешифрування даних за допомогою відповідних програмних компонентів. Також присутні компоненти для авторизації та ідентифікації користувачів системи.

У системі також використовуються фізичні та логічні засоби безпеки, такі як брандмауери та антивірусне програмне забезпечення, для забезпечення безпеки та захисту інформації.

В цілому, дана модель інформаційної системи освітнього процесу ЗВО демонструє структуру та функціональні можливості системи, які допомагають забезпечити безпеку та ефективну обробку даних у контексті освітнього процесу у вищому навчальному закладі (рис. 2.3.).

						ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			33

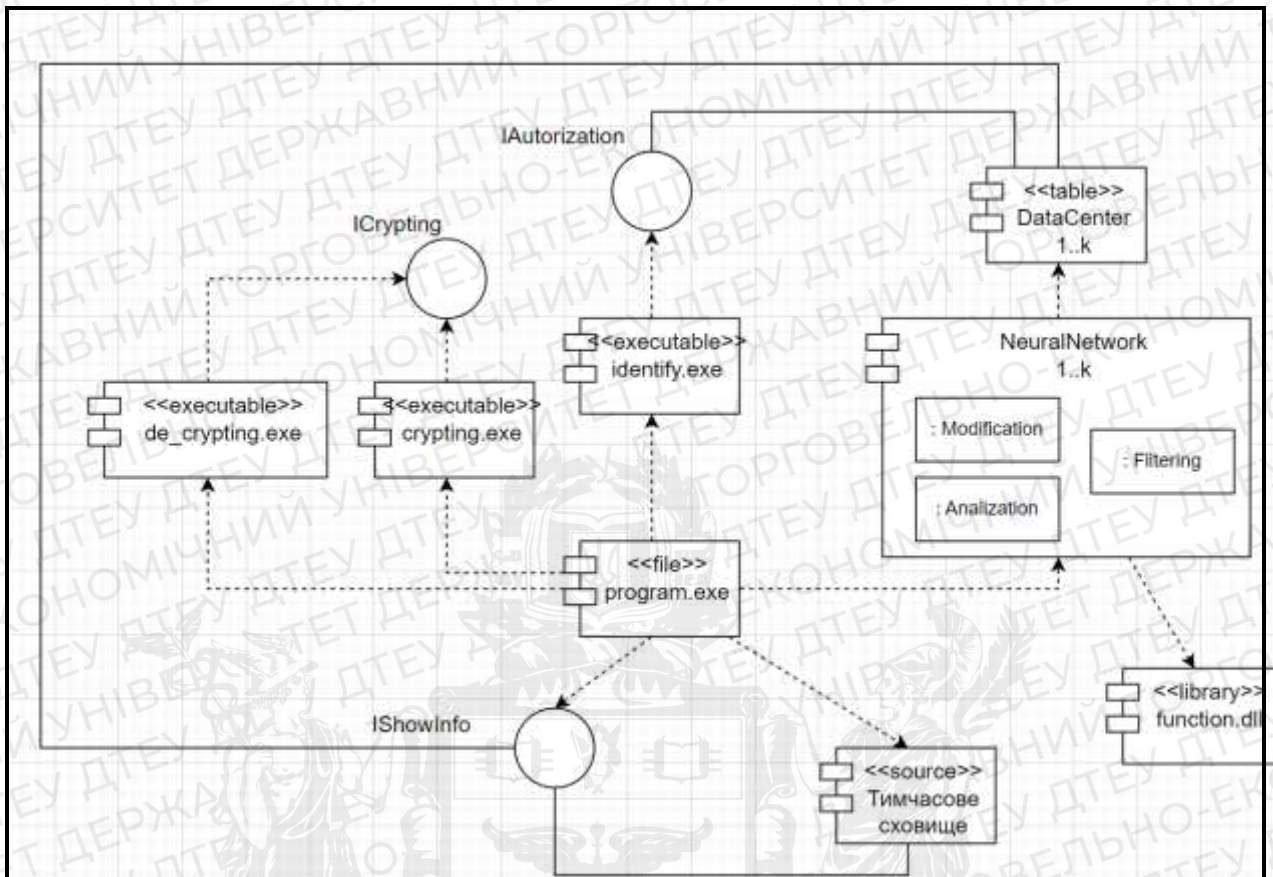


Рис. 2.3. Архітектура моделі компонентів програмового додатку

Джерело: побудовано автором

Опис компонентів змодельованої системи програмного додатку:

- **ICrypting** (інтерфейс) – компонент є інтерфейсом, який визначає методи та функції, пов'язані з криптографічними операціями. Інші компоненти реалізують цей інтерфейс для забезпечення функцій шифрування та розшифрування даних;
- **IAuthorization** (інтерфейс) – визначає методи та функції, пов'язані з авторизацією користувачів. Інші компоненти реалізують цей інтерфейс для забезпечення функцій авторизації та перевірки доступу;
- **IShowInfo** (інтерфейс) – характеризує методи та функції, пов'язані з відображенням інформації та/або результатів. Інші компоненти

реалізують цей інтерфейс для забезпечення відображення результатів або повідомлень користувачеві;

- *crypting.exe* (виконуваний файл) – відповідає за шифрування даних, використовується для шифрування даних або файлів для забезпечення їх конфіденційності та безпеки;
- *de_crypting.exe* (виконуваний файл) – відповідає за розшифрування даних, використовується для дешифрування зашифрованих даних або файлів;
- *identify.exe* (виконуваний файл) – даний компонент відповідає за ідентифікацію або перевірку ідентичності користувачів або системи, використовується для перевірки правильності введених даних автентифікації або для ідентифікації об'єктів системи;
- *program.exe* (виконуваний файл) – представляє виконавчий файл програми, яка має певні функціональні можливості;
- `<<source>>` – компонент позначає джерело початкового коду програми, який використовується для розробки або компіляції інших компонентів системи;
- `<<table>>` – компонент вказує на таблицю даних, яка використана в системі для зберігання та обробки даних, до даного компоненту відноситься база даних, спеціальна структура даних або інші засоби для зберігання даних;
- `<<library>> function.dll` – позначає бібліотеку функцій, яка містить готові функції або використані в інших компонентах системи ресурси. *function.dll* містить певний набір функцій, які використовуються для виконання певних операцій з обробки даних;
- *DataCenter 1..k* – позначає центр обробки даних або інфраструктуру, що забезпечує зберігання та обробку даних. Кількість екземплярів

						ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			35

(1..k) вказує, що може бути більше одного центру обробки даних у системі;

- *Central server 1..k* – представляє центральний сервер, який виконує централізовану обробку та управління даними та ресурсами. Кількість екземплярів (1..k) вказує, що може бути більше одного центрального сервера у системі;
- *Modification* – компонент вказує на модуль або функцію, яка відповідає за зміну або модифікацію даних та ресурсів у системі. Цей компонент забезпечує можливість змінювати дані або виконувати операції зміни стану системи;
- *Analization* – компонент представляє модуль або функцію, яка відповідає за аналіз даних та ресурсів у системі. Він виконує різні види аналізу, наприклад, аналіз взаємодії інформаційних потоків між інформаційними компонентами та інші аналітичні процеси;
- *Filtering* – компонент являє модуль або функцію, яка відповідає за фільтрацію та обробку даних і ресурсів. Він використовується для відбору або виключення певних даних, застосування певних правил або фільтрів до вхідних даних.

2.5. Проектування бази даних програмного додатку

Проектування бази даних прийнято представляти за допомогою трьох видів моделей концептуальної, логічної та фізичної. Розглянемо детально проектування кожної моделі бази даних.

Концептуальної моделі – моделі найбільш високого рівня, що зазвичай використовується для побудови моделі предметної області майбутньої бази даних без посилання на яку-небудь конкретну СУБД. Концептуальна модель

						Аркуш
					ДТЕУ 121 023-19.МР	36
Зм.	Аркуш	№ докум	Підпис	Дата		

містить тільки назви майбутніх таблиць та зв'язки між ними, та пропускає атрибути та їх типи даних.

Логічної моделі – моделі, що є розширенням концептуальної моделі за допомогою атрибутів, притаманним обраній предметній області. Концептуальна модель містить атрибути таблиць, проте все ще не вказує конкретні типи даних.

Фізичної моделі – моделі, що будується на базі логічної моделі, із врахуванням особливостей конкретної СУБД. Фізична модель вказує конкретні типи даних для всіх атрибутів із врахуванням можливостей та обмежень обраної СУБД, таким чином не є універсальною, на відміну від концептуальної та логічної моделі. У випадку міграції на іншу СУБД, фізичну модель необхідно переробити.

Концептуальна модель бази даних інформаційних компонентів системи закладу вищої освіти складається із 18 таблиць, що описують обрану предметну область.

Концептуальну модель можна зручно поділити на 4 основні частини:
Концептуальну модель можна зручно поділити на 4 основні частини:

1. Модель персоналу – *зелений* колір:

- «Довідник персоналу» – інформація про персонал.
- «Довідник посади» – інформація про наявні посади.
- «Довідник структурних підрозділів» – інформація по наявним структурним підрозділам ЗВО.
- «Проф. зростання персоналу» – інформація про здобуті сертифікати/дипломи та підвищення кваліфікації персоналу.
- «Навчання персоналу» – інформація про проведені майстер класи, лекції, інструктажі для персоналу.

2. Модель програмного забезпечення – *червоний* колір:

						Аркуш
						37
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР	

- «Виробники ПЗ» – список усіх виробників ПЗ, чії програми використовуються у ЗВО.
- «Програмне забезпечення» – інформація про ПЗ, що використовується.
- «Встановлення ПЗ» – інформація про встановлення конкретного ПЗ на комп'ютери.
- «Технічна підтримка ПЗ» – інформація про надану підтримку для встановленого ПЗ.
- «Типи технічної підтримки ПЗ» – доступні типи технічної підтримки ПЗ.

3. Модель мережі – *помаранчевий* колір, складається з таких об'єктів:

- «Довідник типу компонентів мережі».
- «Довідник компонентів мережі».
- «Компоненти структури/мережі».
- «Довідник протоколів та портів».
- «Довідник захисту мережі».
- «Налагодження моніторинг мережі».

4. Модель оргтехніки – *блакитний* колір:

- «Довідник типу ремонту» – доступні типи ремонту для оргтехніки.
- «Технічна підтримка оргтехніки» – інформація про заплановані та проведені технічні підтримки оргтехніки.

					ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		38

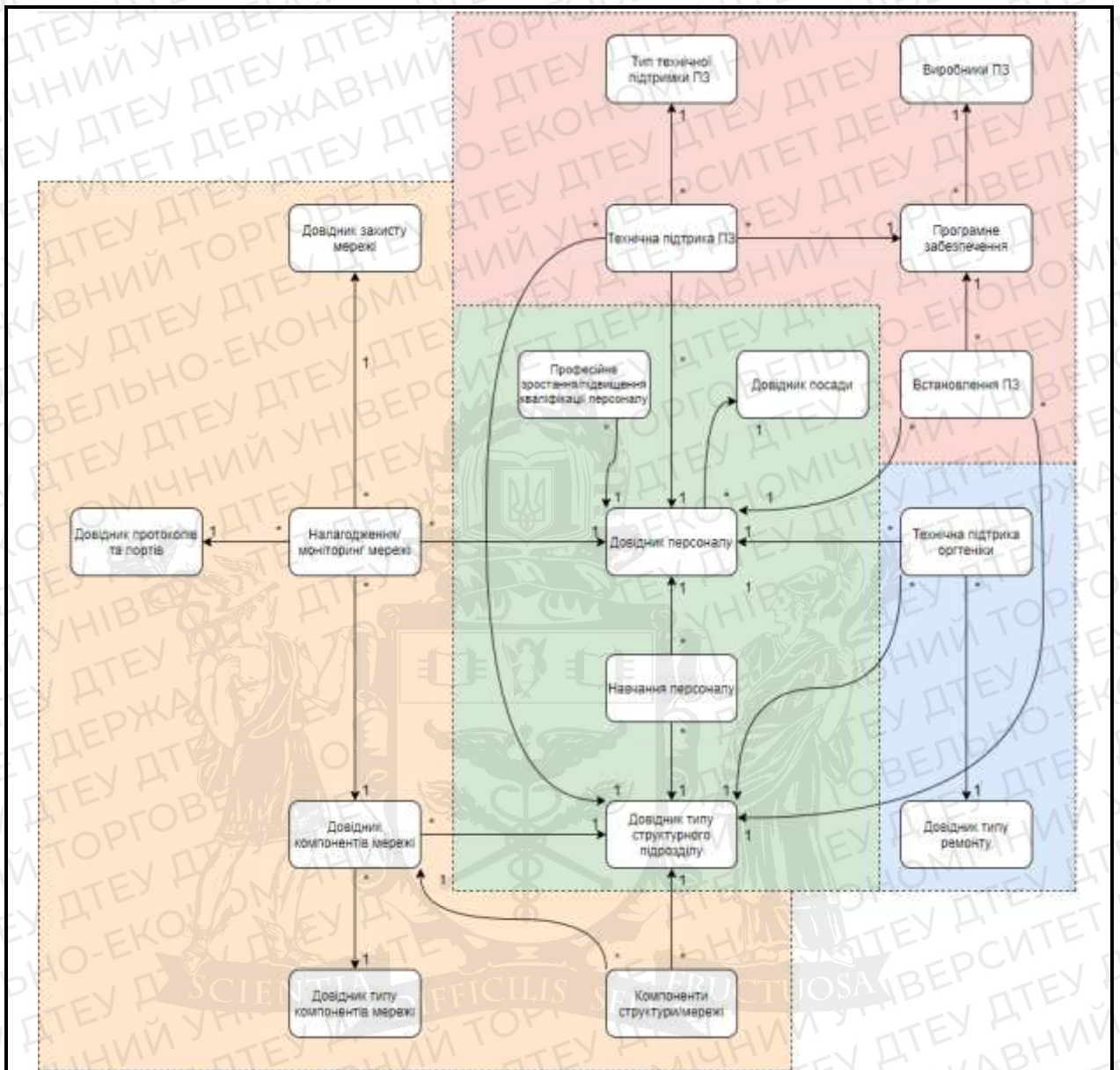


Рис. 2.4. Архітектура концептуальної моделі бази даних

Джерело: побудовано автором

Логічна модель бази даних інформаційних компонентів за своєю структурою продовжує концептуальну модель, проте додає інформацію про атрибути та ключі у кожній таблиці.

Фізична модель бази даних інформаційних компонентів за своєю структурою продовжує логічну модель та додає інформацію типи даних атрибутів відповідно до можливостей MS SQL Server. Назви таблиць та атрибутів задано із дотриманням PowerApps Naming Conventions, де назва починається з префіксу розробника ПЗ.

Таблиця 2.1. – Відношення типів даних у MS SQL та Microsoft Dataverse:

<i>Компонент MS SQL</i>	<i>Компонент Dataverse</i>
uniqueidentifier	Unique Identifier
Foreign Key	Lookup
int	Whole Number
nvarchar(n)	Text(n)
datetime2	Date and Time, Date Only
text	Text Area

2.6. Проектування архітектури апаратного забезпечення інформаційної системи

Апаратне забезпечення інформаційної системи освітнього процесу вищого навчального закладу включає різноманітні компоненти та пристрої, що використовуються для підтримки та забезпечення навчально-освітньої діяльності. Основні складові та вимоги до апаратного забезпечення інформаційної системи ЗВО наведено далі.

Центральний сервер закладу вищої освіти, який використовується для зберігання та обробки великого обсягу даних, надання послуг мережі та роботи з програмно-апаратним забезпеченням освітньої системи.

Центральний сервер розташований в спеціально обладнаному серверному приміщенні, має потужні процесори, велику кількість оперативної

							Аркуш
							42
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР		

пам'яті та великі обсяги дискового простору для зберігання даних. Базова операційна система серверу на основі Linux, Unix.

Центральний сервер виконує різні функції, залежно від потреб ЗВО. Забезпечує централізоване зберігання та управління базами даних, проводить обчислення, надає послуги електронної пошти та спільного доступу до файлів, керує мережними службами, такими як DNS або DHCP тощо. Також він використовується для побудови віртуальних оточень, що дозволяють використовувати фізичні ресурси сервера ефективніше, розділяючи їх на віртуальні машини для запуску окремих додатків чи сервісів.

Для забезпечення безпеки та надійності даних, сервер має системи моніторингу та журналювання подій для виявлення проблем та вчасного реагування на них. Центральний сервер інтегрується з рештою компонентів інформаційної системи ЗВО, надаючи засоби для забезпечення швидкого та безперебійного обміну даними, забезпечує доступ користувачів до необхідних ресурсів та виконує різноманітні завдання, пов'язані з освітнім процесом.

Робочі станції, комп'ютери, наявні в аудиторіях закладу освіти або ноутбуки, які використовуються викладачами, адміністраторами та студентами для виконання навчальних завдань, доступу до електронних ресурсів та здійснення комунікації.

Презентаційна зала, що вміщує різного роду мультимедійні пристрої, проєктори, інтерактивні дошки, звукові системи та інші пристрої, що дозволяють проводити презентації, демонструвати відео та аудіо матеріали під час лекцій та семінарських занять.

Мережне обладнання, включає в себе маршрутизатори, оптичні маршрутизатори, роутери, комутатори, мережні кабелі та інше обладнання, що забезпечує підключення комп'ютерів та інших пристроїв до локальної мережі (LAN), бездротових мереж (WiFi), віртуальних мереж (VPN), Інтернет-мережі тощо.

						Аркуш
					ДТЕУ 121 023-19.МР	43
Зм.	Аркуш	№ докум	Підпис	Дата		

Мережні маршрутизатори виконують функцію маршрутизації даних між різними мережами. Вони забезпечують комунікацію між локальною мережею закладу та зовнішніми мережами, такими як Інтернет. Маршрутизатори також мають вбудовані функції безпеки, які дозволяють фільтрувати трафік, керувати доступом та захищати мережу від несанкціонованого доступу.

Роутери виконують функцію маршрутизації даних в мережі. Вони визначають оптимальний шлях передачі пакетів даних між різними мережними сегментами. Роутери також забезпечують функції брандмауера, мережевого екрана (firewall), що дозволяють контролювати доступ до мережі та забезпечувати безпеку програмно-апаратного забезпечення ЗВО.

Комутатори використовуються для створення локальних мереж (LAN). Вони приймають пакети даних та пересилають їх на відповідні пристрої, що забезпечує швидку та ефективну передачу даних усередині мережі. Комутатори також можуть підтримувати віртуальні локальні мережі (VPN), що дозволяють логічно розділяти мережу на окремі сегменти для забезпечення безпеки та організації.

Додатково до роутерів, комутаторів та мережевих маршрутизаторів, в мережевому обладнанні використовуються такі пристрої, як мережні сервери, маршрутизатори забезпечення безпеки, проксі-сервери, балансувальники навантаження та інші. Кожен з цих пристроїв має свої функціональні можливості, які сприяють стабільності, ефективності та безпеці мережевої інфраструктури вищого навчального закладу.

Пул файлових серверів, систем зовнішнього зберігання (наприклад: NAS або хмапні ресурси OneDrive), які використовуються для централізованого зберігання та резервного копіювання даних.

Сервери зберігання даних є ключовими компонентами інформаційних систем. Вони забезпечують централізоване зберігання та керування даними. Сервери мають велику місткість для зберігання даних, швидкі та надійні

						Аркуш
						44
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР	

механізми резервного копіювання, а також можливість розширення в майбутньому. Вони також мають вбудовані функції реплікації даних, що забезпечують дублювання даних для забезпечення безпеки та доступності.

Для збереження даних також використовується системи зовнішнього зберігання, такі як пристрої на основі твердотілих накопичувачів (SSD) та мережеві пристрої зберігання (NAS). SSD-пристрої надають швидкий доступ до даних та підвищену надійність, оскільки вони не мають рухомих частин, що можуть вийти з ладу. Мережні пристрої зберігання забезпечують централізоване зберігання даних і надають доступ до них через мережу.

Окрім того, системи зберігання даних включають такі компоненти, як системи резервного копіювання та відновлення, системи архівування даних та системи контролю доступу до даних. Системи резервного копіювання та відновлення дозволяють створювати резервні копії даних для захисту від випадкового видалення, пошкодження або втрати даних. Системи архівування даних дозволяють зберігати старі або неактивні дані на довготривалих носіях зберігання, звільняючи ресурси для активних даних. Системи контролю доступу до даних забезпечують безпеку і контроль доступу до збережених даних, що дозволяє обмежувати доступ до конфіденційної інформації лише авторизованим користувачам.

Одним з ключових елементів комунікаційної інфраструктури є локальна мережа (LAN), яка забезпечує підключення комп'ютерів, серверів, принтерів та інших пристроїв в межах одного приміщення або обмеженого територіального простору. LAN забезпечує швидкий обмін даними між пристроями, спільний доступ до ресурсів, таких як принтери або сховища даних, і підтримує забезпечення безпеки мережі.

Для забезпечення зв'язку між віддаленими приміщеннями або віддаленими корпусами вищого закладу освіти використовується віртуальна приватна мережа (VPN) або міжкомп'ютерна мережа (WAN). VPN дозволяє

						Аркуш
						45
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР	

створити безпечне з'єднання через публічну мережу, таку як Інтернет, для передачі даних між віддаленими місцями. WAN використовується для підключення різних мереж та приміщень вищого навчального закладу.

Комунікаційна інфраструктура також включає бездротові мережі (WiFi), які дозволяють підключатись до мережі безпосередньо за допомогою бездротових пристроїв, таких як ноутбуки, смартфони або планшети. Бездротові мережі надають гнучкість та мобільність, дозволяючи користувачам підключатись до мережі з будь-якого місця в межах покриття.

Крім того, комунікаційна інфраструктура включає мережні пристрої, такі як маршрутизатори, комутатори і файрволи, які забезпечують маршрутизацію трафіку, комутацію даних і захист мережі. Ці пристрої дозволяють ефективно керувати трафіком в мережі, забезпечують безпеку і контроль доступу до мережевих ресурсів.

Загалом, комунікаційна інфраструктура інформаційної системи освітнього процесу забезпечує швидку та надійну передачу даних, забезпечує підключення різних компонентів системи і дозволяє забезпечити безпеку і контроль доступу до інформації. Вона є важливою складовою для ефективної роботи інформаційної системи вищого навчального закладу.

Створення засобів безпеки, що складаються з фізичних та логічних засобів захисту, такі як брандмауери, Firewall, антивірусне програмне забезпечення тощо.

Фізичні засоби безпеки включають фізичний контроль доступу до інформаційних ресурсів і приміщень, а також фізичне забезпечення серверних кімнат та інших важливих зон. До них відносяться системи відеоспостереження, пропускні системи, датчики руху, системи контролю доступу тощо. Фізичні засоби безпеки забезпечують захист від несанкціонованого фізичного доступу до обладнання та інформації, а також виявлення та реагування на відповідні можливі загрози.

						ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			46

Логічні засоби безпеки включають брандмауери і антивірусне програмне забезпечення. Брандмауери є системами, що фільтрують мережений трафік і контролюють доступ до мережених ресурсів. Вони дозволяють налаштувати правила і політики безпеки для блокування небезпечного трафіку та захисту мережі від несанкціонованого доступу. Брандмауери також проводять моніторинг подій та реагує на відповідну подію, пов'язану з безпекою.

Антивірусне програмне забезпечення є важливим компонентом логічної безпеки. Воно призначене для виявлення, блокування та видалення шкідливого програмного забезпечення, такого як віруси, троянські програми, шпигунське програмне забезпечення тощо. Антивірусні програми оновлюються регулярно, щоб розпізнавати нові загрози та забезпечувати надійний захист системи.

Загальною метою фізичних та логічних засобів безпеки є забезпечення конфіденційності, цілісності та доступності інформаційних ресурсів. Вони допомагають запобігти несанкціонованому доступу до даних, забезпечують захист від вірусів та інших шкідливих загроз, а також забезпечують стабільну та безперебійну роботу системи.

Для прикладу проектування апаратного забезпечення інформаційної системи освітнього процесу було обрано вищий освітній заклад – Державний торговельно-економічний університет (рис.2.7.).

						Аркуш
					ДТЕУ 121 023-19.МР	
Зм.	Аркуш	№ докум	Підпис	Дата		47

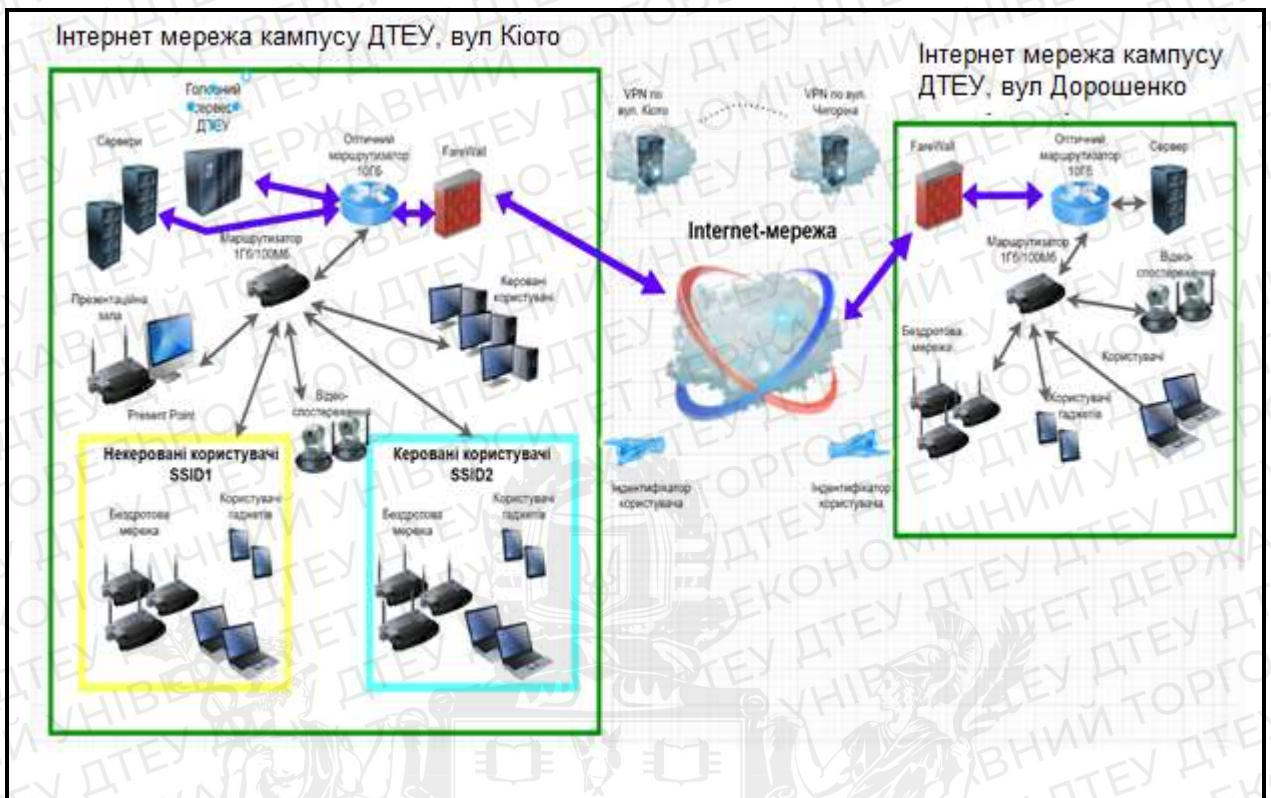


Рис. 2.7. Архітектура апаратного забезпечення інформаційної системи освітнього процесу ДТЕУ

Джерело: побудовано автором

Усі ці пристрої та компоненти мережевого обладнання взаємодіють між собою, створюючи інфраструктуру, що дозволяє забезпечувати надійну, швидку та безперебійну роботу інформаційної системи освітнього процесу. Вони забезпечують передачу даних, комунікацію між різними користувачами та системами, а також забезпечують безпеку, доступність та ефективність функціонування інформаційної системи освітнього процесу.

Компоненти систем зберігання даних сприяють забезпеченню безпеки, доступності та ефективності збереження та управління даними в інформаційній системі освітнього процесу. Вони дозволяють зберігати великі обсяги даних, забезпечують швидкий доступ до них, забезпечують безпеку та контроль доступу до інформації, а також забезпечують резервне копіювання та відновлення даних для захисту від втрати чи пошкодження.

2.7. Обґрунтування вибору програмної платформи реалізації програмного додатку

Power Platform – це група продуктів корпорації Майкрософт для розробки та побудови складних бізнес-рішень, аналізу та створення візуалізації даних та автоматизації бізнес-процесів.

Power Platform складається з: Power BI, Power Apps, Power Automate, Power Virtual Agents та набору сервісів, що їх пов'язують та обслуговують. Всі ці продукти пов'язує те, що вони потребують мінімум налаштування та працюють за принципом low-code, тобто мало коду, таким чином, роблять можливе їх використання користувачами без знання принципів програмування, так званими citizen developer`ами.

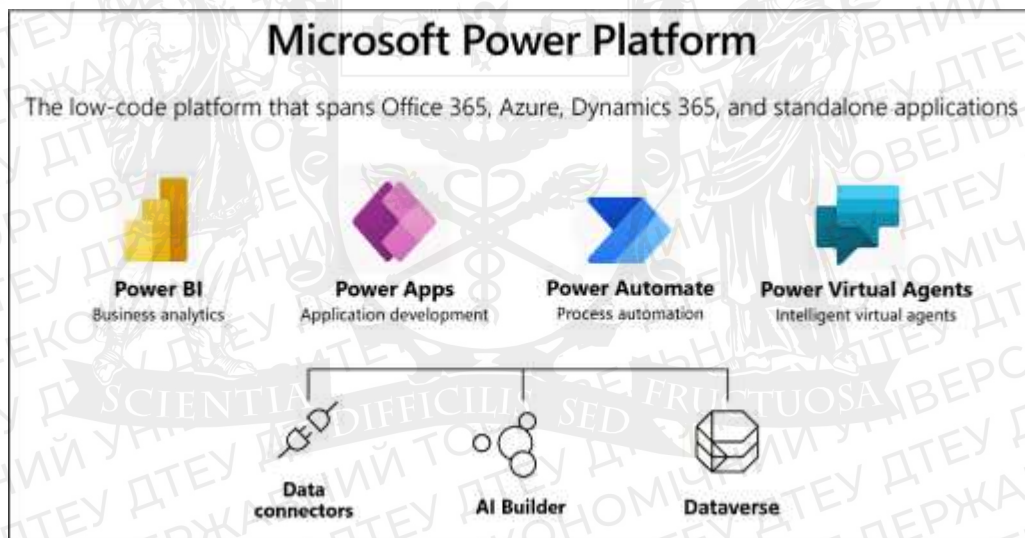


Рис. 2.8. Структура Power Platform

Іншою перевагою Power Platform є те, що її продукти безкоштовно інтегруються з іншими продуктами Microsoft, таким чином даючи можливість синхронізувати дані із будь-яких зовнішніх джерел з даними у Dataverse, базі даних Power Platform.

Power BI.

Power BI – це колекція додатків та служб, що взаємодіють між собою, щоб перетворити різноманітні джерела даних в інтерактивні аналітичні

						Аркуш
						49
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР	

представлення. Power BI дає можливість будувати графічні представлення даних, різноманітні графіки, створювати докладні звіти та інше. У якості вхідних даних можуть виступати таблиці Excel, SharePoint, різноманітні файли, Dataverse, інші бази даних та інші хмарні сховища.

Power BI складається з двох головних компонентів:

- Power BI Desktop – безкоштовна програма, що використовується для безпосереднього моделювання та побудови представлень.
- Служби Power BI – веб-додаток, що має розширений функціонал та використовується для перегляду результатів та для доступу до представлень інших користувачів закладу освіти.

Додатковими компонентами Power BI є:

- Power BI Mobile
- Power BI Report Builder

Power Apps

Power Apps – це веб-сервіс, що використовуються для побудови бізнес-додатків для власного або загального користування. Головним принципом побудови додатків у PowerApps є принцип low-code, тобто мало коду.

За допомогою PowerApps можна створювати 3 типи додатків:

- Компонований додаток (Canvas application)
- Додаток на основі моделі (Model-driven application)
- Портал (Portal application)

Canvas Application

Компонований додаток – це тип додатку, де ви маєте повний контроль над інтерфейсом користувача та будуєте його за допомогою готових компонентів. Зазвичай, цей тип додатку використовується, якщо необхідно розробити додаток для планшету або телефону.

						Аркуш
						50
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР	

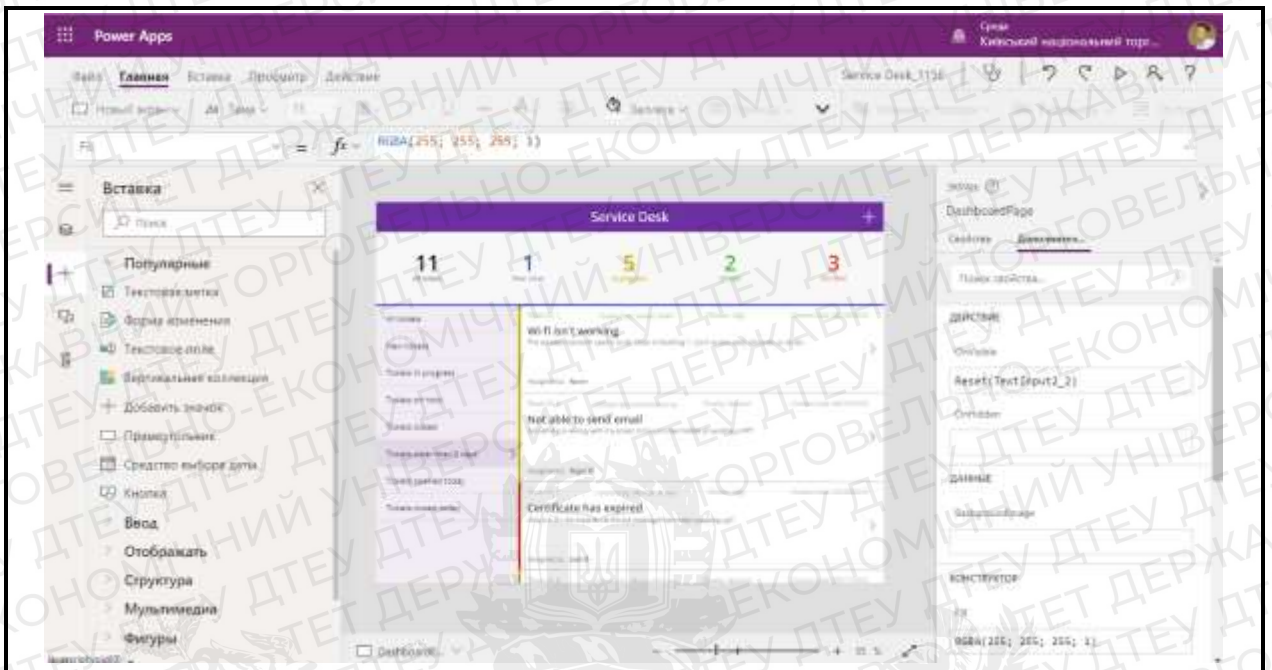


Рис. 2.9. Інтерфейс редактору Power Apps Studio

Процес створення компонованого додатку складається з будування макету за допомогою готових компонентів та програмування їх поведінки при взаємодії з користувачем. Для програмування сценаріїв використовується мова PowerFx, що за своїм синтаксисом схожа на формули у Excel.

Додаток на основі компонентів має можливість використовувати зовнішні дані за допомогою різноманітних підключень та має можливість підключитися до Microsoft Dataverse.

```

Collect(
    TicketsCollect1,
    {
        ID: Text(CountRows(TicketsCollect1) + 1),
        Owner: CreatedBy.Text,
        AccountName: Dropdown_Area.SelectedText.'Account Name',
        Priority: Dropdown_Priority.SelectedText.Priority,
        Subject: Subject.Text,
        Description: Description.Text,
        Status: "New",
        DateCreated: Text(Today())
    }
);

```

Рис. 2.10. Приклад формули PowerFx

					Аркуш
					ДТЕУ 121 023-19.МР
Зм.	Аркуш	№ докум	Підпис	Дата	51

Model-Driven Application

Головною функцією додатку на основі моделі є збір, збереження, відображення та передача даних іншим системам, саме тому, на відміну від компонованого додатку, для додатку на основі моделі є обов'язковою наявністю підключення до бази даних Microsoft Dataverse. Інтерфейс користувача шаблонний, для таблиць у Dataverse можна створювати представлення для відображення даних та форми для створення записів.

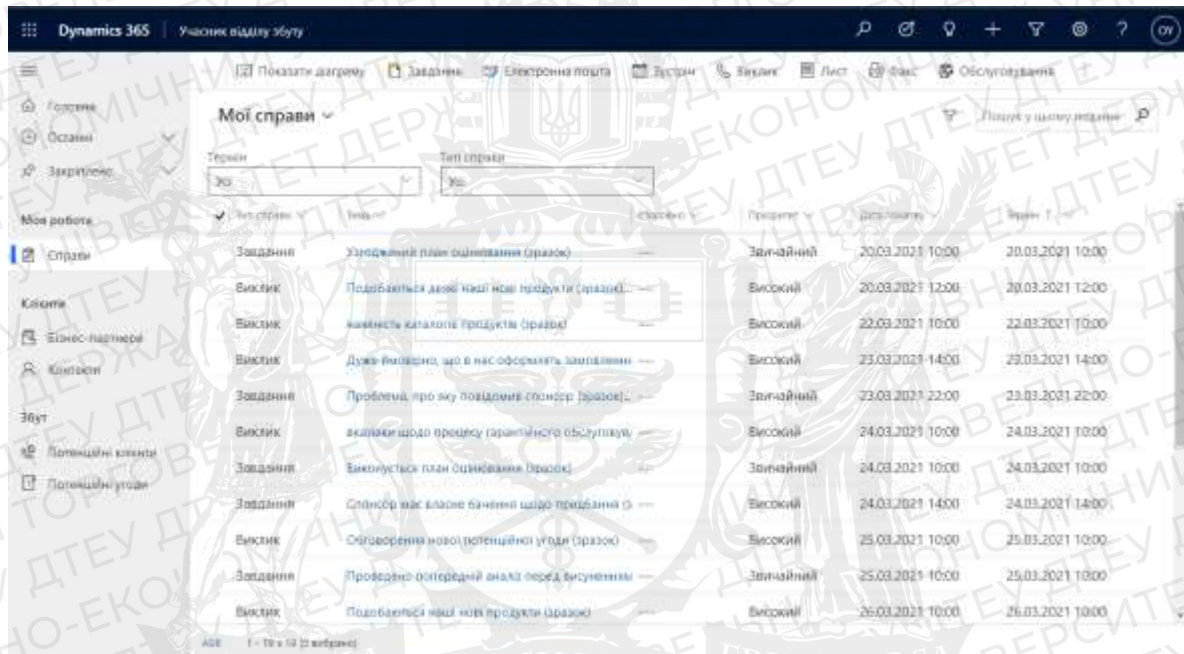


Рис. 2.11. Приклад інтерфейсу додатку на основі моделі

За допомогою додатку на основі моделі можна створювати як і прості застосунки, яким може вистачити функціональності редактору Power Apps, так і більш складні, у яких можна розширювати стандартний інтерфейс за допомогою власних компонентів, написаних на TypeScript та React або програмувати складну бізнес-логіку за допомогою C#.

Процес налаштування додатку на основі моделі складається з налаштування компонентів, які використовуються у даному додатку, тобто з налаштування карти навігації, таблиць, що зберігають дані, рядка з командами

для таблиці, подань для таблиць, форм для таблиць та обробників подій, написаних на мові C# тощо.

За допомогою редактору Power Apps для подань (View), можна налаштувати, які поля включати у вибірку, у тому числі поля із споріднених відношень, робити фільтрації та змінювати порядок записів.

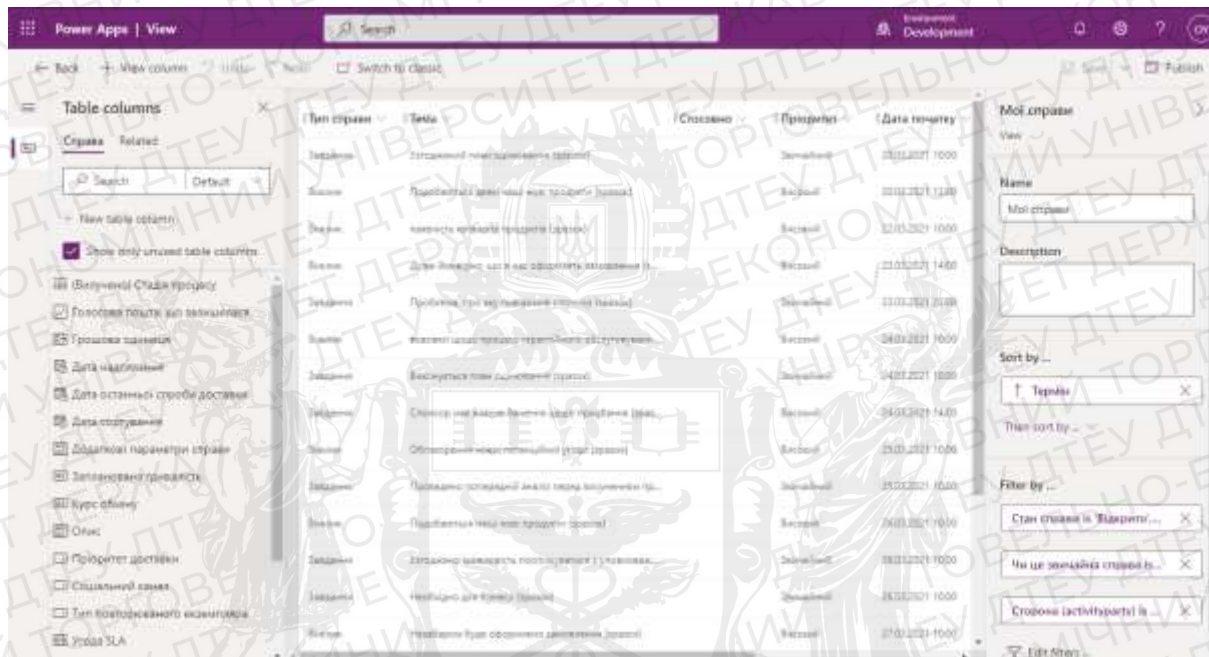


Рис. 2.12. Редактор PowerApps для подань

Редактор для форм дозволяє обирати які поля розмістити на формі для створення та відображення записів у таблиці, дозволяє робити поля обов'язковими/необов'язковими, блокувати поля, якщо не виконалась певна умова тощо. Також, для більш складних сценаріїв налаштування форми та поведінки окремих полів, можна завантажувати обробники подій, написаних на JavaScript, що дасть повний контроль над поведінкою форми та обробці або валідації даних, що зараз збираються.

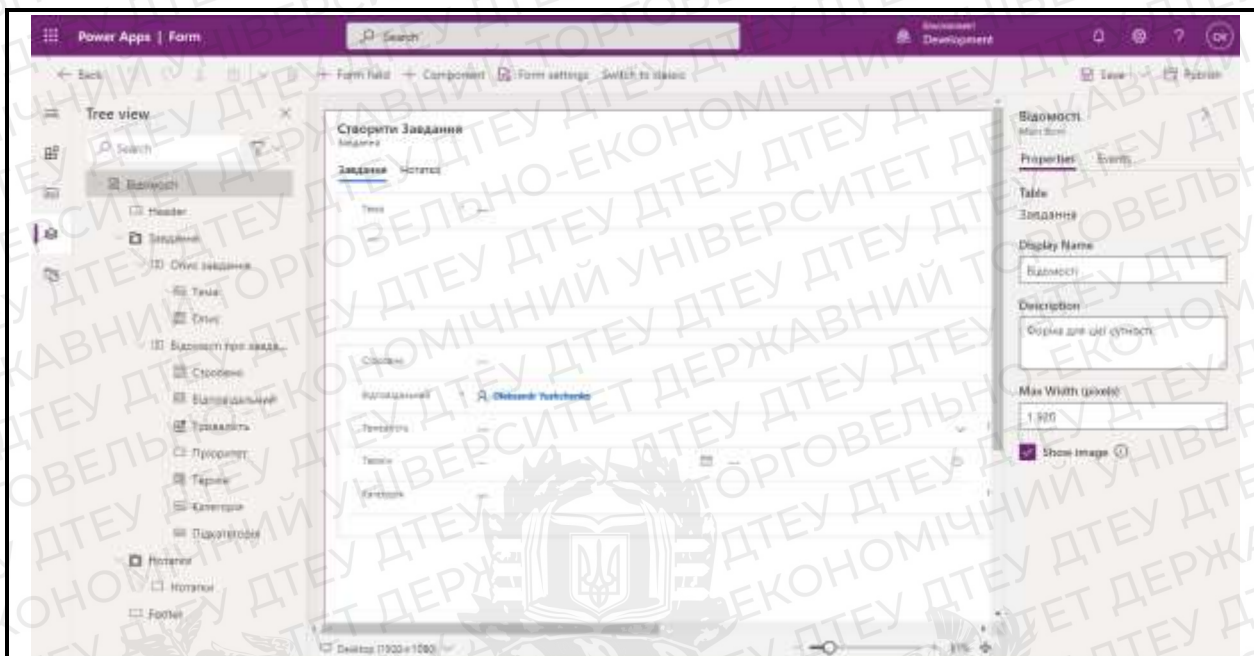


Рис. 2.13. Редактор PowerApps для форм

Portal Application

Одною із спільних ознак двох попередніх типів додатків є те, що для їх використання кінцевий користувач повинен мати акаунт в Office 365 та мати відповідні права доступу. Ключовою відмінністю портального додатку є те, що він є публічним в Інтернеті та для його використання не потрібен обліковий запис. Редактор портального додатку схожий на інші популярні редактори сайтів, наприклад Wix або WordPress.

						Аркуш
					ДТЕУ 121 023-19.МР	54
Зм.	Аркуш	№ докум	Підпис	Дата		

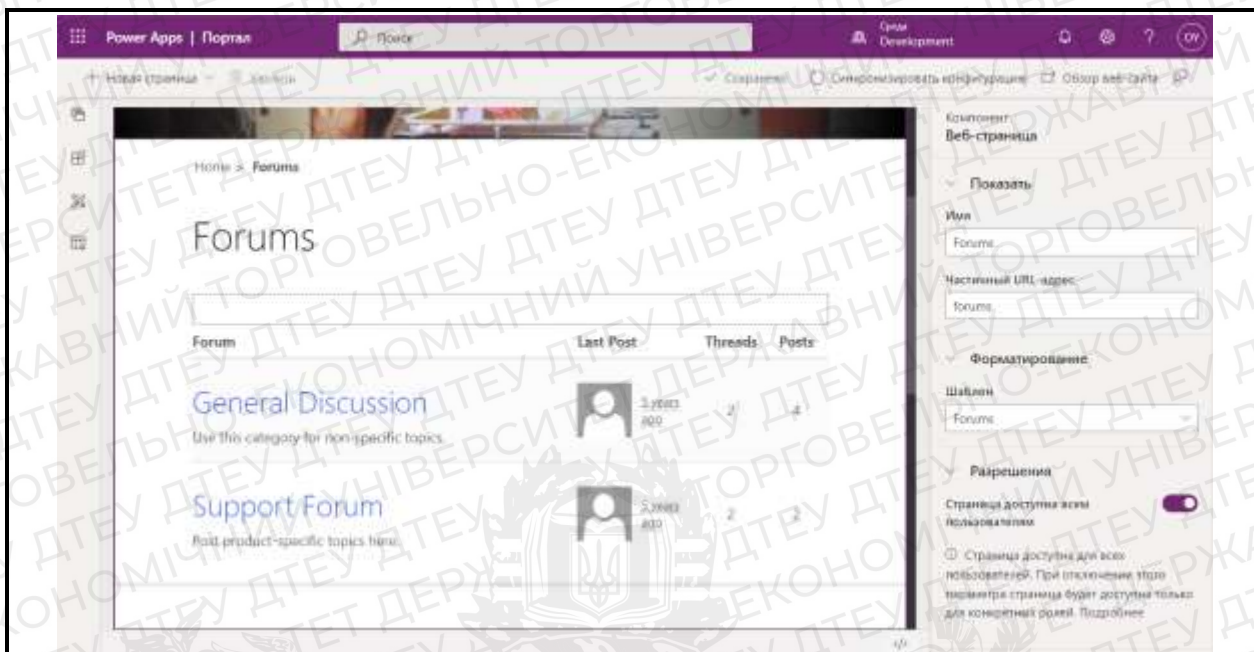


Рис. 2.14. Редактор Portal Application

Power Automate

Power Automate – це компонент Power Platform, що пропонує інструменти для автоматизації бізнес-процесів і працює за принципом «якщо-це-тоді-те». За допомогою графічного конструктора Power Automate та величезної кількості готових кроків та дій, можна будувати алгоритми та виконувати і автоматизувати такі дії, як:

- відправка нагадувань користувачам;
- створення звітів по розкладу;
- закриття прострочених активностей;
- перевід даних з одного сховища в інше;
- автоматичне збереження всіх файлів у листах, що надходять у вашу скриньку, тощо.

								Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР			55

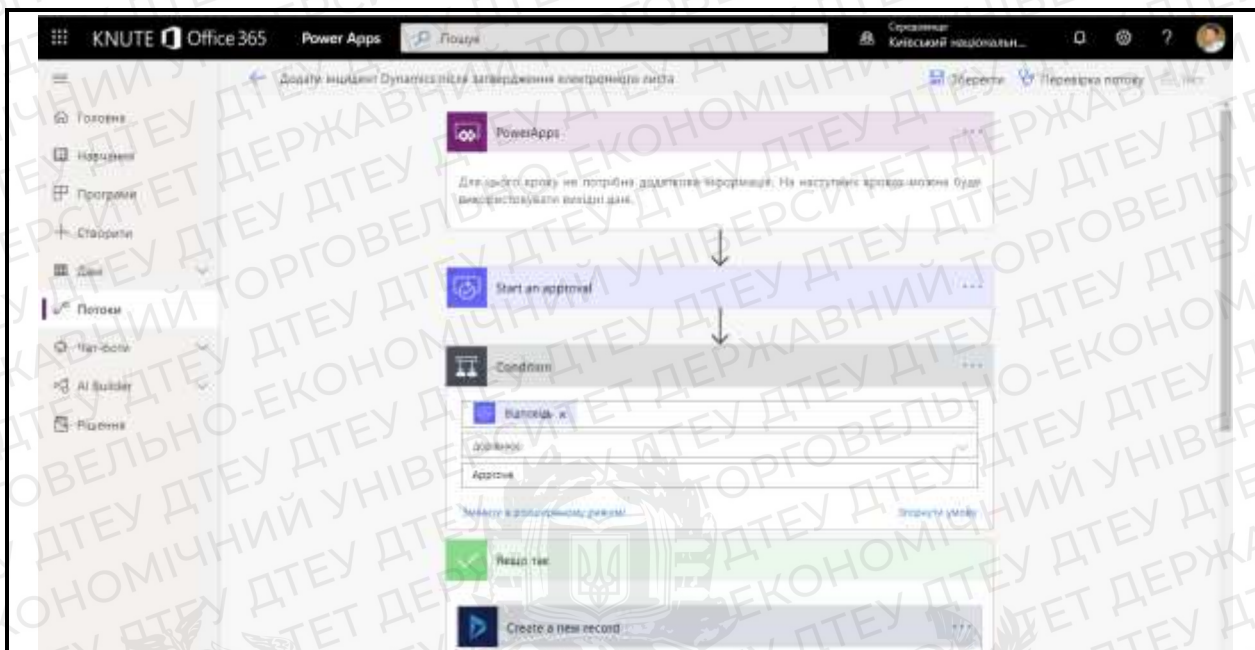


Рис. 2.14. Редактор потоків Power Automate

Power Automate має більш ніж 300 готових підключень до зовнішніх сервісів та ще більше запрограмованих дій, що можна використовувати разом.

За допомогою Power Automate можна створювати такі типи потоків:

- Автоматичні потоки – це потоки, які запускаються автоматично, якщо виконалась певна умова або дія. Налаштування таких потоків починаються з так званого тригера або ініціатора, спеціального компонента, який підключається до зовнішнього сервісу та чекає на відповідне повідомлення, та закінчується використанням готових запрограмованих дій.

- Потоки за розкладом, що запускаються по графіку.
- Потоки, що запускаються власноруч, наприклад з мобільного пристрою із спеціального додатку, або із будь-якого додатку, створеного за допомогою Power Apps.

Power Virtual Agents

Virtual Agents – це відносно нова технологія, що дозволяє будувати «розумних» чат-ботів за допомогою простого візуального конструктору.

						Аркуш
						56
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР	

- Перед валідацією операції
- Перед головною операцією
- Після головної операції

Таким чином, можна запрограмувати будь-яку бізнес логіку, що охоплює базові операції над даними. У випадку, якщо стандартних повідомлень не вистачає, можна реєструвати власні, більш специфічні, повідомлення, додавати на ці повідомлення нові обробники подій та потім ініціювати ці повідомлення власноруч або з інших повідомлень.

2.8. Висновки до розділу 2

У другому розділі «Проектування програмно-апаратної архітектури інформаційної системи», автором були розглянуті основні аспекти такого проектування, а саме визначено особливості проектування, зокрема врахування вимог до продукту, його функціональних та нефункціональних характеристик, а також вимог до апаратного забезпечення.

У розділі також була розглянута архітектура класів програмного додатку була розглянута з метою організації структури програми, розподілу функцій між класами та забезпечення модульності та розширюваності системи. Моделювання архітектури діаграмами діяльності програмного додатку надає зрозумілу візуалізацію послідовності дій, що відбуваються в програмі.

Розроблена архітектура моделі компонентів програмного додатку описує структуру програми з точки зору компонентів, їх взаємодії та відповідності. Ця архітектура допомагає забезпечити розподіл функціональності між компонентами і їх залежність та взаємодію.

Проектування бази даних програмного додатку виявляється критичним для ефективної роботи системи. Автор проаналізував різні аспекти структури бази даних та взаємозв'язків між її елементами.

						Аркуш
						58
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР	

Також автор детально розглянув та запроєктував архітектуру апаратного забезпечення інформаційної системи. Вибір правильного апаратного забезпечення відіграє важливу роль у забезпеченні ефективності та надійності системи.

Автор обґрунтував вибір програмної платформи для реалізації програмного додатку, звернувши увагу на її відповідність вимогам проекту та забезпечення необхідного функціоналу.



								Аркуш
								59
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР			

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ДОДАТКУ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Попередні приготування

Розробка будь-яких додатків у PowerApps починається з обрання середовища (Environment), де цей додаток буде знаходитися. Середовище – це одне із ключових понять Power Platform. Середовище представляє собою контейнер, що зберігає інформацію про додатки, користувачей, їх права доступу та підключення до бази даних Microsoft Dataverse. У одному середовищі може бути максимально одна база даних Dataverse, при чому, базу даних, що вже використовується не можна відключити та використовувати в іншому середовищі.

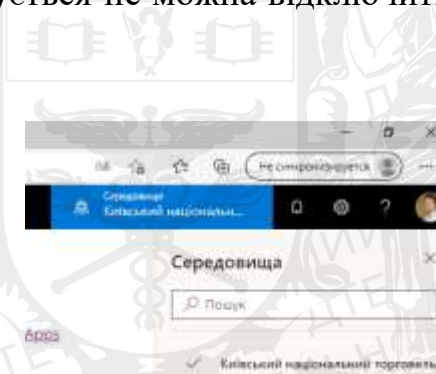


Рис. 3.1. Перемикач середовищ у редакторі PowerApps

Наступне ключове поняття необхідно для розробки додатків – це рішення (Solution). Рішення – це контейнер, що використовується для зберігання налаштувань та для їх переносу між середовищами.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 02з-19.МР</i>			
Зав. каф.		Криворучко О.В.			Архітектура інформаційної системи освітнього процесу ЗВО	Стадія	Аркуш	Аркушів
Керівник		Харченко О.А.				РЗ	60	82
Гарант		Котенко Н.О.				Факультет інформаційних технологій 2м курс, 2з група		
Розробив		Шестак Я.І.						
					Реалізація програмного додатку інформаційної системи			

Рішення бувають двох типів: керовані та некеровані. Керовані рішення зазвичай використовуються для безпосередньої розробки додатків та компонентів, а некеровані – у випадку, якщо готову роботу потрібно встановити у нове середовище.

Головна відмінність двох типів рішень у тому, що якщо із середовища видалити некероване рішення, всі налаштування та увесь зміст некерованого рішення не видалиться, а залишиться у стандартному рішенні середовища (стандартне рішення середовища присутнє завжди та його неможливо видалити). Якщо із середовища спробувати видалити кероване рішення, увесь його зміст видалиться разом із рішенням. Таким чином, за допомогою керованих рішень можна розповсюджувати, інсталювати та видаляти цілі додатки із середовищ. Так, наприклад, за допомогою керованих рішень Microsoft розповсюджує свої додатки Dynamics 365.

Створене рішення по замовчуванню є некерованим, його тип можна змінити при експорті рішення.

Нове рішення X

SCIENTIA DIFFICILIS SED FRUCTUOSA

Коротке ім'я *

Назва *

Видавець *

KNUTE

Змінити видавця

Версія *

1.0.0.0

Більше параметрів ▾

Рис 3.2. Створення нового рішення

					ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		61

Рішення прийнято називати відповідного до його призначення.

Видавець рішення – об’єкт, що зберігає інформацію про розробника рішення та деяку іншу базову інформацію, таку як префікс до всіх компонентів, що створюється у даному рішенні. За допомогою префіксу розробника можливо легко відрізнити системні компоненти (що не мають префіксу), від власних або зовнішніх компонентів. Так, компоненти продуктів Microsoft Dynamics 365 зазвичай мають префікс *msdyn*.

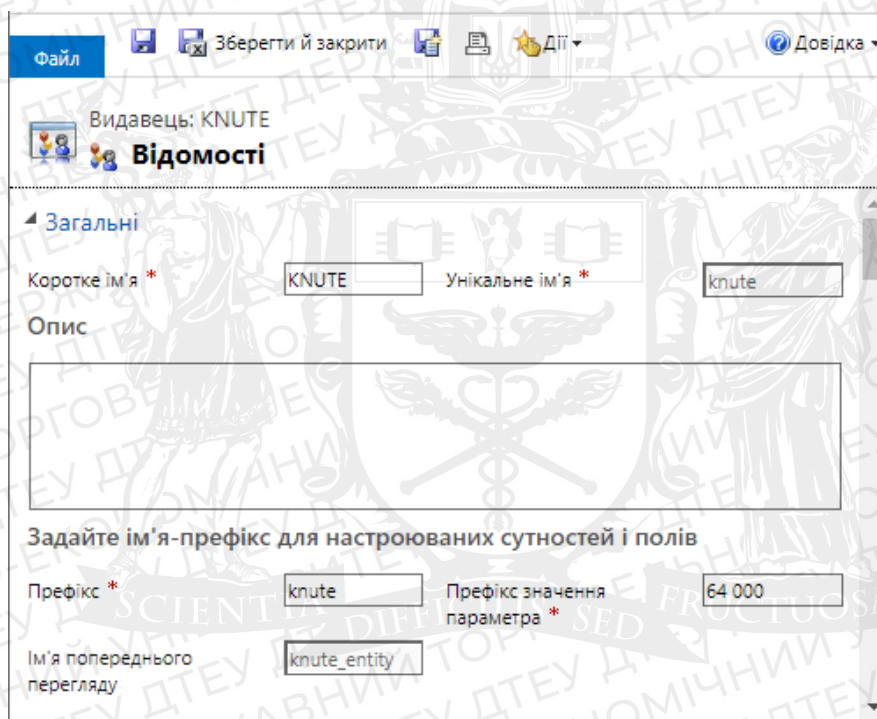


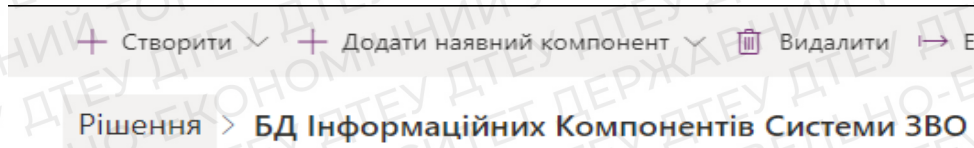
Рис. 3.3. Інформація про розробника

Створимо нове рішення, для групування усіх компонентів, що буде створено протягом роботи. У якості короткого імені рішення можна обрати тему роботи – «БД Інформаційних компонентів системи ЗВО», у якості розробника – заздалегідь створеного розробника KNUTE, а у якості назви – переклад імені на англійську – *DB_Inform_Components*. Нове рішення готове для використання.

						Аркуш
						62
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР	

3.2. Створення таблиць

Для створення таблиць потрібно перейти у новостворене рішення та натиснути кнопку «+ Створити».



Далі, у випадаючому меню обрати пункт таблиця.

Створити таблицю ✕

Коротке ім'я *

Коротке ім'я в множині *

Назва * ⓘ

knote_

Стовпець основного імені ⓘ

Коротке ім'я *

Name

Назва * ⓘ

knote_ Name

Увімкнути вкладення (включно з примітками та файлами)

Рис. 3.4. Створення нової таблиці

Обов'язкові поля при створенні нової таблиці:

- Коротке ім'я – назва таблиці будь-якою мовою, за цією назвою створені таблиці можна буде знайти у додатку при навігації. Можна змінити після створення таблиці.

					ДТЕУ 121 02з-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		63

- Коротке ім'я у множині – назва таблиці у множині. Можна змінити після створення таблиці.
- Назва – назва таблиці англійською, безпосередня назва таблиці у базі даних. Назва таблиці завжди починається з префіксу, що обирається при створенні рішення шляхом вибору розробника. Незмінна після створення таблиці.
- Увімкнуті вкладення – цей параметр визначає, чи можна у таблиці створювати стовпці з типом даних «Файл». Можна змінити після створення таблиці.

Після створення таблиці, автоматично додаються системні стовпці, що притаманні всім таблицям у Dataverse, наприклад:

- Дата створення запису.
- Ким створено запис.
- Дата останньої зміни запису.
- Ким змінено запис.
- Статус та стан запису, тощо.

Коротке ім'я ↑	Назва
Автор	createdby
Автор (представник)	createdonbehalfby
Дата змінення	modifiedon
Дата створення	createdon
Дата створення запису	overriddencreatedon

Рис. 3.5. Приклад системних стовпців (без префіксу)

Для того щоб додати власний стовпець, у редакторі таблиці потрібно натиснути кнопку «+ Додати стовпець».

Властивості стовпця ✕

Коротке ім'я *

Назва * ⓘ

Тип даних * ⓘ

Обов'язково * ⓘ

Можна шукати ⓘ

Обчислювані або зведення ⓘ

Опис ⓘ

Рис. 3.6. Створення нового стовпця

- Коротке ім'я – назва стовпця будь-якою мовою, за цією назвою поле можна буде знайти на формі або у представленні. Можна змінити після створення стовпця.
- Назва - назва стовпця англійською, безпосередня назва стовпця у базі даних. Назва стовпця завжди починається з префіксу, що обирається при створенні рішення шляхом вибору розробника. Незмінна після створення стовпця.
- Тип даних – тип даних стовпця. В залежності від типу даних стовпця змінюється можливий набір даних, що у даний стовпець можна зберегти та змінюється елемент керування на формі.

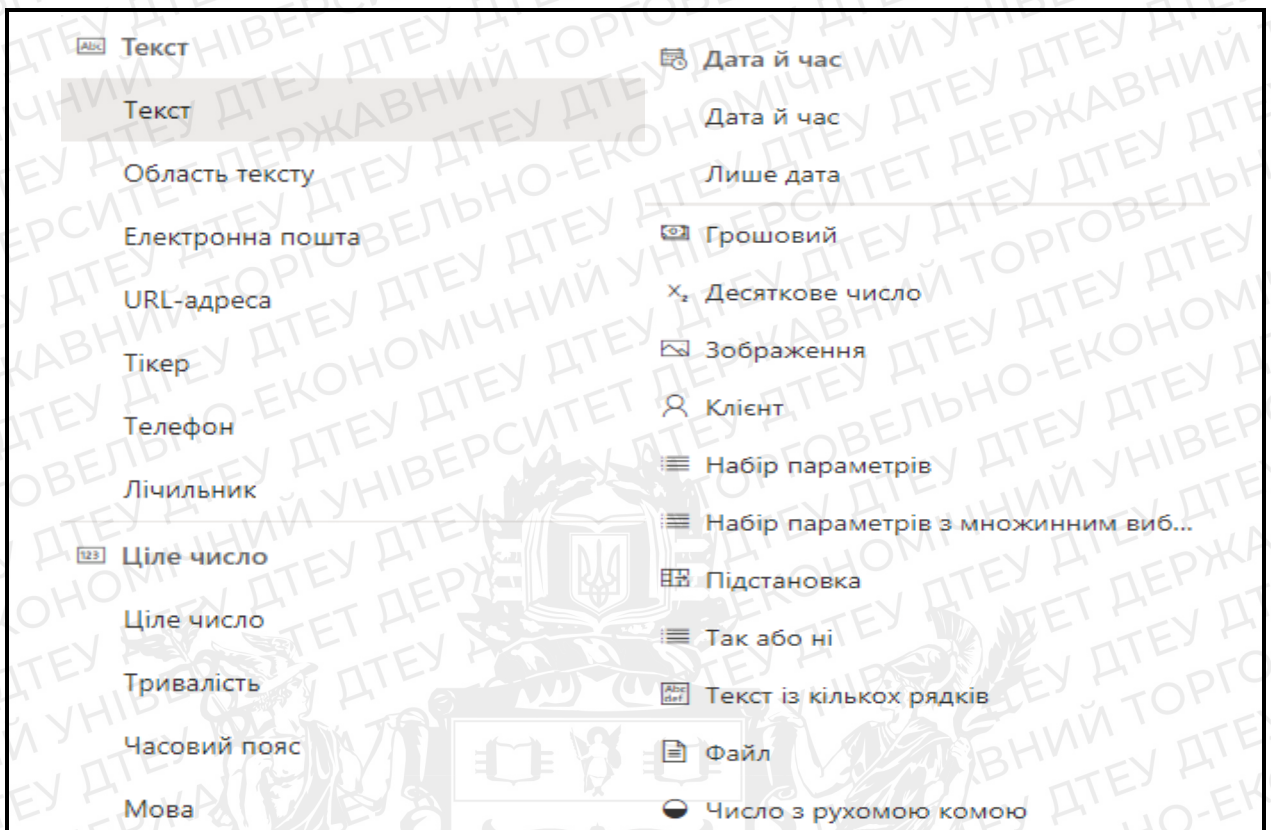


Рис. 3.7. Можливі типи даних у Dataverse

- **Обов'язковість** – вказує, чи є поле обов'язковим при створенні нового запису за допомогою форми. Можливі значення:
 - **Обов'язково.**
 - **Рекомендовано.**
 - **Необов'язково.**

3.3. Налаштування форм

Форми використовуються для створення нових записів та для відображення детальної інформації по створеному запису.

Типи форм:

- **Головна форма** – форма, що використовується для збору та відображення інформації. Головна форма підтримує найбільшу кількість різноманітних налаштувань, у тому числі додавання

					ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		66

скриптів мови JavaScript, завантаження різноманітних вебресурсів, вбудовані html сторінки (IFrame), та елементи керування, створені за допомогою PCF.

- Форма швидкого створення – форма, що використовується тільки для створення нових записів. На відміну від головної форми, створення запису за допомогою «швидкої» форми не відкриває нову сторінку, залишає користувача на ініціюючій сторінці або представленні.
- Форма швидкого перегляду – форма, що використовується для перегляду інформації із споріднених записів. Для використання форму швидкого запису її необхідно помістити на головну форму.
- Форма картки – форма, що використовується тільки для відображення записів, зазвичай на приладних дошках. Форма картки підтримує не більше п’яти простих полів.

Після створення необхідних стовпців можна створити форму, що включає створені стовпці. Таблиця може включати безліч головних форм, при цьому, якщо у таблиці більше ніж одна активна головна форма, перемикач форм стане активним у додатку. Новостворена таблиця завжди має 3 форми: головну, форму картки та форму швидкого перегляду.

Для створення та редагування форм у редакторі таблиці потрібно перейти у вкладку «Форми». Для створення нової форми необхідно натиснути на кнопку «Додати форму» та обрати бажаний тип, а для редагування існуючих форм потрібно обрати бажану форму.

						ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			67

кількість записів та роботи над ними певні дії, наприклад: групове редагування, групове видалення записів тощо.

Для створення або редагування існуючих подань у редакторі таблиць необхідно перейти у вкладку подання.

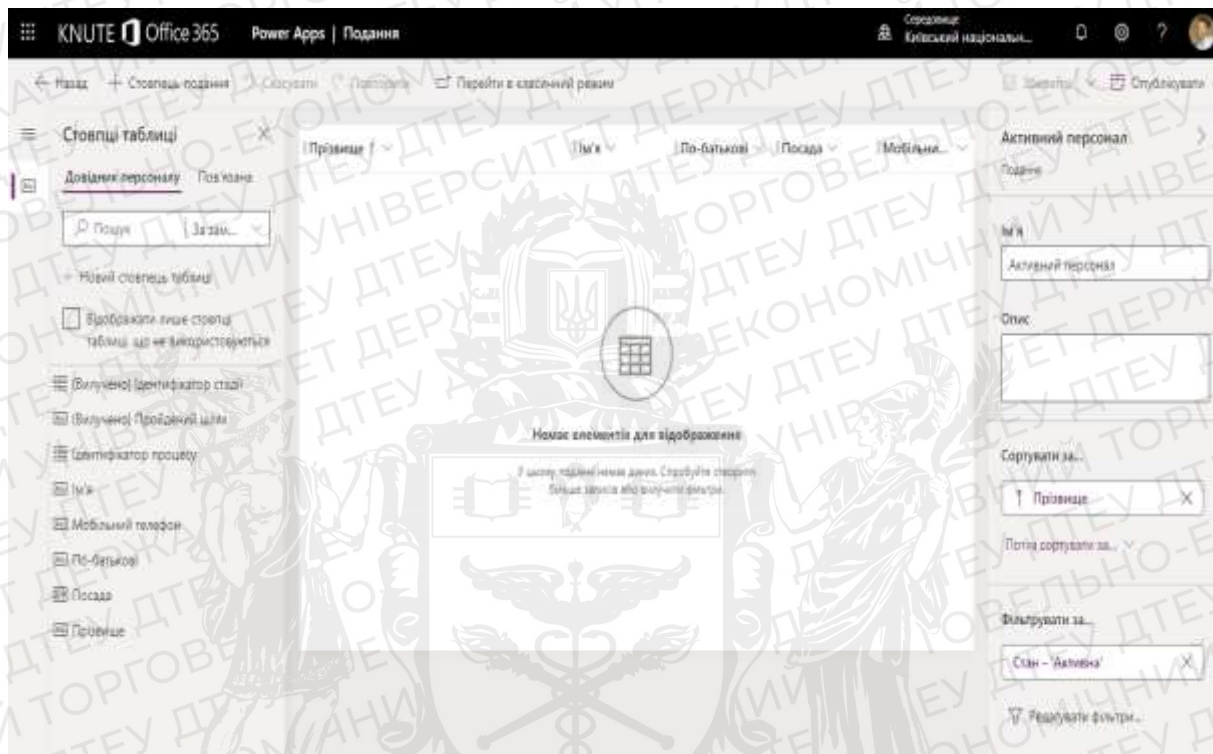


Рис. 3.9. Редактор подань

За допомогою редактору подань можна додавати існуючі стовпці з обраної таблиці та із споріднених. Редактор подань також дозволяє робити фільтрації та сортування за стовпцями головної та споріднених таблиць.

Після закінчення всіх бажаних налаштувань, подання треба обов'язково зберегти. Для того щоб проведені налаштування відображалися у додатку, подання необхідно опублікувати.

					ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		69

3.5. Створення додатку на основі моделі

Після створення та налаштування всіх необхідних таблиць, можна приступити до створення додатку. Для створення нового додатку необхідно перейти у рішення та натиснути кнопку «+ Створити», обрати пункт програма, програма на основі моделі.

Рис. 3.10. Створення нового додатку

Ім'я програми задає те, як ця програма буде називатися, за цим іменем її можна буде знайти у списку всіх програм. Унікальне ім'я – це ім'я програми у системі.

Після заповнення всіх необхідних полів, необхідно перейти до конструктору додатку. За допомогою конструктору можна обирати які таблиці включати у додаток та обирати форми і подання для даних таблиць.

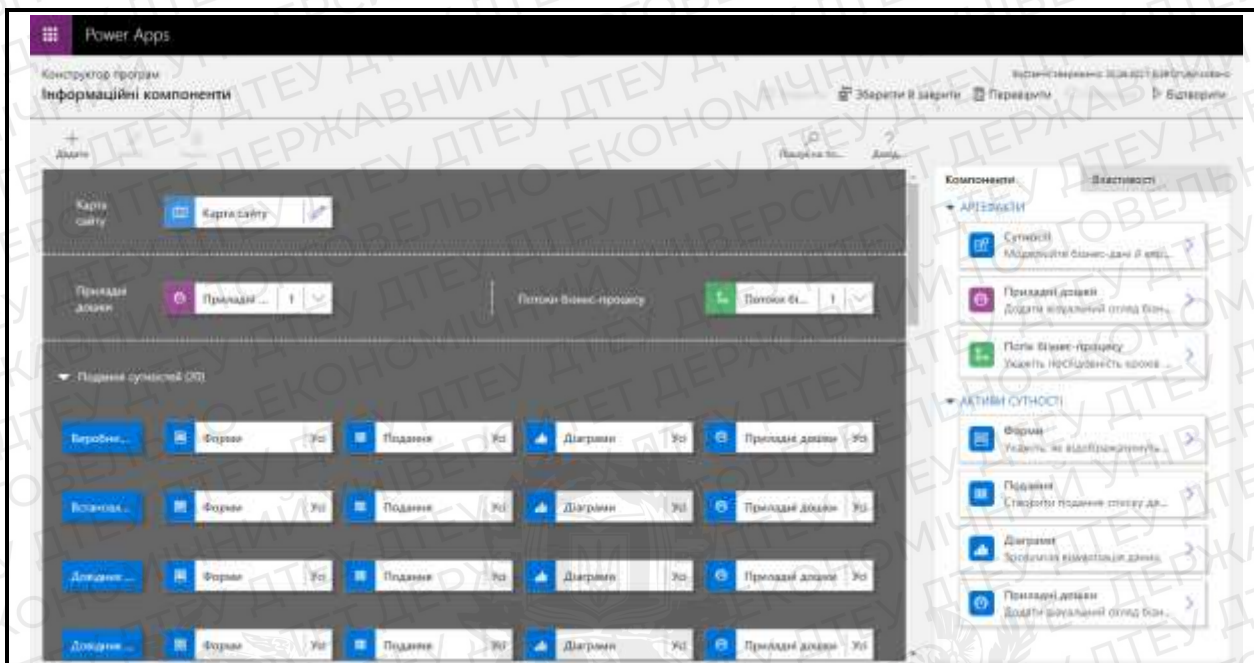


Рис. 3.11. Конструктор додатку

Після додавання бажаних таблиць, форма та подань, необхідно створити мапу додатку, що буде використовуватися для навігації по додатку.

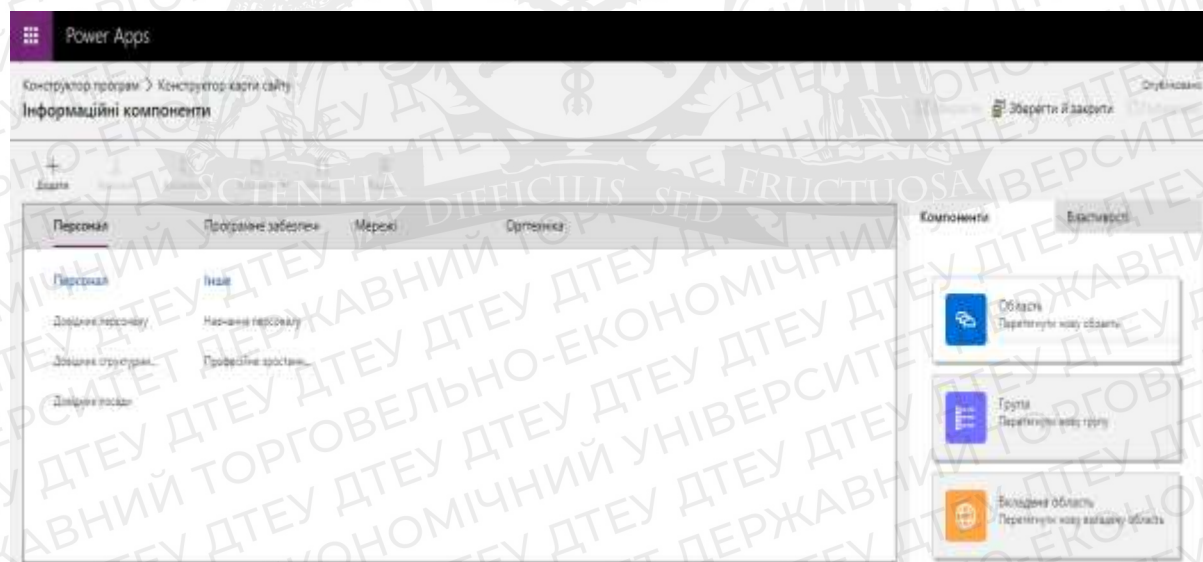


Рис. 3.12. Конструктор мапи додатку

У конструкторі мапи можна створювати області, тобто вкладки, які можна буде перемикати, в області створювати групи з таблицями.

Після налаштування мапи її необхідно зберегти та опублікувати, потім зберегти та опублікувати додаток. Додаток готовий для тестування.

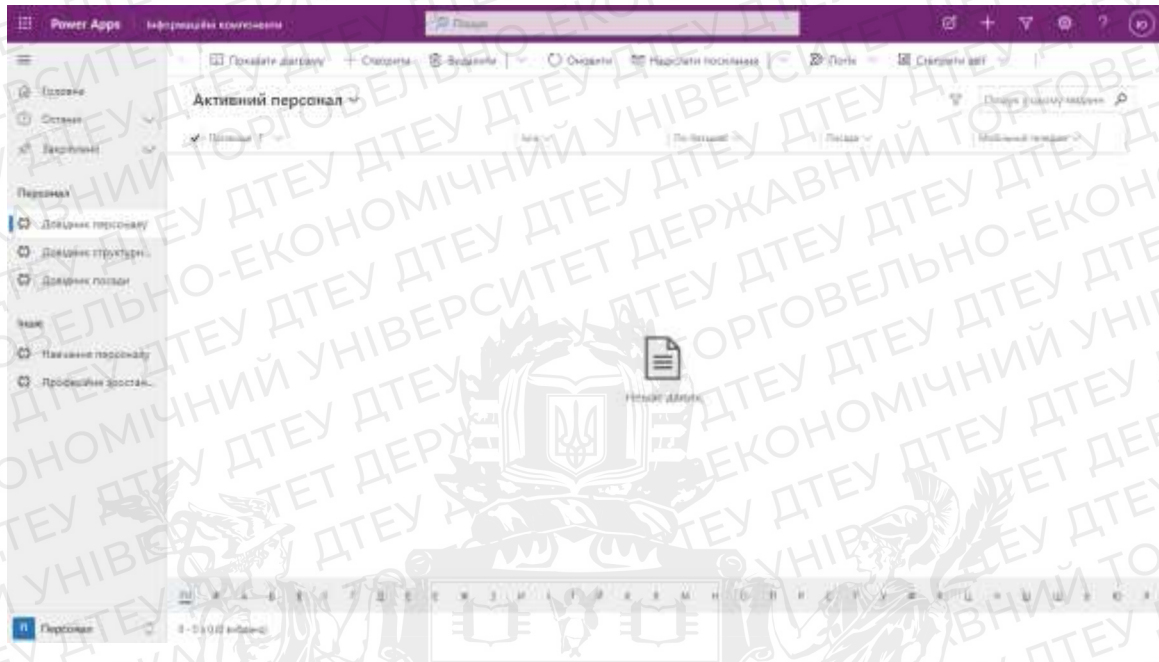


Рис. 3.13. Додаток на основі моделі

Зліва знаходиться мапа навігації, при натисканні на таблицю відкривається подання для цієї таблиці. Якщо подань декілька, їх можна перемикаати. За допомогою командної стрічки, що знаходиться над поданням, можна створити новий запис або редагувати існуючі та проводити інші операції над даними.

3.6. Налаштування ролей доступу

Наразі створений додаток доступний тільки адміністраторам середовища PowerApps, оскільки для цього додатку не налаштовані ролі доступу. Базовою практикою для створення нових ролей доступу є копіювання стандартних системних ролей та додавання у скопійовану роль потрібних прав.

						Аркуш
						72
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР	

Для створення нової ролі на базі існуючої необхідно перейти у розширені налаштування, обрати пункт безпека, ролі безпеки. Далі у поданні для ролей безпеки необхідно обрати роль, що стане базою для нової ролі та натиснути копіювати.

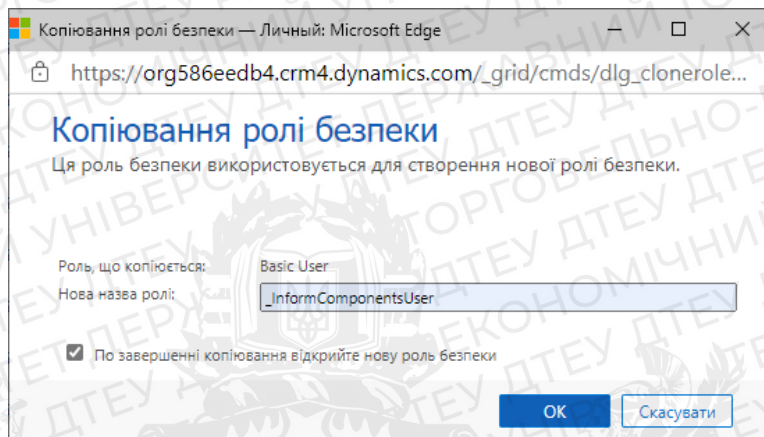


Рис. 3.14. Створення нової ролі на базі існуючої

Після збереження нової ролі можна переглянути її права доступу.

Всі налаштування можна залишити як є (за замовчанням), тільки у вкладці «Робота над рішенням» (рис. 3.15) додати права доступу до тих таблиць, що використовуються у додатку. Права доступу у таблиці можна налаштовувати для таких операцій як:

- Читання
- Редагування
- Видалення
- Створення
- Передача запису тощо.

					ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		73

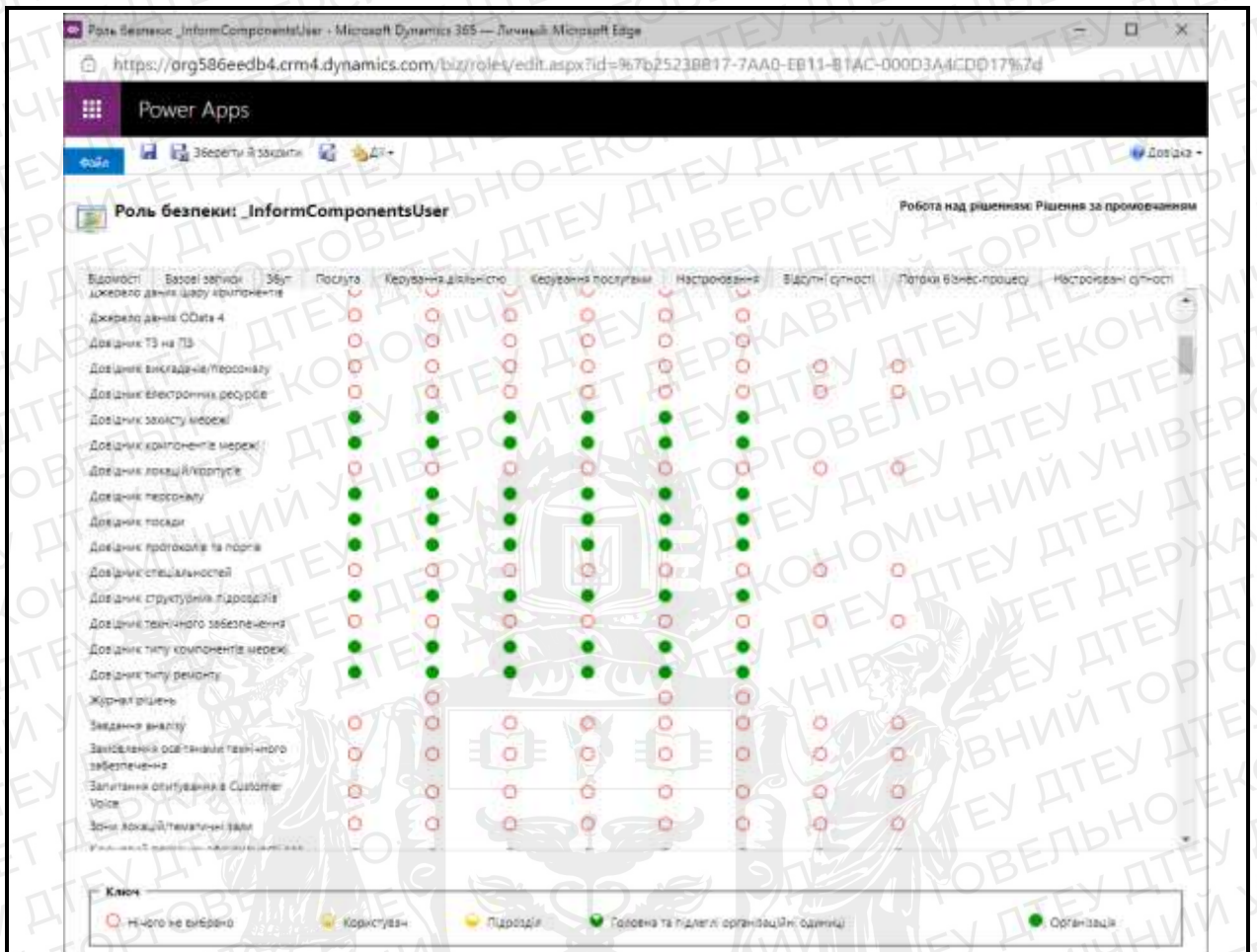


Рис. 3.15. Налаштування програмного додатку

Після збереження нової ролі її необхідно додати у додаток. Для цього треба перейти безпосередньо у додаток, натиснути на його назву зліва згори, у списку додатків знайти бажаний та відкрити для нього параметри.

Доступ до бази даних (рис. 3.17).

Всі таблиці, що використовуються у додатку, створюються та редагуються за допомогою редактору PowerApps, який у свою чергу робить зміни у Dataverse. Проте, до бази даних MS SQL все одно є доступ на читання.

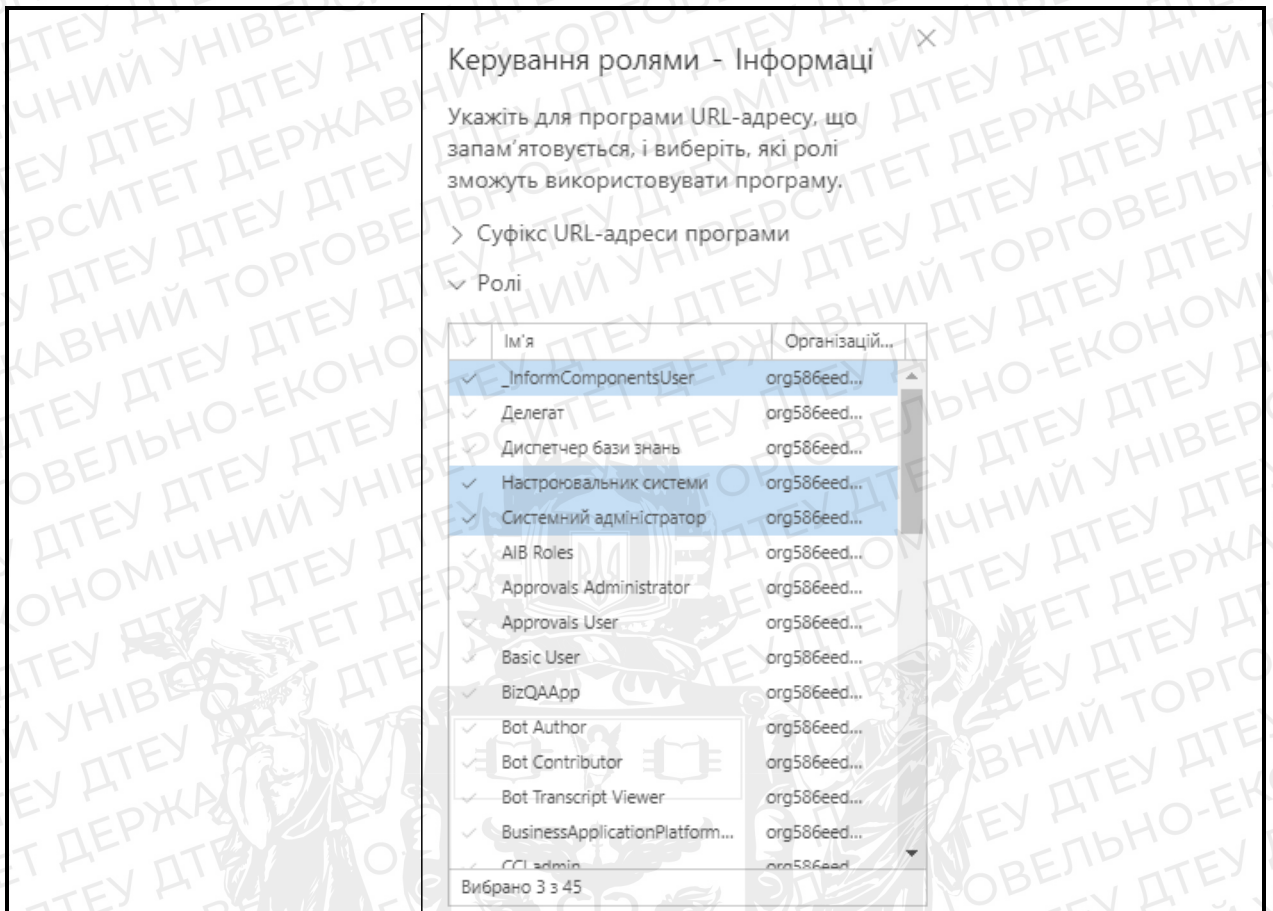


Рис. 3.16. Керування ролями додатку

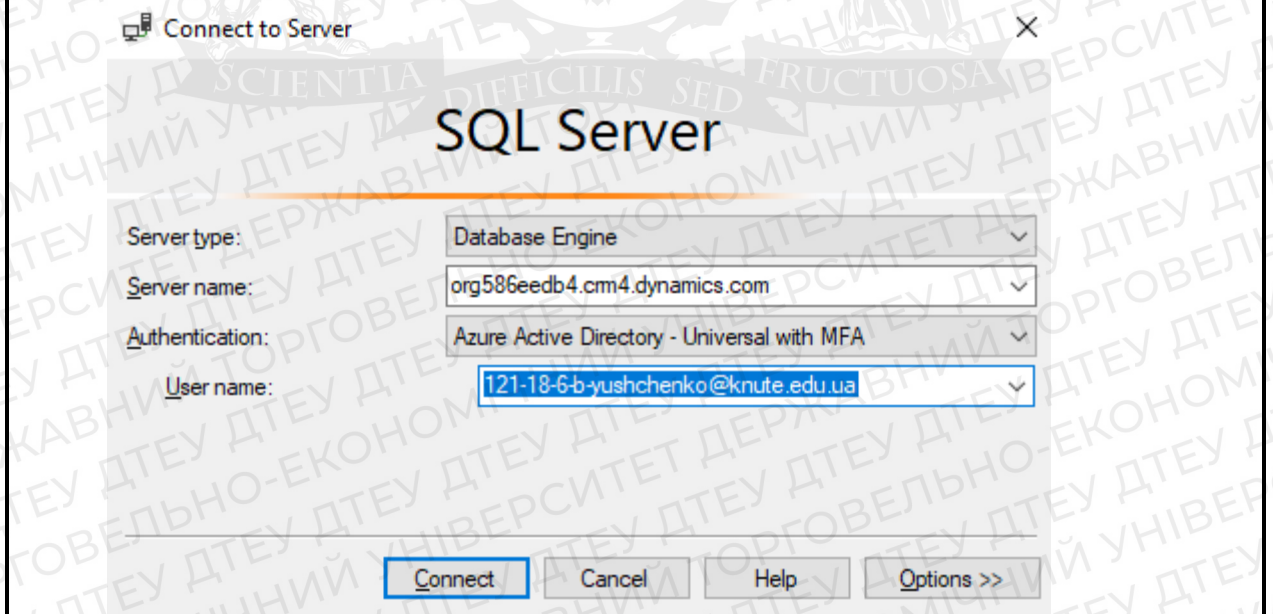


Рис. 3.17. Вхід до серверу БД

					Аркуш
					ДТЕУ 121 023-19.МР
Зм.	Аркуш	№ докум	Підпис	Дата	75

Для цього необхідно перейти у програму SSMS та провести аутентифікацію з наступними параметрами:

- Server Type – Database Engine
- Server Name - org586eedb4.crm4.dynamics.com
- Authentication – Azure Active Directory – Universal with MFA
- User Name – ваш логін у корпоративному просторі ДТЕУ.

3.7. Висновок до розділу 3

Загалом, у цьому розділі були детально розглянуті етапи реалізації програмного додатку інформаційної системи. Автор провів попередні приготування, створення таблиць у базі даних та налаштування форм і подань. Окрім цього, було створено програмний додаток на основі моделі компонентів.

Особливу увагу автор приділив вибору програмної платформи для реалізації. Він використав PowerApps – потужну платформу, що надає широкі можливості для створення багатофункціональних програмних додатків. Завдяки цій платформі, було забезпечено зручну та ефективну розробку програмного додатку.

Також, автор успішно підключив програмний додаток до бази даних, використовуючи Microsoft Dataverse. Ця база даних забезпечує надійне зберігання та керування даними, і її використання дало можливість забезпечити стабільну та безпечну роботу програмного додатку.

Результатом розділу є успішна реалізація програмного додатку, відповідно до вимог та специфікацій проекту. Цей етап розробки є важливим кроком у створенні функціональної та надійної інформаційної системи.

						Аркуш
					ДТЕУ 121 023-19.МР	76
Зм.	Аркуш	№ докум	Підпис	Дата		

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В ході проведення дослідження було проаналізовано сучасні виклики та тенденції в освітньому середовищі вищого закладу освіти, виявлено, що швидкий технологічний розвиток, динамічні зміни в суспільстві та глибока глобалізація національних освітніх систем приводить до обов'язкової адаптації освітнього процесу до цих змін. Нові виклики, такі як інтеграція технологій, дистанційна форма навчання, активна освіта та розвиток навичок, перетворюють систему у більш динамічний та інтерактивний режим функціонування. Перехід до нової архітектури може призвести до численних переваг, таких як покращення якості та ефективності освітнього процесу, більш ефективне управління ЗВО та підвищення комфорту здобувачів у навчанні.

При проектуванні програмно-апаратної архітектури, було зосередження на ключових компонентах реалізації інформаційної системи, а також на розробці програмного додатку для оптимізації освітнього процесу. Проектування архітектури враховувало не лише вимоги до функціональності, а й важливість апаратного забезпечення.

З метою створення структури програмного додатку інформаційної системи, розподілу функцій між класами та забезпечення його модульності й можливості розширення, запроєктовано архітектуру класів програмного додатку. Архітектура моделі компонентів програмного додатку, описує структуру програми з погляду її компонентів. Дана архітектура допомагає забезпечити розподіл функціональності між компонентами та їхню відповідну взаємодію.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 023-19.МР</i>			
Зав. каф.		Криворучко О.В.			Архітектура інформаційної системи освітнього процесу ЗВО	Стадія	Аркуш	Аркушів
Керівник		Харченко О.А.				ВП	77	82
Гарант		Котенко Н.О.				Факультет інформаційних технологій		
Розробив		Шестак Я.І.				2м курс, 23 група		
					<i>Висновки та пропозиції</i>			

Також змодельовано архітектуру діаграми діяльності програмного додатку, яке надає наочну ілюстрацію послідовності подій, що відбуваються в програмі.

Проектування бази даних програмного додатку інформаційної системи освітнього процесу ЗВО було важливим кроком у розробці системи. Під час цього процесу були ретельно розглянуті та враховані особливості освітнього процесу, а також потреби користувачів. Структура бази даних спроектована з метою ефективного зберігання, оновлення та доступу до інформації, що підтримує основні функції системи. Інтеграція бази даних з іншими компонентами інформаційної системи, масштабованість були також враховані, щоб забезпечити надійність та продуктивність інформаційної системи освітнього процесу ЗВО.

Обрання відповідного апаратного забезпечення для ефективного функціонування інформаційної системи освітнього процесу ЗВО відіграє важливу роль у забезпеченні ефективності та надійності її функціонування. Автором запроєктовано модель апаратної архітектури інформаційної системи.

Розроблено програмний додаток, на основі запроєктованих архітектурних моделей. Обрана програмна платформа PowerApps, для реалізації даного додатку, підкреслює відповідність платформи вимогам проекту і забезпечує необхідний функціонал. Підключення до бази даних за допомогою Microsoft Dataverse дозволило створити стійкий і надійний інструмент для управління інформацією. Ця розробка є ключовим кроком у створенні функціональної та ефективної інформаційної системи для закладу вищої освіти.

						ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			78

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Биков В.Ю. Проблеми та перспективи інформатизації системи освіти в Україні / В.Ю. Биков // Науковий часопис НПУ імені М.П. Драгоманова. Серія №2. Комп'ютерно-орієнтовані системи навчання. – К.: НПУ імені М.П. Драгоманова, 2012. – № 13 (20). – С. 3 – 18.
2. Управління розвитком складних систем:(КНУБА) УДК 004.94:378.4, Моделювання єдиного інформаційного простору закладу вищої освіти, Шестак Ярослав, ст 82-89, DOI: dx.doi.org\10.32347/2412-9933.2022.49
3. Міжнародний науково-практичний журнал «Товари і ринки»: (КНТЕУ) (№1 2021): УДК 004.7.056.5(477)(045), Кібербезпека та захист інформації під час пандемії COVID-19, Білявська Юлія, Микитенко Неля, Шестак Ярослав, ст 34-46, DOI: [https://doi.org/10.31617/10.31617/tr.knute.2021\(37\)03](https://doi.org/10.31617/10.31617/tr.knute.2021(37)03)
4. Міжнародний науково-практичний журнал «Товари і ринки»: (ДТЕУ) (№3 2022): УДК 004.056:004.9, Кібербезпека та кібергігієна: нова ера цифрових технологій, Білявська Юлія, Шестак Ярослав, ст 47-59, DOI: [https://doi.org/10.31617/3.2022\(43\)04](https://doi.org/10.31617/3.2022(43)04)
5. Пилипчук А. Ю. Система освіти як об'єкт інформатизації: структура системи освіти. [Електронний ресурс] / Пилипчук А. Ю // Інформаційні технології і засоби навчання. – 2008. – №4. – Режим доступу: <http://www.nbuu.gov.ua/e-journals/ITZN/em8/emg.html>

<i>ДТЕУ 121 023-19.МР</i>					
Зм.	Аркуш	№ докум.	Підпис	Дата	
Зав. каф.		Криворучко О.В.			
Керівник		Харченко О.А.			
Гарант		Котенко Н.О.			
Розробив		Шестак Я.І.			
Архітектура інформаційної системи освітнього процесу ЗВО					
			Стадія	Аркуш	Аркуши
			СВД	79	82
Список використаних джерел					
Факультет інформаційних технологій 2м курс, 2з група					

6. Інформаційні системи в сучасному бізнесі : навчальний посібник / В. С. Пономаренко, І. О. Золотарьова, Р. К. Бутова та ін. – Х. : Вид. ХНЕУ, 2011. – 484 с.
7. Берко А.Ю. Системи баз даних та знань. Книга 2. Системи управління базами даних та знань / А.Ю. Берко, О.М. Верес, В.В. Пасічник. – Львів: «Магнолія-2006», 2015. – 470с.
8. Гайна Г.А. Основи проектування баз даних: Навчальний посібник. – К.: КНУБА, 2016. – 204 с.
9. Харрінгтон Дж. Проектирование реляционных баз данных / Дж. Харрінгтон. – К. : «Лори», 2016. – 230 с.
10. Тарасов, О. В. Використання мови SQL для роботи з сучасними системами керування базами даних. Практикум з навчальної дисципліни "Організація баз даних та знань" : навч.-практ. посіб. / О. В. Тарасов, М. Ю. Лосев, В. В. Федько. – Харків : ХНЕУ, 2013. – 347 с.
11. Федько, В. В. Організація баз даних та знань : навч.-практ. посіб. для самот. підготов. студ. / В. В. Федько, О. В. Тарасов, М. Ю. Лосев. – Харків : ХНЕУ, 2013. – 198 с.
12. Кавун, С. В. Інформаційна безпека : підручник / С. В. Кавун. – Харків : ХНЕУ, 2009. – 368с.
13. Кормич, Б. А. Інформаційна безпека: організаційно-правові основи : навч. посібник для студентів вузів / Б. А. Кормич. – К. : Кондор, 2004. – 384с.
14. Анісімов А.В. Інформаційні системи та бази даних: навчальний посібник / А.В. Анісімов, П.П. Кулябко. – Київ: КНУ, 2017. – 110 с.
15. Toby Velte Cloud Computing: A Practical Approach /Toby Velte, Anthony Velte, Robert C. Elsenpeter. - McGraw Hill Professional, Inc., 2009. – 400 p.

					<i>ДТЕУ 121 023-19.МР</i>	Аркуш
						80
Зм.	Аркуш	№ докум	Підпис	Дата		

16. Chandrasekara C. Hands-On Functional Test Automation: With Visual Studio 2017 and Selenium Paperback / Chaminda Chandrasekara, Pushpa Herath—Apress: May 2, 2019. – 252 p.
17. Cocchiario C. Selenium Framework Designin Data-Driven Testing: Builddata-driven test framework susing Selenium Web Driver, Appium Driver, Java, and Test NG Paperback/ Carl Cocchiario – Packt Publishing: January 23, 2018. – 354 p.
18. Duckett, Jon (2014). HTML and CSS: Design and Build Websites. Indianapolis: John Wiley & Sons.
19. Nixon, Robin (2015). Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. USA, Sebastopol: O’Reilly Media, Inc.
20. Delisle, Marc (2014). Mastering phpMyAdmin 3.3.x for Effective MySQL Management. Packt Publishing. p. 359
21. 2. ISO/IEC 15288 Systems and software engineering - System life cycle processes. – [Чинний від 2008-03-18] – 70 с. (міжнародний стандарт).
22. 3. M.C. Paulk, C.V. Weber, B. Curtis, M.B. Chrissis et al The Capability Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley, Boston. 2015. – 456 p.

					<i>ДТЕУ 121 023-19.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		81

ТЕХНІЧНЕ ЗАВДАННЯ

Система повинна мати модульну структуру, складатися з підсистем, таких як модуль управління студентами, модуль планування курсів, модуль обліку успішності тощо. Кожна підсистема повинна виконувати свої функції та взаємодіяти з іншими підсистемами.

Вимоги до способів і засобів інформаційного обміну між компонентами системи механізмами та протоколів для передачі даних між підсистемами, наприклад, використання веб-сервісів, REST API тощо.

Вимоги до режимів функціонування системи: система повинна бути доступною для користувачів 24/7, забезпечувати швидкий доступ до інформації і високу продуктивність.

Вимоги до режимів управління системою: можливість налаштування прав доступу для користувачів, можливість встановлення системних налаштувань тощо.

Масштабованість. Програмна частина системи повинна вимагати переробок і доопрацювань зі збільшенням навантаження на систему. Збільшення можливостей має досягатися розширенням апаратної частини (додаванням задіяних процесорів, розширенням дискового простору). При цьому між навантаженням та швидкодією системи має бути забезпечена залежність, яку можна спрогнозувати.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 023-19.МР</i>			
Зав. каф.		Криворучко О.В.			Архітектура інформаційної системи освітнього процесу ЗВО	Стадія	Аркуш	Аркушів
Керівник		Харченко О.А.				ТЗ	82	88
Гарант		Котенко Н.О.				Факультет інформаційних технологій		
Розробив		Шестак Я.І.				2м курс, 2з група		
					<i>Технічне завдання</i>			

Вимоги до інтерфейсу користувача: інтуїтивно зрозумілий інтерфейс користувача, який дозволяє зручно та ефективно взаємодіяти з системою.

Вимоги до можливостей настройки інтерфейсу: необхідність користувачам мати можливість налаштовувати інтерфейс системи згідно зі своїми потребами та вподобаннями.

Вимоги до інтерфейсу взаємодії з системою: необхідність системи мати інтерфейси для взаємодії з різними типами користувачів, такими як студенти, викладачі, адміністрація ЗВО.

Надійність і стійкість системи ґрунтуються на надійності її базових програмних компонентів, таких як СУБД, засоби розробки та використовувані технології. Особливо вагомим є достовірність даних, зокрема їх актуальність, цільність і достовірність. Для цієї системи має бути оснащений механізмами, що гарантують ці характеристики. який, необхідно:

Створити модель даних, яка б не містила помилок.

Забезпечити можливість відкату транзакцій у разі помилок або неможливості завершити транзакцію правильно.

Відмовостійкість є важливою характеристикою системи. Це означає, що система повинна бути здатна швидко відновити свою працездатність у випадку аварій, збоїв чи втрати даних. Для досягнення відмовостійкості використовують наступні підходи:

Регулярне резервування та створення копій даних.

Використання резервних серверів.

Застосування розподіленої структури бази даних.

Гнучкість системи є важливою характеристикою. Вона забезпечує можливість пошуку, модифікації та адаптації систем для виконання конкретних задач організації, навіть які можуть виникнути несподівано під час розробки. Цю гнучкість можна досягти завдяки таким якостям системи, як налаштованість, модульність та відкритість.

						ДТЕУ 121 023-19.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			83

Налаштованість представляє собою здатність системи розширювати свій функціонал без необхідності проводити глибокі зміни в її структурі.

Модульність забезпечується відділеною підсистемою в системі з чіткими та описаними інтерфейсами між ними. На більш низькому рівні досягається розділення модулів на функціональні одиниці або інші програмні компоненти. Розробка системи на функціональні модулі дозволяє розвивати, розширювати або змінювати функції системи, які впливають на будь-який етап обслуговування, незалежно від інших. Модульність є основою для створення переносної системи, яка має перевагу в можливості використання технології розділених серверів.

Немонолітність – здатність системи модифікуватися окремими блоками без зупинки всієї системи, трансформація окремих блоків не впливає на роботу всієї інформаційної системи.

Відкритість системи є тим, що формати даних та інтерфейси відповідає певним стандартам. Ця риса дозволяє замінювати одні модулі системи іншими, а також підключати зовнішні системи за використання стандартних протоколів.

Переносність системи порушена в її можливості працювати на різних платформах та з ефективними системами управління базами даних (СУБД). Це забезпечує більшу гнучку та масштабовану роботу системи. Такий підхід дає можливість використовувати більш потужне обладнання чи продуктивність СУБД, що сприяє стійкості системи. Розміщення системи на різних платформах може також сприяти її стійкості до відмов, більшість функцій можуть бути розподілені між повними пристроями та використовувати розподілені або репліковані бази даних.

									Аркуш
									84
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР				

ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

З метою забезпечення якості розробленого програмного забезпечення нами було проведено тестування додатку електронного документообігу.

Для веб-платформи застосовуються наступні види тестування:

Тестування приватного доопрацювання – випробування, спрямовані на тестування приватних доробок платформи, без перевірки повного функціоналу платформи. Якщо приватне доопрацювання має на увазі під собою використання зовнішніх систем, для тестування застосовуються спеціалізовані заглушки, що імітують зовнішні системи.

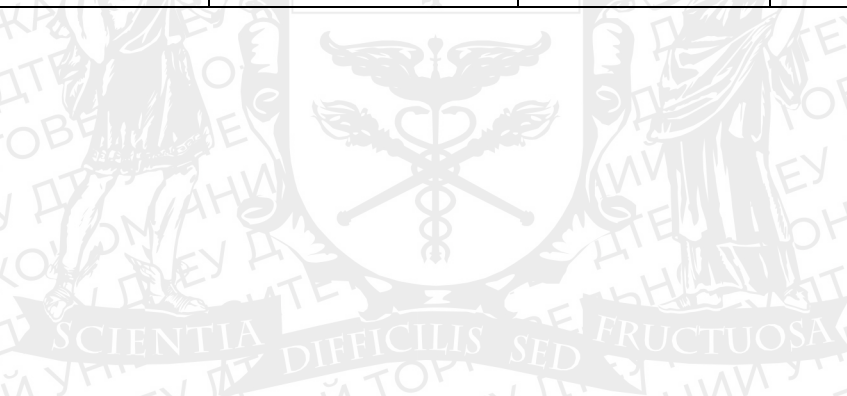
Інтеграційне тестування – випробування, спрямовані на виявлення проблем взаємодії окремих компонентів системи або взаємодії з зовнішніми системами.

Регресійне тестування – випробування, спрямовані на підтвердження коректності поведінки системи в цілому, що має на увазі наскрізну перевірку взаємодії всіх компонент системи і взаємодії із зовнішніми системами.

Тестування навантаження – випробування інформаційної системи в умовах прогнозованої нормального навантаження і стресового навантаження. Під величиною навантаження розуміється кількість призначених для користувача запитів до системи, яке вона повинна встигати обробляти, не перевищуючи певний вихідними вимогами час відгуку.

					<i>ДТЕУ 121 023-19.МР</i>		
Зм.	Аркуш	№ докум.	Підпис	Дата			
Зав. каф.	Криворучко О.В.				Архітектура інформаційної системи освітнього процесу ЗВО	Стадія	Аркуші
Керівник	Харченко О.А.					ПМТ	85 88
Гарант	Котенко Н.О.					Факультет інформаційних технологій 2м курс, 2з група	
Розробив	Шестак Я.І.						
					<i>Програма та методика тестування</i>		

Пов'язане з тест-кейсом вимога	Модуль і підмодуль додатки	Назва (суть) тест-кейса	Очікувані результати
обчислення в обраному користувачем форматі.		форматах txt png ftp	
Опціонально передбачається підтримка декількох мов.	Сканер (обробник помилок)	Запуск програми з іноземної мови	Українська, Англійська,
Швидкість обчислень повинна бути максимально високою.	Реєстратор (консольний реєстратор)	Нагрузити процесор та відкрити програму перевіряючи швидкісну роботу	Швидкість макс при нарузці процесору до 300 Мб/с.
Додаток має запитувати підтвердження («Результат не збережено. Вийти?») В разі, якщо користувач не зберіг результати роботи.	Сканер (обробник помилок)	Вийти з програми з очікуванням діалогового вікна	Має питати у користувача вийти з програми чи ні



									Аркуш
									87
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 023-19.МР				

