

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Інформаційна система управління навчальним процесом освітньої установи
Назва теми

КВРІСТ. 200182.20.01.06 ПЗ
Шифр


Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»
Шифр, назва

Освітня програма «Інформаційні системи та технології»
Назва

Виконав: студент IV курсу, група ІСТ-20-1  Р.Ю. Назар
Підпис Ініціали, прізвище

Керівник  А.О. Нічепорук
Підпис, дата Ініціали, прізвище

Нормоконтролер  І.О. Засорнова
Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем

 Т.О. Говорущенко
Підпис Ініціали, прізвище

«___» _червня_ 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ ” _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Назар Роман Юрійович

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інформаційна система управління навчальним процесом освітньої установи

Керівник проекту (роботи) Нічепорук А.О., доцент кафедри КІІС

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

Затверджена наказом ректора університету від 15.02.2024 р. № 8

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

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

Аналіз відомих засобів та інформаційних систем для навчальної діяльності

Проектування інформаційної системи управління навчальним процесом освітньої

установи. Концептуальна модель даних

Реалізація інформаційної системи управління навчальним процесом освітньої установи





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

Схема бази даних

Креслення структури інформаційної системи та UML діаграми

Креслення вигляду екранних форм

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Засорнова І.О., доцент кафедри КПС		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 11 » 01 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2024	виконано
3	Робота над розділом 1 – аналіз відомих засобів та рішень	01.03.2024	виконано
4	Робота над розділом 2 – Вибір засобів реалізації та проектування інформаційної системи	01.04.2024	виконано
5	Робота над розділом 3 – Реалізація інформаційної системи управління навчальним процесом освітньої установи	30.04.2024	виконано
6	Оформлення пояснювальної записки згідно вимог	10.05.2024	виконано
7	Попередній захист ВКР	26.05.2024	виконано
8	Захист ВКР на засіданні ЕК	Червень 2024 року	

Студент


Підпис

Назар Р. Ю.
Ініціали, прізвище

Керівник проекту (роботи)


Підпис

Нічепорук А. О.
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інформаційна система управління навчальним процесом освітньої установи».

Автор роботи: *Назар Роман Юрійович*.

Керівник роботи: *Нічепорук Андрій Олександрович*.

Пояснювальна записка: 55 с., 24 рис., 6 табл., 3 дод., 60 джерел.

Графічна частина: 7 презентаційних слайдів

ІНФОРМАЦІЙНА СИСТЕМА, КЕРУВАННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ, БАЗА ДАНИХ

Мета кваліфікаційної роботи: проектування та розробка інформаційної системи для управління навчальним процесом освітньої установи.

Інформаційна система управління навчальним процесом освітньої установи є надзвичайно важливою в сучасному освітньому середовищі. Зростаюча складність та обсяги даних, пов'язані з навчальними програмами, студентами, викладачами, розкладами та іншими аспектами управління, вимагають ефективного та організованого підходу до керування навчальними процесами. Такі системи допомагають автоматизувати та оптимізувати рутинні адміністративні процеси, забезпечуючи зручний доступ до інформації для адміністраторів, викладачів, студентів та інших учасників навчального процесу. Вони сприяють управлінню ресурсами, плануванню розкладу занять, моніторингу академічної успішності студентів, організації екзаменаційних сесій та багато іншого.

Підпис студента

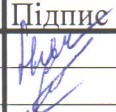





Дата

13.05.2024

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.....	4
ВСТУП.....	5
1 Аналіз відомих засобів та інформаційних систем навчальної діяльності	7
1.1 Інформаційні системи. Концепції та архітектура	7
1.2 Огляд відомих рішень для забезпечення освітньої діяльності навчальної установи.....	12
1.3 Процес проектування інформаційних систем	20
1.4 Висновки. Постановка задачі	23
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ ОСВІТНЬОЇ УСТАНОВИ. КОНЦЕПТУАЛЬНА МОДЕЛЬ ДАНИХ	24
2.1 Програмні засоби для реалізації інформаційної системи управління навчальним процесом освітньої установи	24
2.2 Визначення вимог до проєктованої інформаційної системи	29
2.3 Проектування концептуальної моделі бази даних.....	30
Висновки	42
3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ ОСВІТНЬОЇ УСТАНОВИ.....	43
3.1 Архітектура інформаційної системи управління навчальним процесом освітньої установи.....	43
3.2 Реалізація серверної частини	45
3.3 Реалізація клієнтської частини	49
3.4 UML діаграм для візуалізація роботи інформаційної системи	54
Висновки	58
ВИСНОВКИ.....	59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	61

КВРІСТ. 200182.20.01.06 ПЗ								
					Інформаційна система управління навчальним процесом освітньої установи			
Зм.	Арк.	№докум.	Підпис	Дата		Літера	Аркуш	Аркушів
Виконав		Назар Р.Ю.					2	62
Перевір.		Нічепорук А.О.				ХНУ, ІСТ-20-1		
Н.контр.		Засорнова І.О.						
Затвер.		Говоруценко Т.О.						

ДОДАТОК А Копія креслення «Екранні форми».....	67
ДОДАТОК Б Копія креслення «Схема бази даних».....	68
ДОДАТОК В Копія креслення «Схема структурна варіантів використання».....	69

					КВРІСТ. 200182.20.01.06 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		3

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – База даних

ІС – Інформаційна система

ІТ – Інформаційні технології

НУ – Навчальна установа

ООТП – Об'єктно-орієнтовані технології програмування

СУБД – Система управління базами даних

CORS – (Cross-Origin Resource Sharing)

CRUD – Create, read, update, delete

UI – User Interface

					КВРІСТ. 200182.20.01.06 ПЗ	Арк.
						4
Зм.	Арк.	№докум.	Підпис	Дата		

ВСТУП

В еру інформатизації, яка охопила практично кожну сферу життя, освіта не залишається осторонь. Сучасні технології відкривають перед освітніми установами безліч можливостей для покращення процесів управління та навчання. Інформаційні системи стають ключовим інструментом у сфері освіти, допомагаючи вирішувати виклики та вдосконалювати якість навчання. Впровадження цих систем стає необхідним кроком для модернізації освітнього процесу, забезпечуючи більш ефективне використання ресурсів та забезпечення потреб всіх учасників освітньої спільноти.

Інформаційна система управління навчальним процесом освітньої установи є надзвичайно важливою в сучасному освітньому середовищі. Зростаюча складність та обсяги даних, пов'язані з навчальними програмами, студентами, викладачами, розкладами та іншими аспектами управління, вимагають ефективного та організованого підходу до керування навчальними процесами. Такі системи допомагають автоматизувати та оптимізувати рутинні адміністративні процеси, забезпечуючи зручний доступ до інформації для адміністраторів, викладачів, студентів та інших учасників навчального процесу. Вони сприяють управлінню ресурсами, плануванню розкладу занять, моніторингу академічної успішності студентів, організації екзаменаційних сесій та багато іншого.

Таким чином тема інформаційна система управління навчальним процесом освітньої установи є актуальною у сучасному освітньому середовищі, що продиктовано наступними факторами:

– ефективне управління ресурсами: За допомогою інформаційних систем можна оптимізувати використання ресурсів, таких як час викладачів, аудиторії, матеріальні ресурси та інше. Це дозволяє забезпечити ефективний розподіл ресурсів і підвищити продуктивність навчального процесу;

– зручний доступ до інформації: Інформаційні системи дозволяють зручно та швидко отримувати доступ до різноманітної інформації про розклад занять,

					КвРІСТ. 200182.20.01.06 ПЗ	Арк. 5
Зм.	Арк.	№докум.	Підпис	Дата		

рейтинги студентів, вільні аудиторії, плани занять та інше. Це сприяє підвищенню доступності та прозорості навчального процесу для всіх учасників;

– підвищення якості освіти: Інформаційні системи дозволяють збирати та аналізувати дані про навчальний процес, що допомагає ідентифікувати слабкі місця та ефективніше управляти навчальним процесом. Це сприяє підвищенню якості освіти та досягненню кращих результатів у навчанні;

– адаптація до сучасних тенденцій: В умовах швидких змін у сфері технологій та освіти, інформаційні системи дозволяють освітнім установам бути більш гнучкими та адаптивними до нових вимог і тенденцій. Вони дозволяють швидко впроваджувати нові технології та методики навчання.

Метою роботи є проектування та розробка інформаційної системи для управління навчальним процесом освітньої установи

Об'єкт дослідження є процеси організації обліку іспитів, груп, студентів, предметів та викладачів освітньої установи.

Предмет дослідження є інформаційна система для управління навчальним процесом освітньої установи.

					КВРІСТ. 200182.20.01.06 ПЗ	Арк.
						6
Зм.	Арк.	№докум.	Підпис	Дата		

1 АНАЛІЗ ВІДОМИХ ЗАСОБІВ ТА ІНФОРМАЦІЙНИХ СИСТЕМ НАВЧАЛЬНОЇ ДІЯЛЬНОСТІ

1.1 Інформаційні системи. Концепції та архітектура

Інформаційні системи є ключовим елементом в сучасному світі, де вони використовуються для збору, обробки, збереження та передачі інформації для підтримки різних бізнес-процесів та функцій. Можна виділити наступні концепції, що пов'язані із інформаційними системами:

– цільове спрямування: Ціль інформаційної системи полягає в забезпеченні підтримки для конкретної мети або функції, будь то автоматизація бізнес-процесів, обробка та аналіз даних, взаємодія з клієнтами тощо;

– архітектура системи: Це структурна організація і взаємозв'язки між компонентами інформаційної системи, включаючи апаратне забезпечення, програмне забезпечення, бази даних, інтерфейси користувача та інші;

– збір, обробка та збереження даних: Ця концепція визначає процеси збору даних з різних джерел, їх обробку та аналіз, а також зберігання відповідних даних для майбутнього використання;

– взаємодія з користувачами: Інформаційні системи повинні мати інтерфейси, які дозволяють користувачам легко взаємодіяти з системою та виконувати необхідні завдання;

– безпека та захист даних: Ця концепція визначає заходи, які приймаються для захисту інформації від несанкціонованого доступу, втрати або пошкодження;

– аналітика та звітність: Інформаційні системи можуть надавати можливості для аналізу даних та створення звітів для прийняття рішень на основі зібраної інформації.

Ці концепції допомагають розуміти різні аспекти функціонування та розвитку інформаційних систем, а також визначають їхню важливість для підтримки бізнес-процесів та завдань користувачів.

В загальному вигляді інформаційні системи включають наступні процеси та компоненти (рисунок 1.1):

Введення даних:

- зовнішні джерела: Інформація може надходити з різних джерел, таких як Інтернет, датчики, інші інформаційні системи або від користувачів;
- внутрішні джерела: Це дані, які вже знаходяться в межах системи, такі як бази даних, файли або документи, що були зібрані раніше.

Зберігання даних:

- бази даних: Дані можуть бути збережені у різних типах баз даних, таких як SQL, NoSQL, XML або JSON;
- файли: Це різні типи файлів, включаючи текстові, табличні, зображення, аудіо та відео файли.

Обробка даних:

- очищення: Включає видалення помилок, дублікатів та неповноцінних даних;
- трансформація: Зміна формату, структури або значення даних для оптимізації їх використання;
- аналіз: Виявлення закономірностей, тенденцій та прогнозів на основі оброблених даних.

Видача інформації:

- звіти: Представлення даних у вигляді текстових, табличних або графічних звітів;
- візуалізація: Представлення даних у вигляді діаграм, карт або графіків для легшого сприйняття;
- інтерфейс: Надання доступу до інформації через веб-сайти, мобільні додатки або API для користувачів.

Зворотний зв'язок:

- користувачі: Надають оцінку, відгуки та пропозиції, що можуть бути використані для поліпшення системи;

– система: Вдосконалюється на основі зворотного зв'язку від користувачів для покращення їх задоволення від використання.

Додаткові компоненти:

– мережа: Включає доступ до Інтернету або локальної мережі для обміну даними між різними компонентами системи;

– безпека: Захист даних від несанкціонованого доступу та збереження конфіденційності даних;

– адміністрування: Управління системою, включаючи керування користувачами, даними та іншими аспектами функціонування.

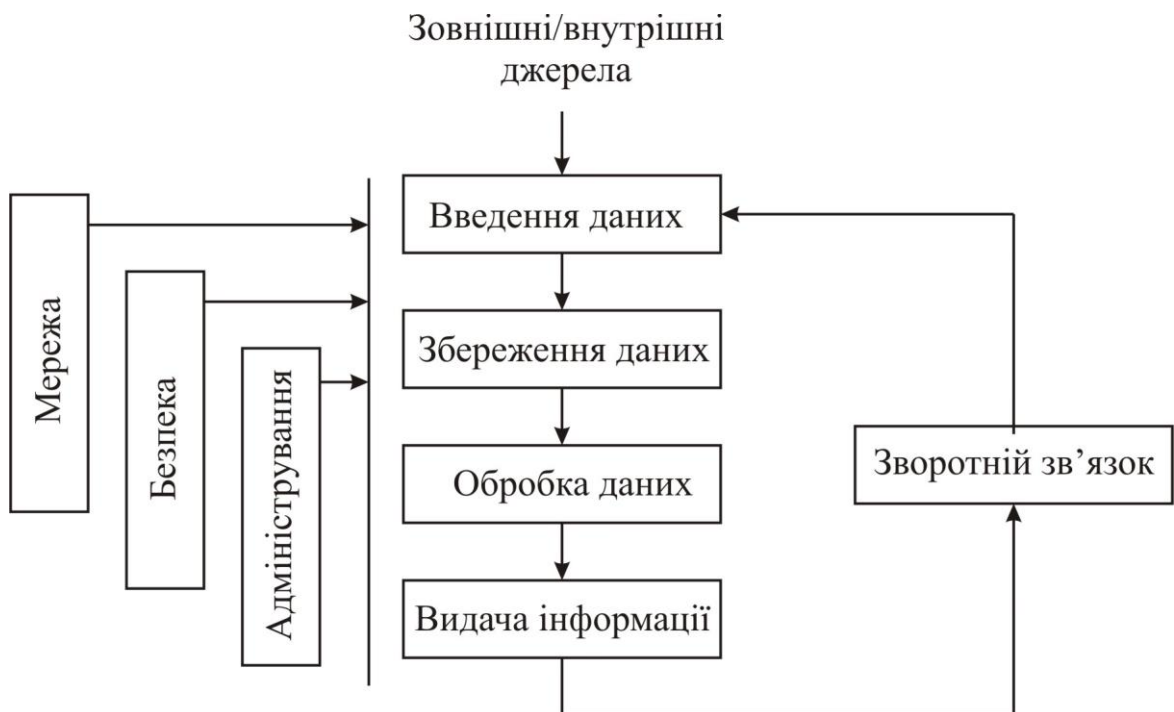


Рисунок 1.1 – Процеси та компоненти інформаційної системи

Щодо типу системної архітектури, які зазвичай використовуються в областях програмного забезпечення та інформаційних технологій, то існує кілька типів, зокрема монолітна, клієнт-серверна та мікросервісна архітектури.

Монолітна архітектура: у цій архітектурі вся програма створена як єдине самодостатнє ціле. Усі компоненти та функції тісно пов'язані та розгорнуті разом. Монолітна архітектура є відносно простою та підходить для невеликих додатків,

але її може стати складно підтримувати та масштабувати у міру зростання системи.

Архітектура клієнт-сервер: ця архітектура передбачає поділ системи на два основні компоненти: клієнти та сервери (рисунок 1.2). Клієнти – це пристрої кінцевих користувачів, які запитують послуги від серверів, які надають ресурси або виконують певні завдання. Архітектура «клієнт-сервер» забезпечує розподілену обробку, масштабованість і розділення проблем між клієнтською та серверною логікою.

Сервісно-орієнтована архітектура (SOA): SOA – це архітектурний стиль, який зосереджується на розробці систем як набору слабо пов'язаних, сумісних сервісів. Кожна служба виконує певну бізнес-функціональність і може бути незалежно розроблена, розгорнута та використана. SOA сприяє повторному використанню, гнучкості та сумісності між різними компонентами та системами.

Мікросервісна архітектура: Архітектура мікросервісів є розширенням концепції SOA, де система розділена на набір невеликих незалежних сервісів. Кожна служба представляє певну бізнес-можливість і взаємодіє з іншими службами за допомогою спрощених протоколів, таких як HTTP/REST. Мікросервіси забезпечують незалежне масштабування, легкість розгортання та ізоляцію помилок.

Архітектура, керована подіями (EDA): EDA – це архітектурний шаблон, який наголошує на виробництві, виявленні та споживанні подій у системі. Події використовуються для запуску дій або передачі змін між компонентами. Ця архітектура сприяє слабкому зв'язку та масштабованості шляхом роз'єднання компонентів за допомогою зв'язку на основі подій.

Багаторівнева архітектура: багаторівнева архітектура передбачає організацію системи в кілька рівнів, де кожен рівень представляє різний рівень абстракції або функціональності. Як правило, рівні включають презентацію, бізнес-логіку та доступ до даних. Ця архітектура сприяє поділу завдань, модульності та зручності обслуговування.

					КВРІСТ. 200182.20.01.06 ПЗ	Арк. 10
Зм.	Арк.	№докум.	Підпис	Дата		

Однорангова архітектура (P2P): архітектура P2P забезпечує децентралізований зв'язок між вузлами або одноранговими вузлами в мережі. Однорангові партнери можуть виступати як клієнтами, так і серверами, обмінюватися ресурсами та співпрацювати безпосередньо один з одним. Архітектури P2P часто використовуються для обміну файлами, розподілених обчислень або децентралізованих систем.

Архітектура обміну повідомленнями, керована подіями: ця архітектура використовує черги повідомлень або потоки подій для забезпечення асинхронного зв'язку та роз'єднання між компонентами. Повідомлення або події надсилаються між різними компонентами для ініціювання дій або обміну інформацією. Це забезпечує масштабованість, відмовостійкість і слабкий зв'язок.

Гібридні архітектури: багато реальних систем використовують комбінацію різних архітектурних стилів для задоволення конкретних вимог. Гібридні архітектури використовують сильні сторони кількох архітектур для досягнення бажаних функцій, таких як поєднання мікросервісів із керованою подіями або багаторівневою архітектурою.

Важливо зауважити, що це загальні типи архітектури, і можуть бути варіації та комбінації на основі конкретних вимог і використовуваних технологій. Кожна архітектура має свої переваги та міркування, і вибір залежить від таких факторів, як потреби в масштабованості, складність системи, вимоги до сумісності та досвіду команди розробників.

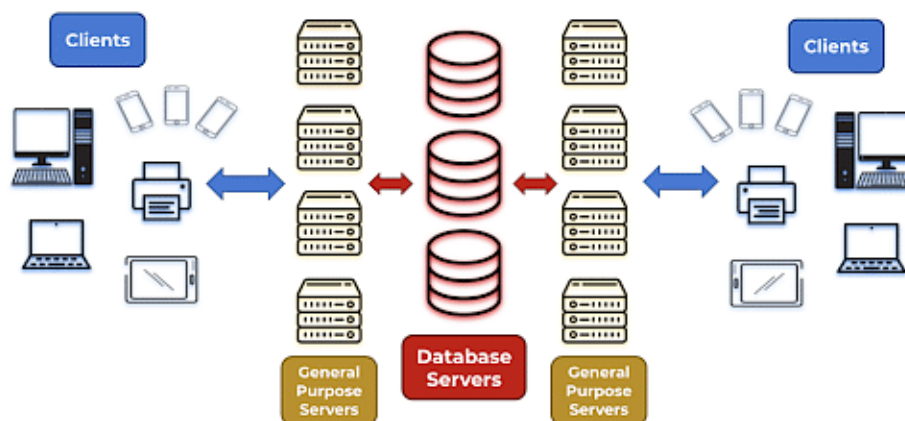


Рисунок 1.2 – Клієнт-серверна архітектура

Зм.	Арк.	№докум.	Підпис	Дата

1.2 Огляд відомих рішень для забезпечення освітньої діяльності навчальної установи

Освітні платформи – це онлайн-ресурси або програмні засоби, які надають можливість навчання та освіти через Інтернет. Вони забезпечують доступ до різноманітних навчальних матеріалів, курсів, відеоуроків, тестів, інтерактивних завдань та інших засобів для вивчення різних предметів і навичок. Освітні платформи можуть бути спрямовані на різні аудиторії, включаючи студентів, викладачів, професіоналів, дітей та дорослих. Вони можуть пропонувати навчальні курси з різних предметів, включаючи науку, технології, інженерію, мистецтво, мови, бізнес, та інші.

Освітні платформи дозволяють користувачам навчатися в зручній для них час і місце, забезпечуючи гнучкість і доступність навчальних ресурсів. Вони часто використовують інтерактивні технології, адаптивні методи навчання та індивідуалізовані підходи, щоб забезпечити ефективне засвоєння матеріалу. Деякі освітні платформи також надають можливість спілкування та співпраці між користувачами, обміну досвідом та знаннями, а також підтримують процес оцінювання навчальних успіхів.

На сьогоднішній день існують різноманітні системи управління навчальним процесом, які використовуються для відображення інформації про студентів, викладачів, іспити та інші аспекти навчального закладу. Ось деякі з найбільш поширених систем:

– Moodle: Це одна з найпопулярніших відкритих платформ для управління навчанням, яка надає засоби для створення онлайн-курсів, організації навчальних матеріалів, проведення тестів та іспитів, а також взаємодії між студентами та викладачами;

– Blackboard Learn: Ця платформа пропонує широкий спектр інструментів для організації навчального процесу, включаючи завдання, обговорення, онлайн-конференції та інші функції спілкування;

					КвРІСТ. 200182.20.01.06 ПЗ	Арк. 12
Зм.	Арк.	№докум.	Підпис	Дата		

– Canvas: Це ще одна популярна система управління навчанням, яка надає інструменти для створення курсів, виконання завдань, оцінювання студентів та спілкування в межах навчальної спільноти;

– Schoology: Ця платформа спрямована на освітні заклади і надає інструменти для організації навчальних матеріалів, спілкування, спільної роботи та відстеження успішності студентів;

– Google Classroom: Це безкоштовний інструмент від Google, який дозволяє вчителям створювати курси, додавати завдання та матеріали для навчання, а також взаємодіяти зі своїми учнями через онлайн-платформу.

Розглянемо детальніше відомі платформи для організації навчання в освітній установі.

Однією із популярних платформ є Moodle (рисунок 1.3). Moodle – платформа із відкритим вихідним кодом для створення навчального середовища (онлайн-курсів, дистанційного навчання, віддаленого доступу до навчальних файлів, обміну повідомленнями).

Модульне середовище для навчання

Категорії курсів

► Розгорнути всі

- Матеріали для розробників курсів ⁽⁵⁾
- Освітні програми
- Англomовні освітні програми ⁽⁵⁾
- Кафедра автоматизації, комп'ютерно-інтегрованих технологій та робототехніки
- Кафедра архітектури та містобудування ⁽⁶¹⁾
- Кафедра будівництва та цивільної безпеки ⁽⁵¹⁾
- Кафедра вищої математики і комп'ютерних застосувань ⁽²⁸⁾
- Кафедра галузевого машинобудування та агроінженерії
- Кафедра германської філології та перекладознавства ⁽⁶³⁾
- Кафедра дизайну ⁽¹⁾
- Кафедра екології та біологічної освіти ⁽¹¹³⁾
- Кафедра економіки, аналітики, моделювання та інформаційних технологій в бізнесі ⁽¹⁵⁷⁾



Хмельницький
національний
університет



Co-funded by the
Erasmus+ Programme
of the European Union



МСН працює на серверах, придбаних за
кошти проекту Еразмус+ QUAREE

Головне меню

- Новини сайту
- Реєстрація
- Рекомендації по роботі в МСН
- Куди і як слід завантажувати контрольні роботи (для заочників)
- Система тестування абітурієнтів
- Інформаційна система "Електронний університет"

Рисунок 1.3 – Платформа moodle

Зм..	Арк.	№докум.	Підпис	Дата

КВРІСТ. 200182.20.01.06 ПЗ

Арк.
13

Основні можливості Moodle включають:

– створення онлайн-курсів. Текстові документи, презентації та відео в Moodle можна об'єднати у навчальний курс, який буде доступний усім учням або окремому класу. Тестування. У Moodle вбудовано редактор тестів. За замовчуванням є 15 типів завдань: від вибору однієї правильної відповіді до перетягування об'єктів. Щоб учні не списували, можна обмежити час на вирішення тесту та кількість спроб. Система автоматично перевіряє відповіді, показує допущені помилки та вказує набраний бал;

– форум та коментарі. Щоб зв'язатися з викладачем, поставити запитання чи обговорити тему уроку, учні можуть залишати коментарі під курсами або починати розмови на вбудованому форумі;

– база знань – це архів навчальних матеріалів, що цілодобово доступний всім користувачам. У будь-який момент учні можуть зайти до бази знань та знайти потрібну доповідь, відеоурок чи статтю;

– мобільне навчання. У сервісу є мобільний додаток Moodle Mobil, який дозволяє проходити курси та вирішувати тести з планшета чи смартфона;

– статистика навчання. Moodle відстежує успішність учнів та складає звіти для викладачів. Наприклад, показує, скільки часу студенти проходили курс, яких помилок припустилися в тесті, кому подобається вчитися, а кому ні.

Ще однією платформою для керування навчальною діяльністю є CANVAS (рисунок 1.4).

CANVAS – це веб-система керування навчанням (LMS) Каліфорнійського державного університету Фуллертон (CSUF). Викладацький склад CSUF може використовувати CANVAS для керування своєю діяльністю в Інтернеті для традиційного очного, змішаного навчання, повністю онлайн-курсів і дистанційного навчання (синхронно через Zoom, наприклад), а також для інших проектів електронного навчання та співпраці викладачів. CANVAS забезпечує зручне середовище з кількома інструментами, які викладачі та студенти можуть використовувати для навчання, обміну інформацією та взаємодії один з одним.

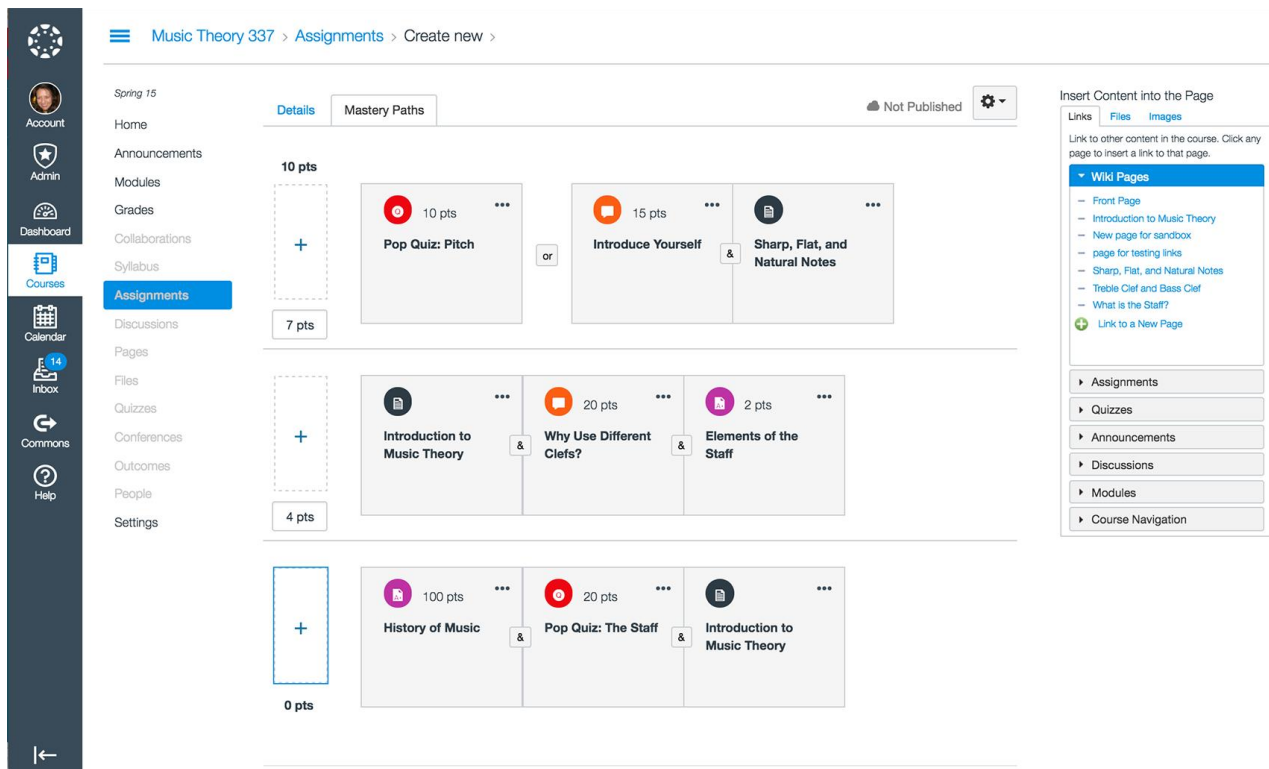


Рисунок 1.4 – Платформа canvas

Основні особливості Canvas включають:

- створення курсів: Викладачі можуть легко створювати нові курси в системі, додавати матеріали для навчання, завдання та інші ресурси;
- виконання завдань: Студенти можуть переглядати та виконувати завдання, додавати відповіді та відправляти їх для оцінювання викладачем;
- оцінювання студентів: Canvas надає зручний інтерфейс для оцінювання робіт студентів, виставлення балів та надання фідбеку;
- спілкування: Система дозволяє викладачам та студентам взаємодіяти між собою через обговорення, чати та повідомлення;
- співпраця: Canvas також підтримує спільну роботу над проектами, завданнями та іншими матеріалами.

Окрім того, навчальна програма в Canvas дозволяє легко повідомити учням те, що від них буде потрібно протягом курсу в хронологічному порядку (завдання, обговорення, тести тощо). Є можливість додати інформацію про навчальний план на веб-сторінку. Canvas автоматично додасть підсумок курсу на основі термінів виконання завдань і подій у курсі. Усі завдання

Зм.	Арк.	№докум.	Підпис	Дата
-----	------	---------	--------	------

(неопубліковані та опубліковані) перераховані в програмі для викладачів. Студенти бачать лише опубліковані завдання.

Ще однією платформою є Google Classroom (рисунок 1.5). Google Classroom – це набір онлайн-інструментів, який дозволяє вчителям ставити завдання, надсилати роботи студентам, виставляти оцінки та повертати оцінені роботи. Він був створений як спосіб відмовитися від паперових документів у класах і перевести навчання у цифровий формат. Спочатку він планувався для використання з ноутбуками в школах, такими як Chromebook, щоб дозволити вчителям і учням ефективніше обмінюватися інформацією та завданнями. Оскільки все більше шкіл перейшли на онлайн-навчання, Google Classroom отримав набагато ширше використання, оскільки вчителі швидко запровадили безпаперове навчання. Google Classroom працює із Google Документами, Таблицями, Презентаціями, Сайтами, Планетою Земля, Календарем і Gmail, а також можуть бути доповнені Google Hangouts або Meet для очного навчання чи запитання.

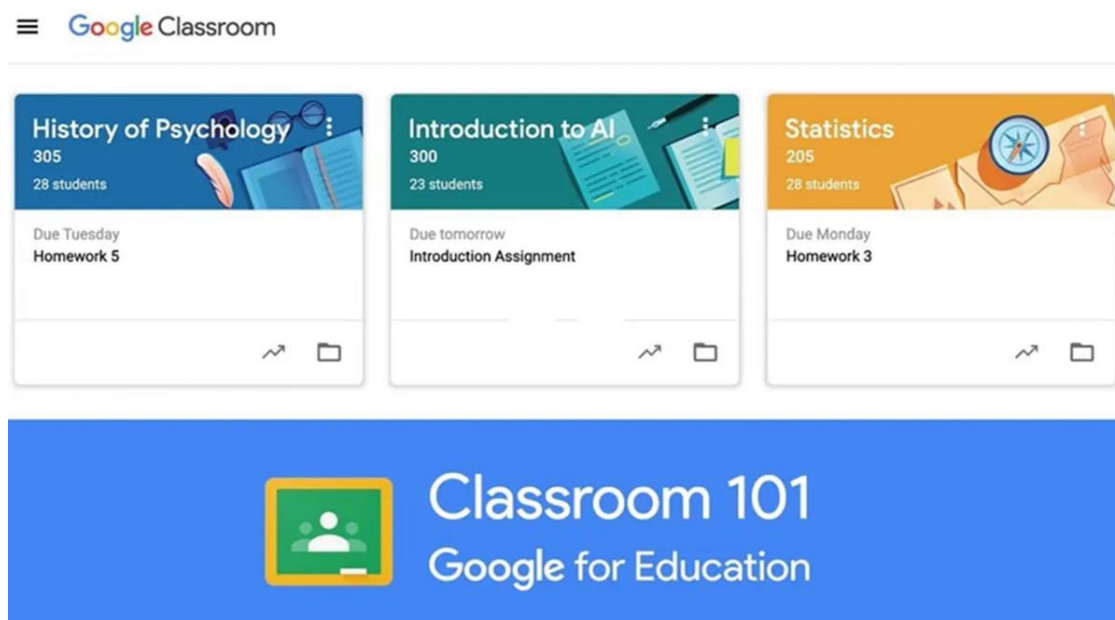


Рисунок 1.5 – Платформа Google Classroom

Іншим рішенням є платформа Schoology (рисунок 1.6). До основних її функцій можна віднести:

					КвРІСТ. 200182.20.01.06 ПЗ	Арк. 16
Зм.	Арк.	№докум.	Підпис	Дата		

– управління курсами та контентом. Schoology дозволяє вчителям легко створювати та управляти онлайн-курсами. Інструктори можуть організовувати контент курсів, такий як уроки, завдання, екзамени та додаткові матеріали, у зручному інтерфейсі. Крім того, Schoology пропонує можливість імпортувати існуючий контент з інших освітніх платформ, що спрощує перехід на нову систему;

– комунікація та співпраця. Schoology сприяє комунікації та співпраці між вчителями та учнями. Користувачі можуть надсилати приватні повідомлення, брати участь у групових обговореннях, ділитися ресурсами та спільно працювати над проектами в реальному часі. Ця функція сприяє взаємодії між учасниками курсу та полегшує вирішення сумнівів та обговорення ідей;

– оцінки та відстеження прогресу. Однією з найбільш помітних особливостей Schoology є його система оцінювання та відстеження прогресу. Вчителі можуть створювати та управляти онлайн-журналами оцінок, призначати бали та коментарі до завдань та екзаменів, а також надавати зворотний зв'язок учням ефективно. Крім того, учні та їх батьки можуть доступатися до своїх оцінок та відстежувати прогрес у будь-який час:

– інтеграція з іншими інструментами та сервісами. Schoology інтегрується з широким спектром освітніх інструментів та сервісів, розширюючи свою функціональність. Наприклад, вчителі можуть легко інтегрувати сторонні додатки, такі як Google Drive та Microsoft Office, для спільного використання та співпраці над документами. Крім того, Schoology дозволяє інтегрувати засоби оцінювання, такі як Turnitin та Proctorio, для забезпечення академічної доброчесності;

– аналіз та звіти. Schoology пропонує інструменти аналізу та звітності, які дозволяють вчителям оцінити успішність учнів та курсів. Вони можуть отримувати детальну інформацію про залученість учнів, час, витрачений на активності, та результати оцінок. Ці звіти допомагають виявити області для вдосконалення та приймати обґрунтовані рішення з оптимізації навчання.

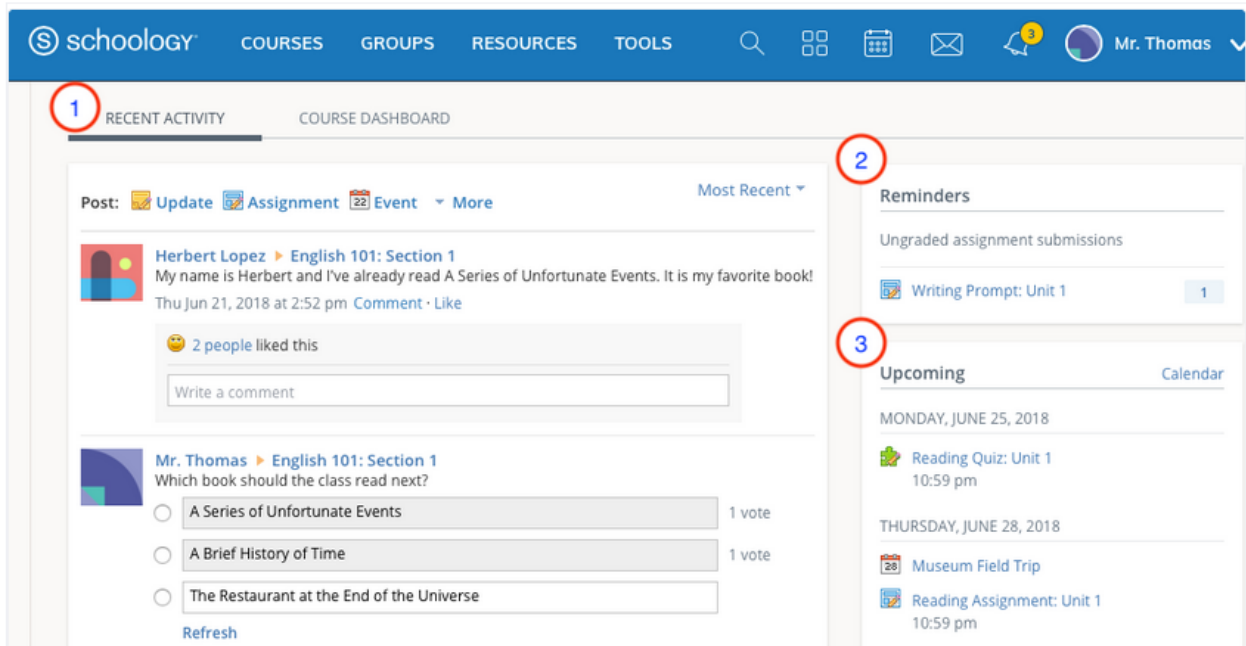


Рисунок 1.6 – Платформа Schoology

Ще однією LMS є Blackboard Learn, яка надає інструменти для віддаленого навчання та організації навчального процесу в університетах, коледжах та школах. Ця система включає в себе широкий спектр функцій, таких як створення курсів, завдань, дискусійних форумів, відеолекцій, онлайн-тестів та інших інструментів для взаємодії вчителів і студентів. Blackboard Learn також надає можливості оцінювання, відстеження прогресу студентів та спілкування в режимі реального часу.

Основні особливості Blackboard Learn включають:

- створення курсів: Викладачі можуть легко створювати онлайн-курси і організовувати матеріали для навчання;
- завдання та тести: Платформа дозволяє викладачам створювати та призначати завдання і тести студентам, а також автоматично оцінювати їхні результати;
- дискусійні форуми: Студенти можуть обговорювати теми курсу в дискусійних форумах, співпрацювати між собою та отримувати відповіді від викладачів;

Таким чином, за допомогою проектованої інформаційної системи є можливість контролювати всі аспекти програми, включаючи функціонал, дизайн та безпеку. Також, є можливість розробити систему так, щоб вона відповідала конкретним потребам заданого навчального закладу. Однак, залежно від обсягу роботи, розробка та підтримка власної системи може вимагати більше часу та зусиль у порівнянні з використанням готових рішень.

1.3 Процес проектування інформаційних систем

Проектування інформаційних систем завжди починається з визначення мети проекту. Основне завдання будь-якого успішного проекту полягає в тому, щоб на момент запуску системи та протягом усього часу її експлуатації можна було забезпечити:

- необхідну функціональність системи і ступінь адаптації до умов її функціонування, що змінюються;
- необхідну пропускну спроможність системи;
- необхідний час реакції системи на запит; безвідмовну роботу системи в необхідному режимі, іншими словами готовність і доступність системи для обробки запитів користувачів;
- простоту експлуатації та підтримки системи;
- необхідну безпеку.

Продуктивність є основним чинником, визначальним ефективність системи. Хороше проектне рішення є основою високопродуктивної системи. Проектування інформаційних систем охоплює три основні сфери:

- проектування об'єктів даних, які будуть реалізовані у базі даних;
- проектування програм, екранних форм, звітів, які забезпечуватимуть виконання запитів до даних;

Це модель або методологія роботи, коли виконання різних етапів проекту або розробки відбувається послідовно, а не паралельно (рисунок 1.7).

У контексті програмування це може виглядати наступним чином:

- аналіз вимог: Отримання і розуміння вимог до системи або програмного забезпечення;
- проектування: Розроблення архітектури та проектування системи або програмного забезпечення на основі вимог;
- розробка: Написання коду на основі проектування;
- тестування: Перевірка роботи програмного забезпечення на відповідність вимогам та виявлення помилок;
- впровадження: Розгортання та запуск програмного забезпечення для використання;
- підтримка: Підтримка та вдосконалення програмного забезпечення після впровадження.

Кожен етап виконується після завершення попереднього, схоже на водопад, де кожен рівень водоспаду спускається до наступного рівня. Ця модель часто використовується в проектуванні систем, які мають статичні вимоги та добре визначені етапи розробки. Однак, у реальних проектах часто використовуються ітеративні методи, такі як Agile, які дозволяють більш гнучкий підхід до розробки.

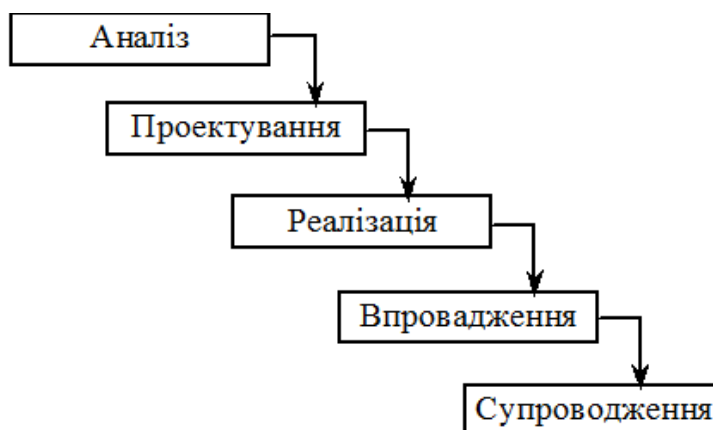


Рисунок 1.7 – Каскадна схема розробки ІС

1.4 Висновки. Постановка задачі

Інформаційна система управління навчальним процесом освітньої установи є надзвичайно важливою в сучасному освітньому середовищі. Зростаюча складність та обсяги даних, пов'язані з навчальними програмами, студентами, викладачами, розкладами та іншими аспектами управління, вимагають ефективного та організованого підходу до керування навчальними процесами. Такі системи допомагають автоматизувати та оптимізувати рутинні адміністративні процеси, забезпечуючи зручний доступ до інформації для адміністраторів, викладачів, студентів та інших учасників навчального процесу. Вони сприяють управлінню ресурсами, плануванню розкладу занять, моніторингу академічної успішності студентів, організації екзаменаційних сесій та багато іншого. Тому метою даної роботи є проектування та розробка прототипу інформаційної системи для управління навчальним процесом освітньої установи, яка має зручний та інтуїтивно зрозумілий інтерфейс із можливістю збереження та опрацювання даних про студентів, викладачів, предмети, іспити, тощо.

Вирішення поставленого завдання потребує виконання наступних кроків:

- 1) виконати аналіз відомих освітніх платформ, що використовуються у освітніх цілях. Визначити їх переваги та недоліки;
- 2) виокремити вимоги до проєктованої інформаційної системи;
- 3) здійснити аналіз предметної області, розробити концептуальну модель бази даних та ERD діаграму;
- 4) вибрати стек технології для реалізації програмного забезпечення інформаційної системи. Визначити структуру інформаційної технології;
- 5) реалізувати інформаційну систему управління навчальним процесом освітньої установи;
- 6) представити функціонування розробленої ІС за допомогою UML діаграм.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ ОСВІТНЬОЇ УСТАНОВИ. КОНЦЕПТУАЛЬНА МОДЕЛЬ ДАНИХ

2.1 Програмні засоби для реалізації інформаційної системи управління навчальним процесом освітньої установи

Для реалізації інформаційної системи управління навчальним процесом освітньої установи використаємо повностекову архітектуру (fullstack), яка включатиме серверну частину, базу даних та користувацький інтерфейс:

– Backend з dotnet core: Використовуючи ASP.NET Core, можна створити потужний та надійний бекенд для інформаційної системи. ASP.NET Core забезпечить обробку запитів від клієнтів, доступ до бази даних та логіку бізнес-процесів;

– Frontend з React.js: React.js використовується для розробки динамічного та інтерактивного користувацького інтерфейсу. За допомогою React.js можна створити зручний та ефективний інтерфейс, який забезпечить швидкий та зручний доступ до функціональності системи для користувачів;

– база даних MS SQL: Microsoft SQL Server використовується для зберігання та керування даними системи. За допомогою MS SQL можна зберігати інформацію про студентів, викладачів, навчальні групи, предмети, розклади, оцінки та інші дані, необхідні для управління навчальним процесом. Розглянемо детальніше обрані засоби реалізації.



Рисунок 2.1 – Взаємодія між програмними засобами у проєктованій інформаційній системі управління навчальним процесом освітньої установи

може легко зрозуміти та почати створювати веб-додатки за допомогою React за кілька днів. Це частина V (перегляд) у моделі MVC (Model-View-Controller), яка називається «однією з фреймворків JavaScript». Він не є повнофункціональним, але має перевагу відкритої бібліотеки інтерфейсу користувача JavaScript (UI), яка допомагає виконувати завдання кращим чином;

– легкість створення динамічних веб-додатків. Створити динамічну веб-програму саме з рядками HTML було складно, оскільки це вимагає складного кодування, але React JS вирішив цю проблему та спростив її. Він забезпечує менше кодування та надає більше функціональних можливостей. Він використовує JSX (розширення JavaScript), який є особливим синтаксисом, що дозволяє використовувати лапки HTML і синтаксис тегів HTML для відтворення окремих підкомпонентів;

– багаторазові компоненти. Веб-програма ReactJS складається з кількох компонентів, і кожен компонент має власну логіку та елементи керування. Ці компоненти відповідають за виведення невеликого багаторазового фрагмента HTML-коду, який можна повторно використовувати будь-де, де вам це потрібно. Багаторазовий код допомагає спростити розробку та підтримку ваших програм. Ці компоненти можна вкладати в інші компоненти, щоб дозволити створювати складні програми з простих будівельних блоків. ReactJS використовує механізм на основі віртуальної DOM для заповнення даними в HTML DOM. Віртуальний DOM працює швидко, оскільки змінює лише окремі елементи DOM замість того, щоб щоразу перезавантажувати повний DOM;

– підвищення продуктивності. ReactJS покращує продуктивність завдяки віртуальному DOM. DOM – це кросплатформний API програмування, який працює з HTML, XML або XHTML. Більшість розробників стикалися з проблемою оновлення DOM, що сповільнювало роботу програми. ReactJS вирішив цю проблему, запровадивши віртуальний DOM. React Virtual DOM існує повністю в пам'яті та є представленням DOM веб-браузера. Через це, коли ми пишемо компонент React, ми не пишемо безпосередньо в DOM. Замість цього ми

пишемо віртуальні компоненти, які реагують на перетворення на DOM, що забезпечує плавнішу та швидшу роботу;

– підтримка підручних засобів. React JS також набув популярності завдяки наявності зручного набору інструментів. Ці інструменти роблять завдання розробників зрозумілими і полегшують. Інструменти розробника React були розроблені як розширення для розробників Chrome і Firefox і дозволяють вам перевіряти ієрархії компонентів React у віртуальному DOM. Це також дозволяє вам вибирати окремі компоненти та перевіряти та редагувати їхні поточні властивості та стан;

– SEO-дружній. Традиційні фреймворки JavaScript мають проблеми з SEO. Пошукові системи, як правило, мають проблеми з читанням програм, у яких багато JavaScript. Багато веб-розробників часто скаржилися на цю проблему. ReactJS долає цю проблему, допомагаючи розробникам легко орієнтуватися в різних пошукових системах. Це тому, що додатки React.js можуть працювати на сервері, а віртуальний DOM відобразатиметься та повертатиметься до браузера як звичайна веб-сторінка;

Ще одним компонентом для проєктованої інформаційної системи є база даних (БД). Вибір бази даних включає багато факторів. Зокрема потрібно визначити, яке навантаження матиме база даних. Яка масштабованість бази даних і зберігання даних. Підтримка, особливо у разі пошкодження бази даних. Наявність функції та додаткових компонентів, тощо. У даній ІС в якості БД було обрано MySQL. Такий вибір був зумовлений наступними перевагами даної СУБД:

– MySQL – реляційна база даних;

– MySQL більш безпечний, оскільки він складається з надійного рівня безпеки даних для захисту конфіденційних даних від зловмисників, а паролі в MySQL зашифровані;

– MySQL доступний для безкоштовного завантаження та використання з офіційного сайту MySQL;

– MySQL сумісний з більшістю операційних систем, включаючи Windows, Linux, NetWare, Novell, Solaris та інші варіанти UNIX;

- MySQL забезпечує можливість запуску клієнтів і сервера на одному або різних комп'ютерах через Інтернет або локальну мережу;
- MySQL має унікальну архітектуру механізму зберігання, що робить його швидшим, дешевшим і надійнішим;
- MySQL дає розробникам більшу продуктивність завдяки використанню представлень, тригерів і збережених процедур;
- MySQL простий і легкий у використанні. Ви можете створювати та взаємодіяти з MySQL, маючи лише базові знання MySQL і кілька простих операторів SQL;
- MySQL має архітектуру клієнт-сервер. Може бути будь-яка кількість клієнтів або прикладних програм, які взаємодіють із сервером бази даних (MySQL) для запиту даних, збереження змін тощо;
- MySQL є масштабованим і здатним обробляти понад 50 мільйонів рядків. Цього достатньо для обробки майже будь-якого обсягу даних. Хоча стандартний розмір файлу становить 4 ГБ, його можна збільшити до 8 ГБ;
- MySQL дозволяє відкочувати транзакції;
- MySQL дуже гнучкий, оскільки підтримує велику кількість вбудованих програм;

Як недолік даної СУБД можна відзначити, що при великих об'ємах даних дана БД працює досить повільно. Проте, даний недолік у проєктованій інформаційній системі є не критичний, адже розмір БД буде не великим.

Таким чином проведено огляд технологій та засобів, що будуть використані для реалізації інформаційної системи управління навчальним процесом освітньої установи.

2.2 Визначення вимог до проєктованої інформаційної системи

Після обраного стеку технологій та засобів реалізації інформаційної системи наступним кроком є визначення вимог, які ставитимуться до

проектованої інформаційній системі управління навчальним процесом освітньої установи.

Головною вимогою щодо проектованої інформаційної системи є забезпечення зручного та ефективного відображення графіку іспитів та доступних аудиторій для викладачів і студентів. Система повинна надавати зрозумілу та легкодоступну інформацію щодо дат та часу проведення іспитів, а також наявності вільних аудиторій для їх проведення. Це дозволить ефективно організувати робочі процеси для викладачів та студентів, спростувати процес планування та зменшувати можливі “накладки” у розкладі занять.

Таким чином проектована інформаційна система повинна забезпечити виконання наступних функційних вимог:

- створювати, оновлювати, видаляти та відображати дані про: студентів навчальної установи, викладачів, оцінки, предмети, які вивчаються, групи, а також екзамени;
- забезпечити фільтрацію даних по атрибутах для студентів навчальної установи, викладачів, оцінок, предметів, груп та екзаменів;
- забезпечити безпеку системи за допомогою авторизації користувачів та обмеженням доступу до функціоналу залежно від їх ролей (адміністратор, викладач, студент);
- забезпечення формування та побудову звітів, наприклад про розклад екзаменів, наявні групи та студенти, викладачів, тощо.

У якості нефункційних вимог виділимо такі вимоги як зручність відображення та інтуїтивний дизайн, а також легкість масштабування інформаційної системи.

2.3 Проектування концептуальної моделі бази даних

Проектування концептуальної моделі бази даних є важливим етапом у розробці будь-якої інформаційної системи. Цей етап передбачає створення абстрактної моделі даних, яка відображає структуру і зв'язки між різними

						КвРІСТ. 200182.20.01.06 ПЗ	Арк. 30
Зм.	Арк.	№докум.	Підпис	Дата			

зв'язки між ними за допомогою зовнішніх ключів, а також SQL запитам створення таблиць.

Таблиця 2.1 – Атрибути та їх тип для сутності Студенти

Назва атрибуту	Тип атрибуту	Опис атрибуту
StudentID	INT	Ідентифікатор студента
FirstName	VARCHAR(50)	Ім'я студента
LastName	VARCHAR(50)	Прізвище студента
PassNum	INT	Номер паспорта
GroupID	INT	Ідентифікатор групи

У сутності Students поле StudentID є первинним ключем, яке однозначно ідентифікує кожного студента. Це поле не може бути порожнім (NOT NULL). Поля FirstName та LastName відповідають ім'ям та прізвищам студентів і обмежені до 50 символів кожне. Вони також не можуть бути порожніми. Поле PassNum містить номер паспорта студента і також не може бути порожнім. Поле "GroupID" вказує на ідентифікатор групи, до якої належить студент. Це зовнішній ключ, який посилається на поле GroupID таблиці Groups. При оновленні або видаленні групи, пов'язані записи у таблиці "Students" також оновлюються або видаляються. Це важливо для збереження консистентності даних в базі даних. Коли наприклад необхідно оновити або видалити групу із таблиці Groups, то це повинно мати вплив на студентів, які є членами цієї групи. Таким чином відповідний SQL запит для створення таблиці Students буде виглядати наступним чином:

```
DROP TABLE IF EXISTS Students;  
CREATE TABLE Students(  
    StudentID INT NOT NULL PRIMARY KEY,  
    FirstName VARCHAR(50) NOT NULL,  
    LastName VARCHAR(50) NOT NULL,  
    PassNum INT NOT NULL,
```

```

GroupID INT NOT NULL,
FOREIGN KEY (GroupID) REFERENCES Groups(GroupID) ON UPDATE
CASCADE ON DELETE CASCADE
)

```

Сутність Students з'єднання відношенням багато-до-одного із сутністю Groups. Це означає, що одна група може мати багато студентів, але кожен студент належить лише до однієї групи. Дана сутність представлена двома атрибутами – GroupID та GroupName, що визначають ідентифікатор групи та назву групи відповідно.

Таблиця 2.2 – Атрибути та їх тип для сутності Групи

Назва атрибуту	Тип атрибуту	Опис атрибуту
GroupID	INT	Ідентифікатор групи
GroupName	VARCHAR(50)	Ім'я групи

Відповідний SQL запит для створення таблиці Групи буде виглядати наступним чином:

```

DROP TABLE IF EXISTS Groups;
CREATE TABLE Groups(
    GroupID INT NOT NULL PRIMARY KEY,
    GroupName VARCHAR(50) NOT NULL
)

```

Ще однією сутністю, яка зберігає кінцевих користувачів (окрім студентів) є сутність Викладачі. Дана сутність спроектована із чотирма атрибутами: TeacherID, FirstName, LastName, а також Position. TeacherID виступає як первинний ключ для унікальної ідентифікації кожного викладача. FirstName та LastName використовуються для зберігання особистої інформації про викладача, тоді як атрибут Position визначає посаду викладача. Всі ці поля мають обмеження NOT NULL, що вказує на обов'язковість вводу даних у ці поля.

Відповідний SQL запит для створення таблиці Предмети буде виглядати наступним чином:

```
DROP TABLE IF EXISTS Subjects;  
CREATE TABLE Subjects (  
    SubjectID INT NOT NULL PRIMARY KEY,  
    SubjectName VARCHAR(50),  
    AmountOfHours INT NOT NULL  
)
```

Центральною сутністю у проєктованій концептуальній базі даних є сутність Екзамен, яка містить інформацію про проведення екзаменів. У реляційній базі даних дана сутність представлена відповідною таблицею, яка складається із таких атрибутів: ExamID, GroupID, SubjectID, TeacherID, RoomNumber, DateOfExam, Lesson. Кожен екзамен має унікальний ідентифікатор ExamID. GroupID вказує на групу студентів, які беруть участь у цьому екзамені, і може бути порожнім, якщо екзамен не призначений для конкретної групи. SubjectID вказує на предмет, який буде перевіряти на екзамені. TeacherID вказує на викладача, який проводить екзамен. Даний атрибут посилається на первинний ключ TeacherID у таблиці Teachers. Це вказує на те, що кожний екзамен має відповідного викладача, який проводить цей екзамен. Використання зовнішнього ключа дозволяє забезпечити цілісність даних і забезпечити, що відповідний викладач існує в таблиці Teachers перед тим, як він буде призначений для екзамену. RoomNumber вказує на номер аудиторії, де проводиться екзамен. DateOfExam вказує на дату проведення екзамену. Lesson вказує на номер уроку або пари, коли проводиться екзамен.

Відповідний SQL запит для створення таблиці Предмети буде виглядати наступним чином:

```
DROP TABLE IF EXISTS Exam;  
CREATE TABLE Exam (  
    ExamID INT NOT NULL PRIMARY KEY,
```

Зм.	Арк.	№докум.	Підпис	Дата

GroupID INT,
 SubjectID INT NOT NULL,
 TeacherID INT NOT NULL,
 RoomNumber INT NOT NULL,
 DateOfExam DATE,
 Lesson INT NOT NULL,
 FOREIGN KEY (GroupID) REFERENCES Groups(GroupID),
 FOREIGN KEY (SubjectID) REFERENCES Subjects(SubjectID),
 FOREIGN KEY (TeacherID) REFERENCES Teachers(TeacherID)

)

Таблиця 2.5 – Атрибути та їх тип для сутності Екзамен

Назва атрибуту	Тип атрибуту	Опис атрибуту
ExamID	INT	Ідентифікатор екзамену
GroupID	INT	Ідентифікатор групи (зовнішній ключ)
SubjectID	INT	Ідентифікатор предмету (зовнішній ключ)
TeacherID	INT	Ідентифікатор викладача (зовнішній ключ)
RoomNumber	INT	Номер аудиторії
DateOfExam	DATE	Дата проведення екзамену
Lesson	INT	Номер уроку

Ще однією сутністю у даній базі даних є Оцінки. Дана сутність пов'язана із сутностями Студент та Предмет. У цій сутності виділені наступні атрибути: GradeID, StudentID, SubjectID та Mark. У концептуальній моделі бази даних ця сутність представлена у вигляді таблиці Grade, що вміщує дані про оцінки, які отримують студенти за різні предмети. Кожен запис у цій таблиці має унікальний ідентифікатор GradeID. Поле StudentID вказує на студента, який отримав цю оцінку, та містить зовнішній ключ, що посилається на ідентифікатор студента. Поле SubjectID вказує на предмет, за який була отримана ця оцінка, та також містить зовнішній ключ, що посилається на ідентифікатор предмета. Поле Mark

представляє собою числове значення оцінки, яку отримав студент за даний предмет.

Відповідний SQL запит для створення таблиці Предмети буде виглядати наступним чином:

```
DROP TABLE IF EXISTS Grade;
CREATE TABLE Grade (
    GradeID INT NOT NULL PRIMARY KEY,
    StudentID INT NOT NULL,
    SubjectID INT NOT NULL,
    Mark INT NOT NULL,
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
    FOREIGN KEY (SubjectID) REFERENCES Subjects(SubjectID)
)
```

Таблиця 2.6 – Атрибути та їх тип для сутності Екзамен

Назва атрибуту	Тип атрибуту	Опис атрибуту
GradeID	INT	Ідентифікатор оцінки
StudentID	INT	Ідентифікатор студента (зовнішній ключ)
SubjectID	INT	Ідентифікатор предмету (зовнішній ключ)
Mark	INT	Оцінка студента із предмету

Таким чином, концептуальна модель бази даних складається із зазначених вище таблиць, які взаємодіють між собою за допомогою наступних зв'язків (відношень):

– Groups - Students: Це зв'язок один до багатьох, оскільки одна група може містити багато студентів, але кожен студент може належати лише до однієї групи. В таблиці Students поле GroupID є зовнішнім ключем, який посилається на ідентифікатор групи в таблиці Groups.

– Teachers - Exam: Це також зв'язок один до багатьох, оскільки один викладач може проводити багато екзаменів, але кожен екзамен має лише одного викладача. У таблиці Exam поле TeacherID є зовнішнім ключем, який посилається на ідентифікатор викладача в таблиці Teachers.

– Subjects - Exam: Це також зв'язок один до багатьох, оскільки один предмет може мати багато екзаменів, але кожен екзамен відбувається лише з одним предметом. У таблиці Exam поле SubjectID є зовнішнім ключем, який посилається на ідентифікатор предмета в таблиці Subjects.

– Students - Grade: Це зв'язок один до багатьох, оскільки кожен студент може мати багато оцінок, але кожна оцінка належить лише одному студентові. У таблиці Grade поле StudentID є зовнішнім ключем, який посилається на ідентифікатор студента в таблиці Students.

– Groups - Exam: Це зв'язок багато до одного, оскільки кожна група може мати багато екзаменів, але кожен екзамен відноситься лише до однієї групи. У таблиці Exam поле GroupID є зовнішнім ключем, який посилається на ідентифікатор групи в таблиці Groups.

– Teachers - Grade: Це також зв'язок багато до одного, оскільки кожен викладач може мати багато оцінок, але кожна оцінка призначена лише одному викладачу. У таблиці Grade поле TeacherID є зовнішнім ключем, який посилається на ідентифікатор викладача, що представлений в таблиці Teachers.

Концептуальна модель бази даних та ERD діаграму для проєктованої інформаційної системи управління навчальним процесом освітньої установи, що представлена у формі сутностей та зв'язків між ними наведено на рисунку 2.1 та 2.2.

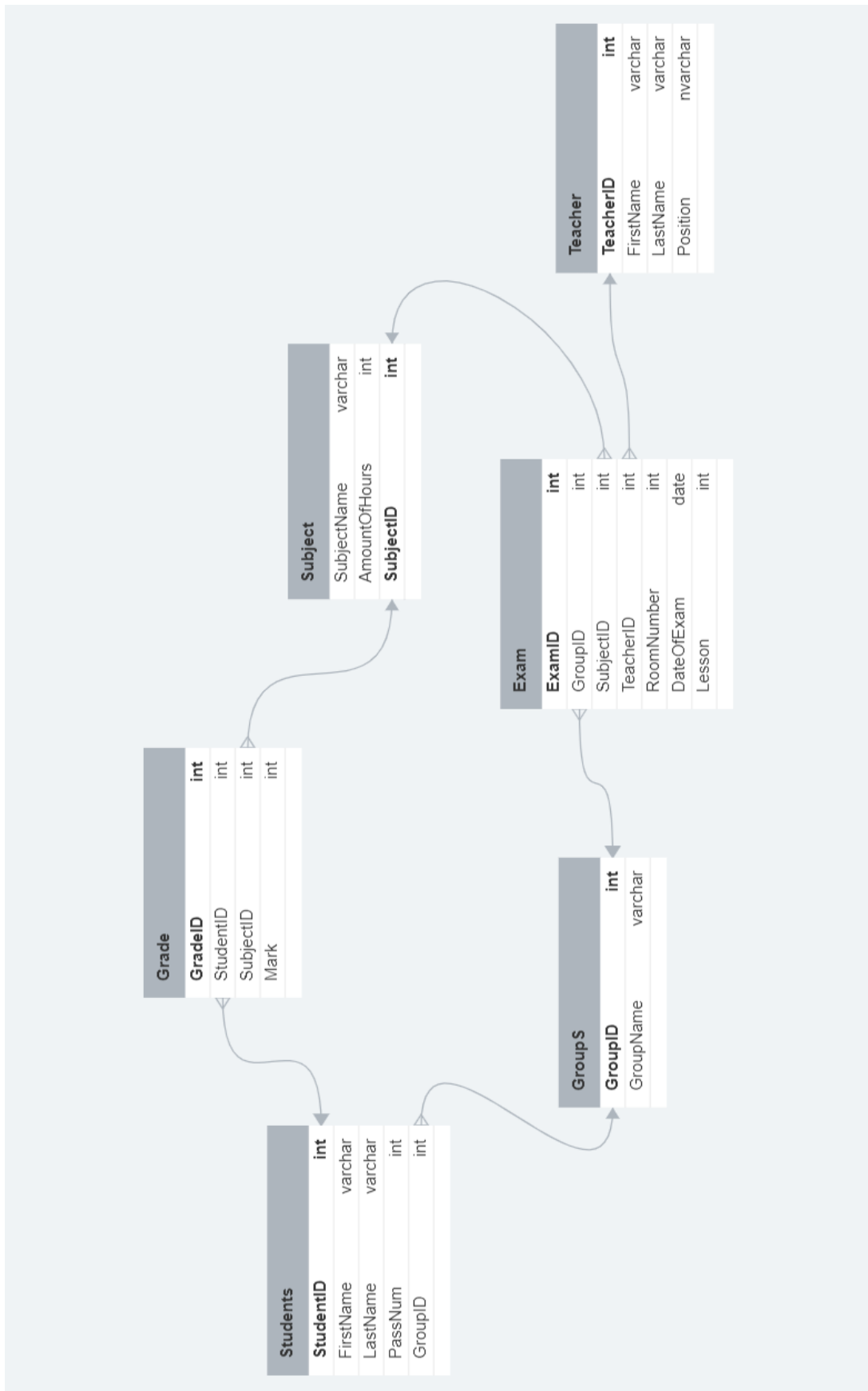


Рисунок 2.2 – Концептуальна модель бази даних для інформаційної системи управління навчальним процесом освітньої установи

На рисунку 2.3 подано ERD діаграму спроектованої бази даних для інформаційної системи управління навчальним процесом освітньої установи. Дана схема відображає взаємозв'язки та взаємодії між студентами, оцінками, іспитами, вчителями та групами. В її складі присутні такі об'єкти:

1. Сутності:

- Студент: Позначений як “Student”. Має атрибути, такі як “Прізвище”, “Ім'я” та “Номер залікової книжки”;
- Оцінка: Пов'язана зі студентом. Має атрибут “Оцінка”;
- Предмет: Пов'язаний з оцінкою. Має атрибут “Назва предмету”;
- Іспит: Пов'язаний з предметом. Має атрибут “Дата іспиту”;
- Вчитель: Пов'язаний з предметом. Має атрибути “Прізвище” та “Ім'я”;
- Група: Пов'язана зі студентом. Має атрибут “Назва групи”;

2. Взаємозв'язки:

- студенти належать до груп;
- студенти отримують оцінки за предмети;
- оцінки призначаються вчителями;
- предмети проводяться на іспитах;

3. Кардинальність:

- взаємозв'язок між студентами та групами: багато до багатьох (багато студентів може належати до багатьох груп, і навпаки);
- взаємозв'язок між студентами та оцінками: багато до багатьох (студент може мати багато оцінок, і навпаки);
- взаємозв'язок між оцінками та предметами: багато до одного (одна оцінка призначається одному предмету);
- взаємозв'язок між предметами та іспитами: один до багатьох (один предмет може мати багато іспитів);

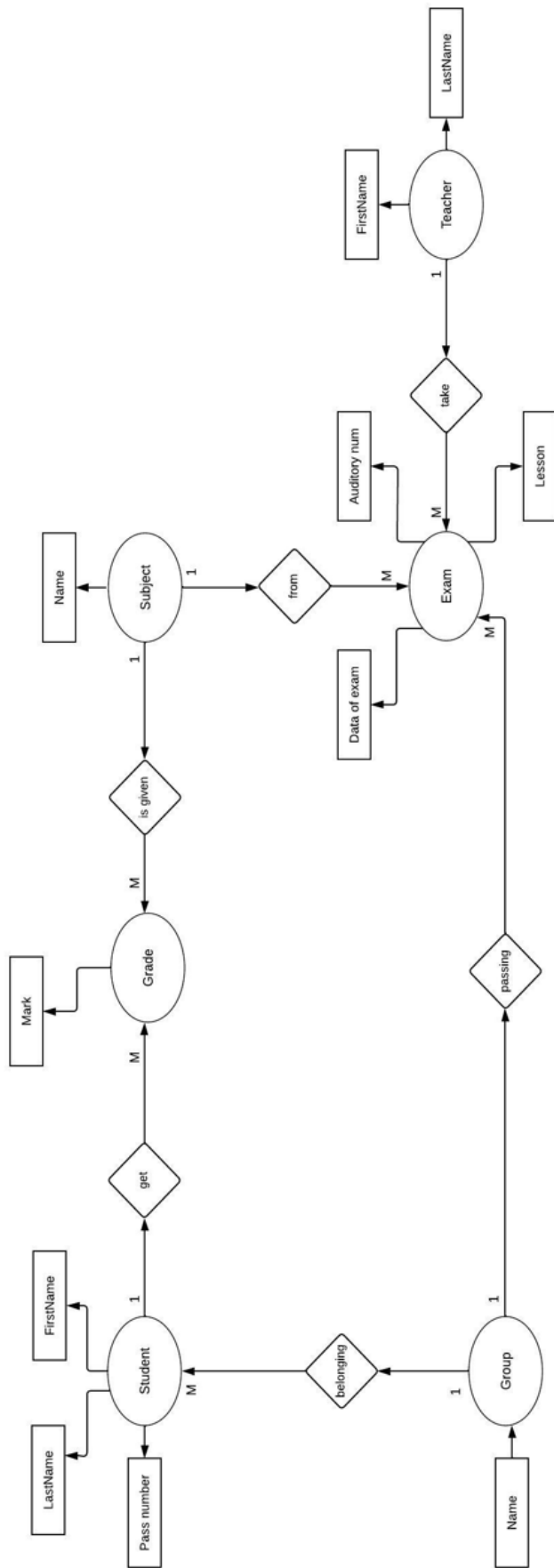


Рисунок 2.3 – ERD діаграма спроектованої бази даних для інформаційної системи управління навчальним процесом освітньої установи

Таким чином в результаті створення концептуальної моделі бази даних були визначені основні сутності, атрибути та зв'язки між ними для відображення предметної області управління навчальним процесом. Було ідентифіковано такі сутності, як групи, викладачі, студенти, предмети, екзамени та оцінки. У кожній сутності визначено характеристики, які відображають специфіку предметної області. Також було визначено зв'язки між цими сутностями, що дозволило зберігати та відслідковувати дані про навчальний процес, включаючи проведення екзаменів та оцінки студентів. Створення концептуальної моделі бази даних дало змогу краще розуміти структуру та взаємозв'язки в предметній області, а також підготувати основу для подальшого розроблення та реалізації бази даних. Це дозволить ефективно зберігати, оновлювати та отримувати інформацію про навчальний процес у навчальній установі.

Висновки

Таким чином, було проведено аналіз технологій та засобів для розробки інформаційної системи управління навчальним процесом освітньої установи, розглянуто їх переваги та недоліки. Визначено функційні та нефункційні вимоги до інформаційної системи управління навчальним процесом освітньої установи. Проведено проектування концептуальної бази даних, визначено сутності та їх атрибути. Створення концептуальної моделі бази даних дало змогу описати структуру та взаємозв'язки в предметній області, а також підготувати основу для подальшого розроблення та інформаційної системи.

					КВРІСТ. 200182.20.01.06 ПЗ	Арк.
						42
Зм.	Арк.	№докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ ОСВІТНЬОЇ УСТАНОВИ

3.1 Архітектура інформаційної системи управління навчальним процесом освітньої установи

З метою організації обліку іспитів, груп, студентів, предметів та викладачів для освітньої установи розроблено відповідну інформаційну систему, яка реалізована у формі програмного забезпечення.

Узагальнену структуру інформаційної системи управління навчальним процесом освітньої установи подано на рисунок 3.1 Вхідними даними для інформаційної системи є дані, що вводяться студентами та/або викладачами. Вихідними даними інформаційної системи є інформація про студентів, групи, оцінки, викладачів, предмети та іспити. Вона включає наступні компоненти:

– клієнтська частина: Це веб-сторінки, які користувач бачить у своєму браузері. Вони відповідають за відображення та інтерактивність з користувачем. Клієнтська частина збирає та відображає інформацію про навчання, розклад занять, результати тощо. Клієнтська частина реалізована за допомогою бібліотеки React js;

– серверна частина (backend): Це програмне забезпечення, яке працює на сервері. Це програмне забезпечення відповідає за збереження, обробку та передачу даних між сервером та клієнтом. Із серверною частиною взаємодіє база даних, яка зберігає інформацію про студентів, групи, оцінки, викладачів, предмети та екзамени. Взаємодія із базою даних реалізована за допомогою провайдера даних SqlConnection (простір імен System.Data.SqlClient). Цей простір імен містить класи та інші засоби для створення з'єднань із SQL Server, виконання SQL-запитів, обробки результатів запитів та керування базою даних. Окрім провайдера даних, у складі серверної частини реалізовано відповідні моделі та API контролери;

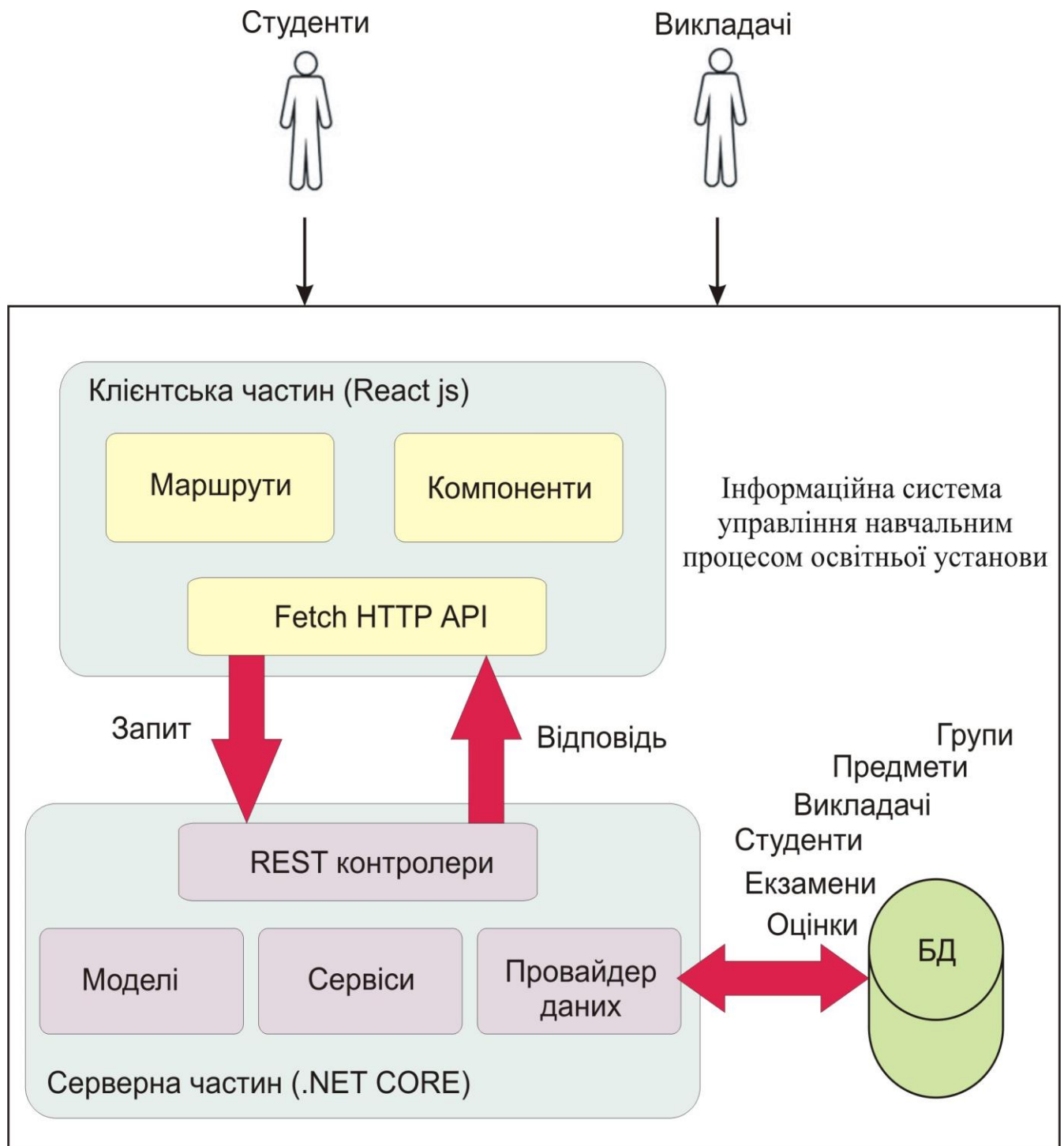


Рисунок 3.1 – Узагальнена структура інформаційної системи управління навчальним процесом освітньої установи

Доступ до функцій сервера реалізується через API (Application Programming Interface), яке надає інтерфейс для взаємодії між клієнтським додатком і сервером. Це означає, що клієнтський додаток взаємодіє з сервером, надсилаючи HTTP-запити на виконання певних операцій або отримання даних. Наприклад, коли

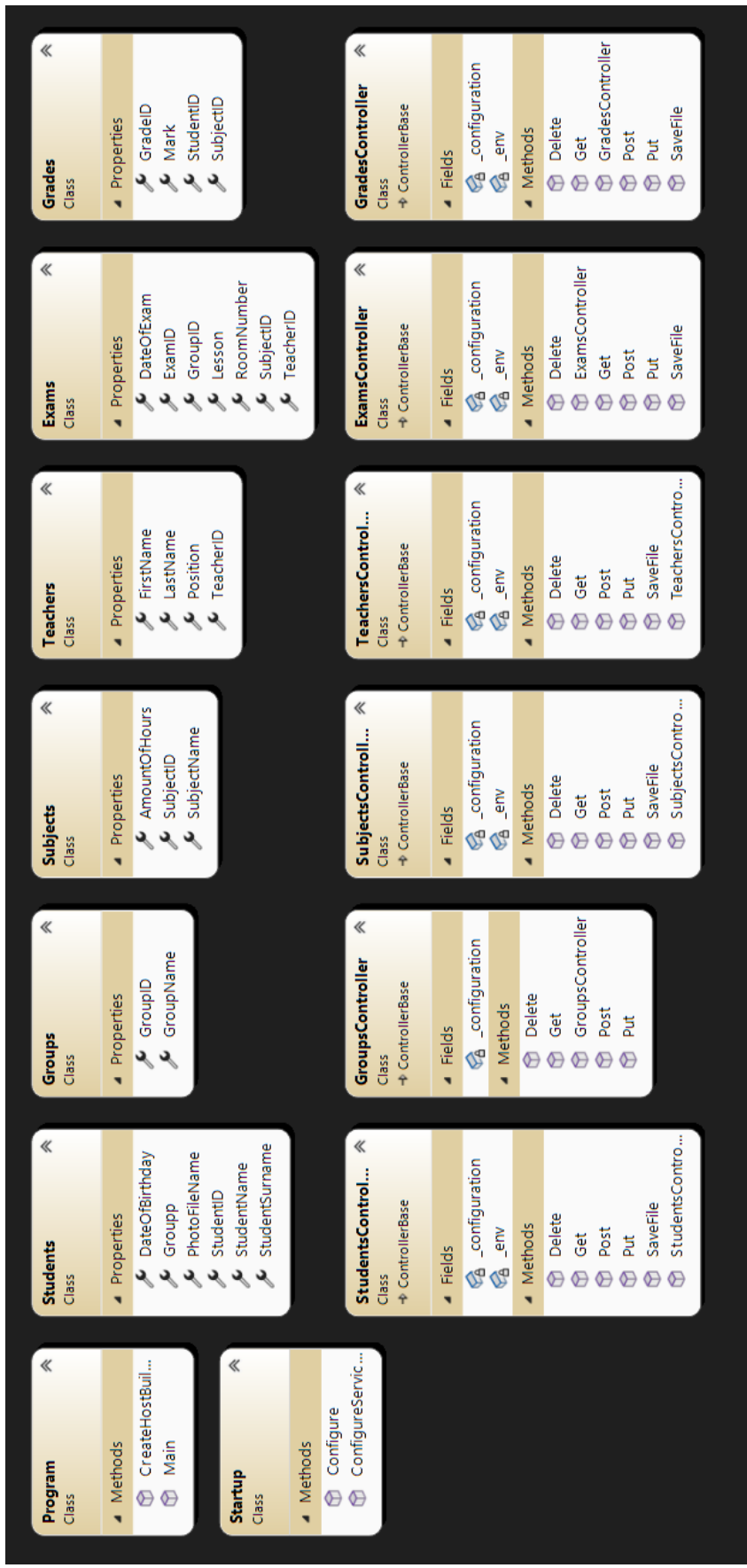


Рисунок 3.2 – Діаграма Class View

Далі у файлі appsettings.json було додано рядок підключення до бази даних, що використовує існуючу інфраструктуру безпеки Windows для автентифікації користувача:

```
"ConnectionStrings": {  
    "Con": "Server=localhost;Database=studMark;Trusted_Connection=True;"  
}
```

Далі було додано моделі даних, що представляли структуру даних, які використовуються у додатку. Було додано моделі Students, Groups, Teachers, Subjects, Grades та Exams.

Після цього було створено набір API контролерів, зокрема StudentsController, GroupsController, TeachersController, SubjectsController, GradeControllers та ExamsController.

Розглянемо до прикладу контролер StudentsController. Даний контролер відповідає за обробку запитів, пов'язаних з операціями CRUD (створення, читання, оновлення та видалення) над студентами в системі. Реалізовано набір методів для опрацювання запитів від клієнта: GET, POST, PUT, DELETE.

У методі Get() контролера виконується запит до бази даних для отримання списку студентів. За допомогою сервісу IConfiguration зчитуються дані, після чого виконується SQL-запит і дані повертаються у форматі JSON. Реалізація методу Get() наведено у наступному коді:

```
public JsonResult Get()  
{  
    string query = @"  
        select StudentID,StudentName,StudentSurname,Group,  
        convert(varchar(10),DateOfBirthday,120) as  
DateOfBirthday,PhotoFileName  
        from  
        dbo.Students ";  
    DataTable table = new DataTable();  
    string sqlDataSource = _configuration.GetConnectionString("Con");
```

```

SqlDataReader myReader;
using (SqlConnection myCon = new SqlConnection(sqlDataSource))
{
    myCon.Open();
    using (SqlCommand myCommand = new SqlCommand(query,
myCon))
    {
        myReader = myCommand.ExecuteReader();
        table.Load(myReader);
        myReader.Close();
        myCon.Close();
    }
}
return new JsonResult(table);
}

```

Метод Post() використовується для створення нового студента. Дані про студента передаються у тілі HTTP-запиту, після чого вони вставляються до бази даних за допомогою SQL-запиту.

Метод Put() використовується для оновлення інформації про студента. Також дані про студента передаються у тілі HTTP-запиту, після чого за допомогою SQL-запиту відбувається оновлення запису в базі даних.

Метод Delete() призначений для видалення студента. ID студента передається як параметр URL-адреси, після чого виконується відповідний SQL-запит для видалення запису з бази даних.

Метод SaveFile() використовується для завантаження фотографії студента на сервер. Він приймає файл, який передається через HTTP-запит, зберігає його на сервері у папці “Photos” та повертає ім’я файлу.

В процесі реалізації API сервера було перевірено реалізовані запити за допомогою програмного забезпечення Postman. На рисунку 3.3 зображено

перевірку GET запиту до сервера. Як можна побачити із рисунка, запит повернув json дані про всіх студентів у БД.

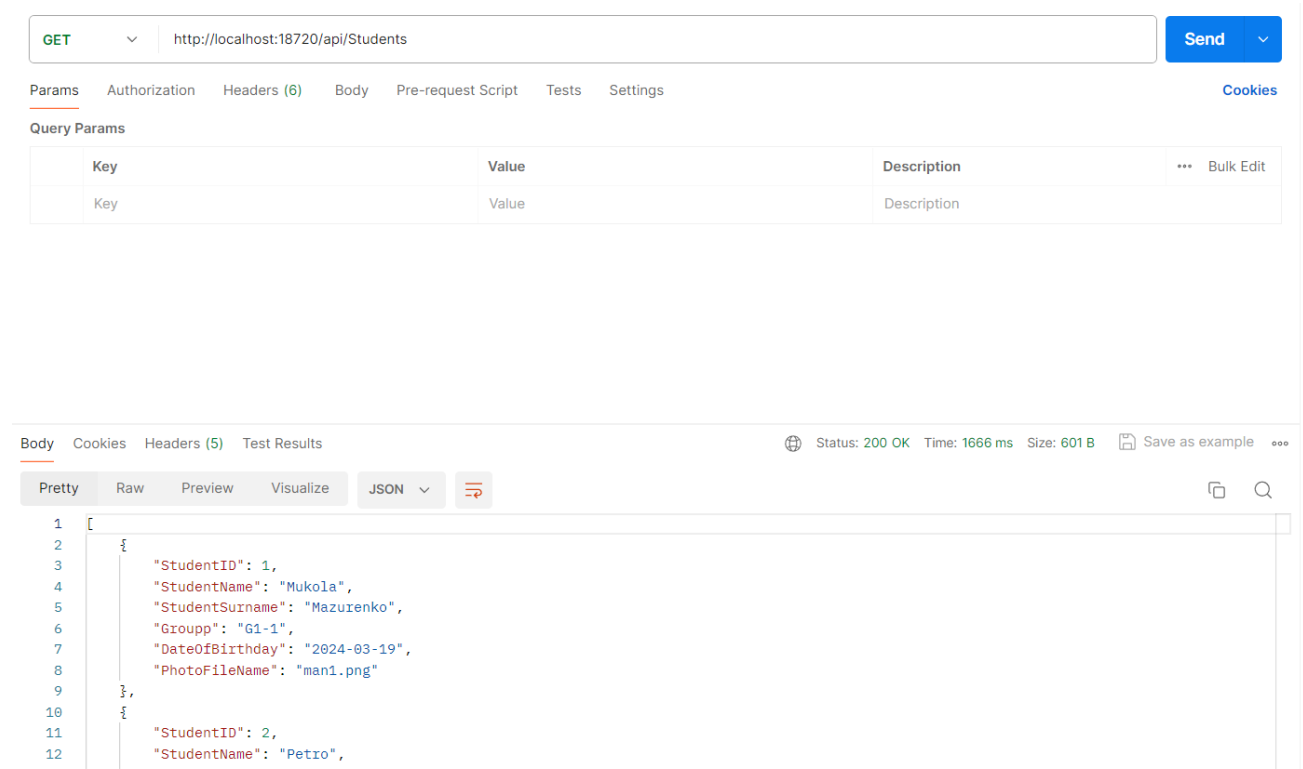


Рисунок 3.3 – Перевірка GET запиту до API сервера

3.3 Реалізація клієнтської частини

Реалізація клієнтської частини програмного забезпечення інформаційної системи управління навчальним процесом освітньої установи передбачало розробку веб сторінки, через яку здійснюється взаємодія користувача із ІС. При визначенні вимог було встановлено, що основною вимогою, яка ставиться до ІС є забезпечення зручного та ефективного відображення графіку іспитів та доступних аудиторій для викладачів і студентів. Система повинна надавати зрозумілу та легкодоступну інформацію щодо дат та часу проведення іспитів, а також наявності вільних аудиторій для їх проведення.

Реалізація клієнтської частини була здійснена за допомогою бібліотеки React js.

При розробці клієнтської частини було створено інтерфейсні вікна для відображення інформації про студентів, групи, викладачів, предмети, оцінки, а також екзамени. При реалізації даних інтерфейсних вікон було реалізовано наступний функціонал:

- створення, оновлення, видалення та відображення даних про: студентів навчальної установи, викладачів, оцінки, предмети, які вивчаються, групи, а також екзамени;
- забезпечення фільтрації даних по атрибутах для студентів навчальної установи, викладачів, оцінок, предметів, груп та екзаменів;
- забезпечення формування та побудову звітів, наприклад про розклад екзаменів, наявні групи та студенти, викладачів, тощо.

На рисунку 3.4 зображено інтерфейсне вікно сторінки Group.

Система управління навчальним процесом освітньої установи

Home Group Student Exam Subject Teachers Grade Add Group

Filter Filter

GroupID	Group Name	Options
1	G1-1	☑ ☒
2	G1-2	☑ ☒
3	G2-1	☑ ☒
7	G4-111	☑ ☒
8	P1-1	☑ ☒

Рисунок 3.4 – Інтерфейсне вікно сторінки Group

При реалізації інтерфейсу основний фокус було спрямовано на зручний та інтуїтивно зрозумілий інтерфейс. Дані відображаються у вигляді таблиці, яка складається із заголовку та рядків. В останньому стовпчику розташовані опції (у вигляді піктограм) для редагування та видалення відповідних даних у заданому стовпчику.

Додано можливість фільтрації даних, зокрема у таблиці Group фільтрувати дані можна як по ідентифікатору групи, так і по назві групи (рисунок 3.5, 3.6). Є

можливість задати порядок сортування – за зростанням чи спаданням (теж представлено у вигляді відповідних піктограм).

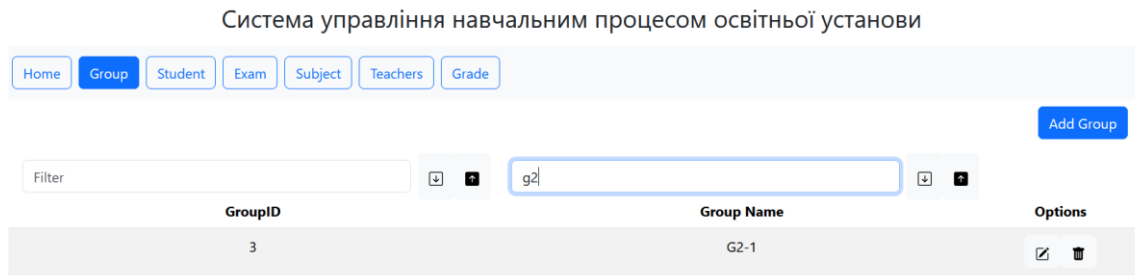


Рисунок 3.5 – Інтерфейсне вікно сторінки Group. Фільтрація груп по імені

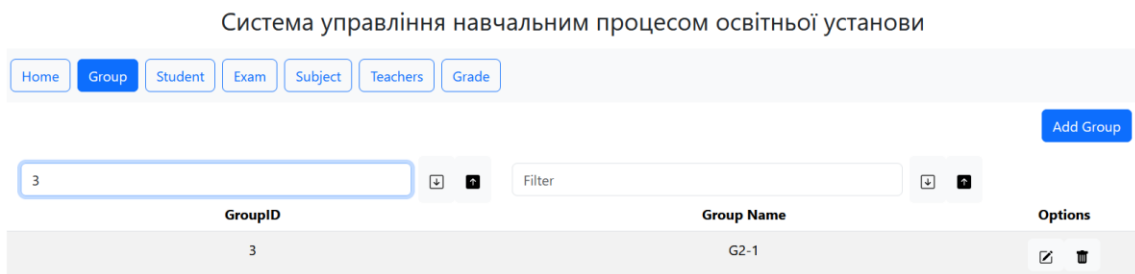


Рисунок 3.6 – Інтерфейсне вікно сторінки Group. Фільтрація груп по ідентифікатору

Також є можливість додавання нової групи. Даний функціонал реалізовано у вигляді модального вікна. Передбачена можливість перевірки, якщо зазначена група вже існує.

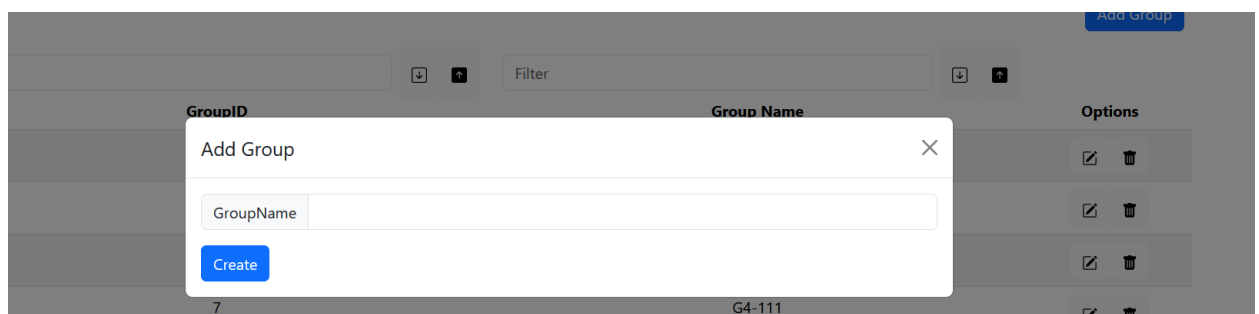


Рисунок 3.7 – Інтерфейсне вікно сторінки Group. Фільтрація даних

Також реалізовано опцію додавання нових студентів. Параметри, які визначають інформацію про студента, збігаються із інформацією, яку можна змінити у вікні оновлення даних. Інтерфейсне вікно додавання даних про студента подано на рисунок 3.10.

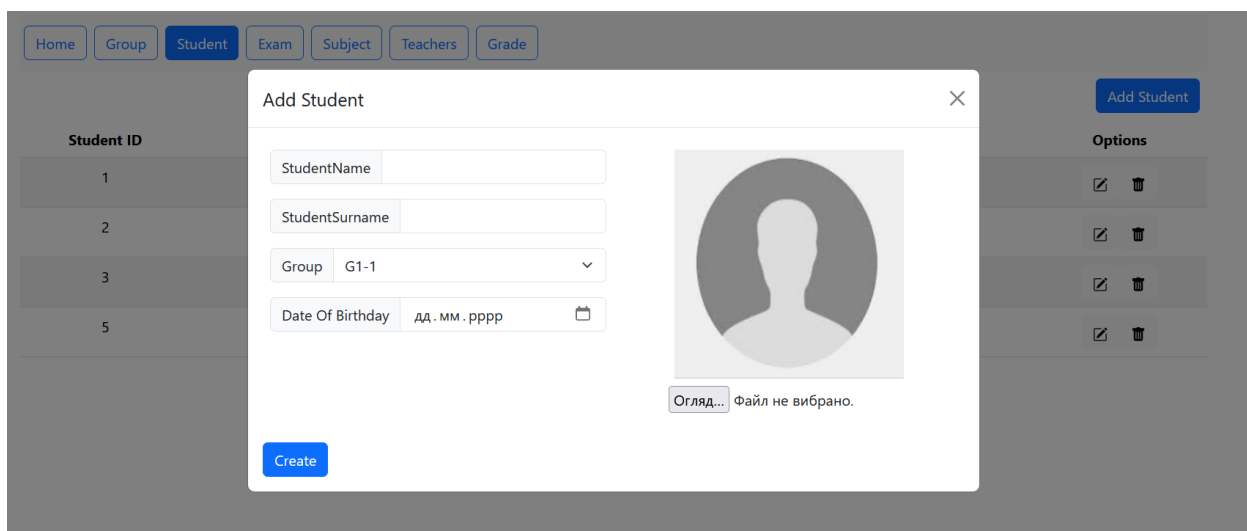


Рисунок 3.10 – Інтерфейсне вікно додавання даних про студента

Ще однією функцією, яка забезпечує програмне забезпечення інформаційної системи є видалення даних про студентів (рисунок 3.11). Для реалізації видалення слід натиснути на відповідну піктограму навпроти студента. Реалізовано модальне вікно, у якому інформується про можливість видалення (для запобігання помилковому видаленні інформації).

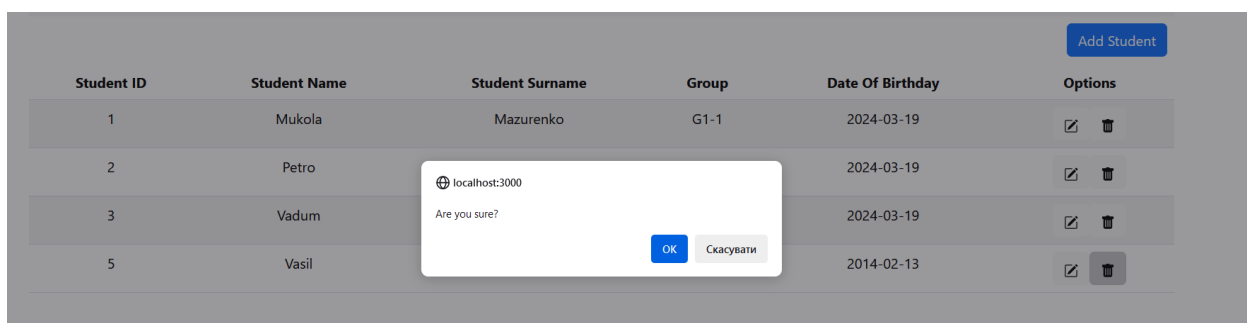


Рисунок 3.11 – Інтерфейсне вікно видалення даних про студента

З метою забезпечення конфіденційності, цілісності та доступності даних, а також захисту системи від несанкціонованого доступу та зловживань реалізовано обмеження функціоналу системи через авторизацію користувачів. Коли користувач намагається увійти до системи, він повинен ввести свої ідентифікаційні дані, такі як ім'я користувача та пароль. Інформаційна система перевіряє ці дані, щоб переконатися, що вони вірні, та авторизує користувача для входу у систему.

Після авторизації користувача система визначає його роль. У даній ІС визначено дві ролі – студент та викладач. Кожна роль має свої визначені права доступу до функціоналу системи. Викладач має повний доступ до всіх функцій системи, в той час як студент має обмежений доступ лише до своїх особистих даних та оцінок.

Таким чином в результаті було спроектовано та реалізовано у вигляді програмного забезпечення інформаційну систему управління навчальним процесом освітньої установи. Реалізована інформаційна система має наступні функції: відображення даних у вигляді звітів про студентів, викладачів, оцінки, предмети, а також екзамени. Передбачено можливість додавання, видалення та оновлення інформації про відповідні сутності, а також реалізовано фільтрацію даних за різними полями. Додано функцію авторизації та визначено дві ролі – студент та викладач. Окрім того варто відзначити досить легкий та зручний інтерфейс, що дозволяє легко зорієнтуватись у функціях інформаційної системи.

3.4 UML діаграм для візуалізація роботи інформаційної системи

Для візуалізації роботи розробленої інформаційної системи управління навчальним процесом освітньої установи використовуємо UML (Unified Modeling Language), що є стандартною мовою моделювання, широко використовуваною розробниками програмного забезпечення для візуалізації процесів та роботи інформаційних систем. Використання UML-діаграм для візуалізації структури розробленої інформаційної системи обумовлене потребою у зрозумілому та

зручному способі відобразити її складові частини та взаємозв'язки між ними. Графічне представлення дозволяє краще розуміти роботу інформаційної системи, виявляти можливі недоліки та оптимізувати процес розробки та управління ним. Існує велика кількість UML діаграм, проте для опису даної інформаційної системи скористаємось такими діаграмами як послідовності та варіантів використання.

Діаграма варіантів використання (use case diagram) – це графічне зображення функціональної поведінки системи, яка моделює, як користувачі взаємодіють з системою для досягнення певних цілей. Основні елементи діаграми варіантів використання включають акторів та варіанти використання:

Актори: Актори представляють ролі, які взаємодіють з системою. Це можуть бути реальні користувачі, інші системи або зовнішні процеси. Актори зазвичай зображуються у вигляді піктограм та розташовуються поза межами системи.

Варіанти використання: Варіанти використання (use cases) описують конкретні дії або функціональність, яку система надає своїм акторам. Вони зображуються у вигляді овалів або прямокутників всередині системи та зв'язані з акторами лініями.

Діаграма варіантів використання допомагає уточнити вимоги до системи та зрозуміти, як користувачі будуть взаємодіяти із цією системою.

Для проєктованої інформаційної системи виділено два актори – студент та викладач. Для викладача передбачено можливість виконання таких операцій по створенню, редагуванню, видалення та перегляду інформації для сутностей викладачі, студенти, оцінки, предмети, групи, а також екзамени. Тоді як для студента визначено можливість роботи із сутностями студент, предмети та оцінки.

UML-діаграму варіантів використання розробленої інформаційної системи управління навчальним процесом освітньої установи представлено на рисунку 3.12.

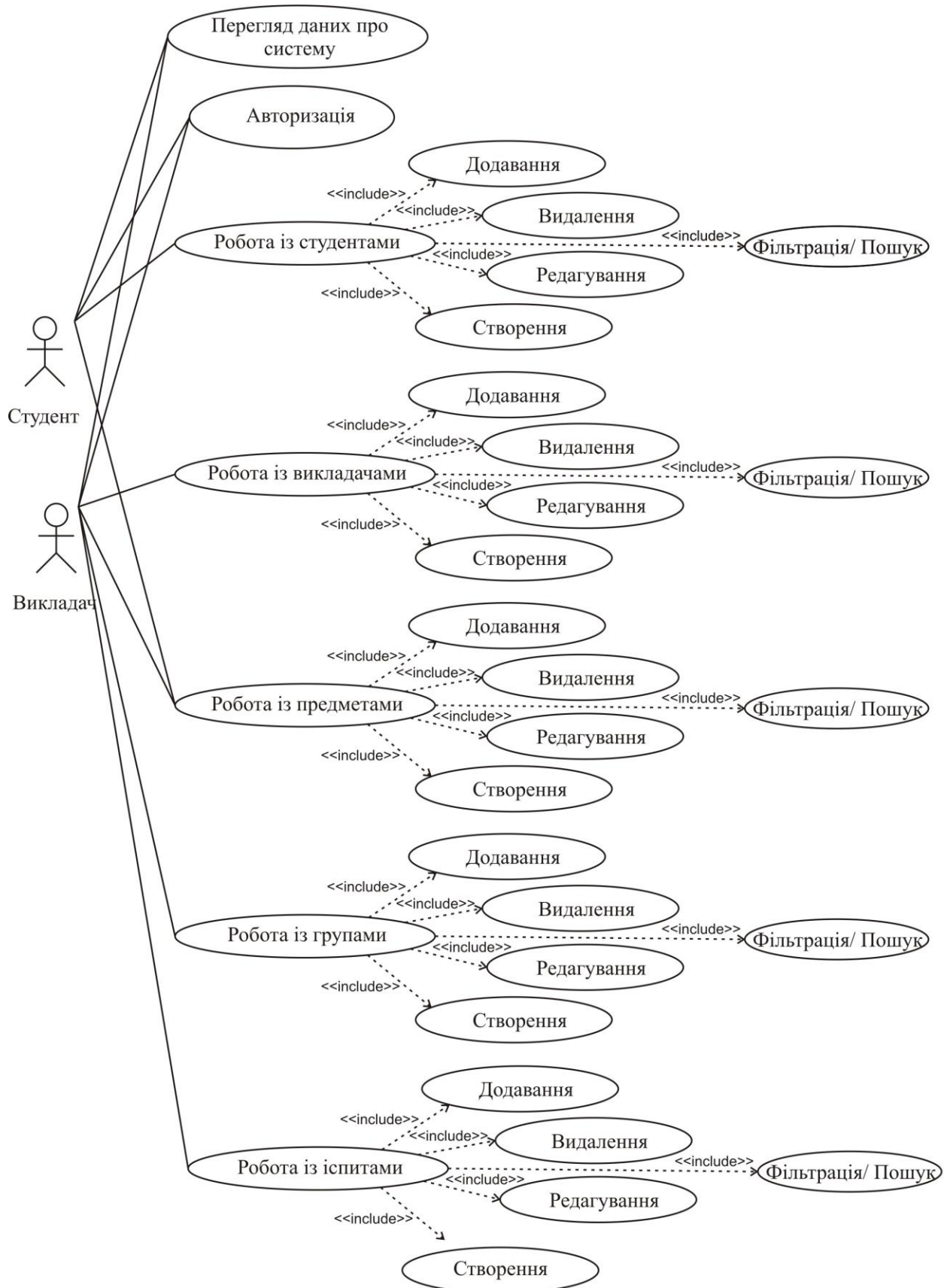


Рисунок 3.12 – UML-діаграма варіантів використання

Для представлення послідовності роботи із розробленою інформаційною системою використаємо UML діаграму послідовності. На рисунку 3.13 представлено послідовність дій із системою для викладача.

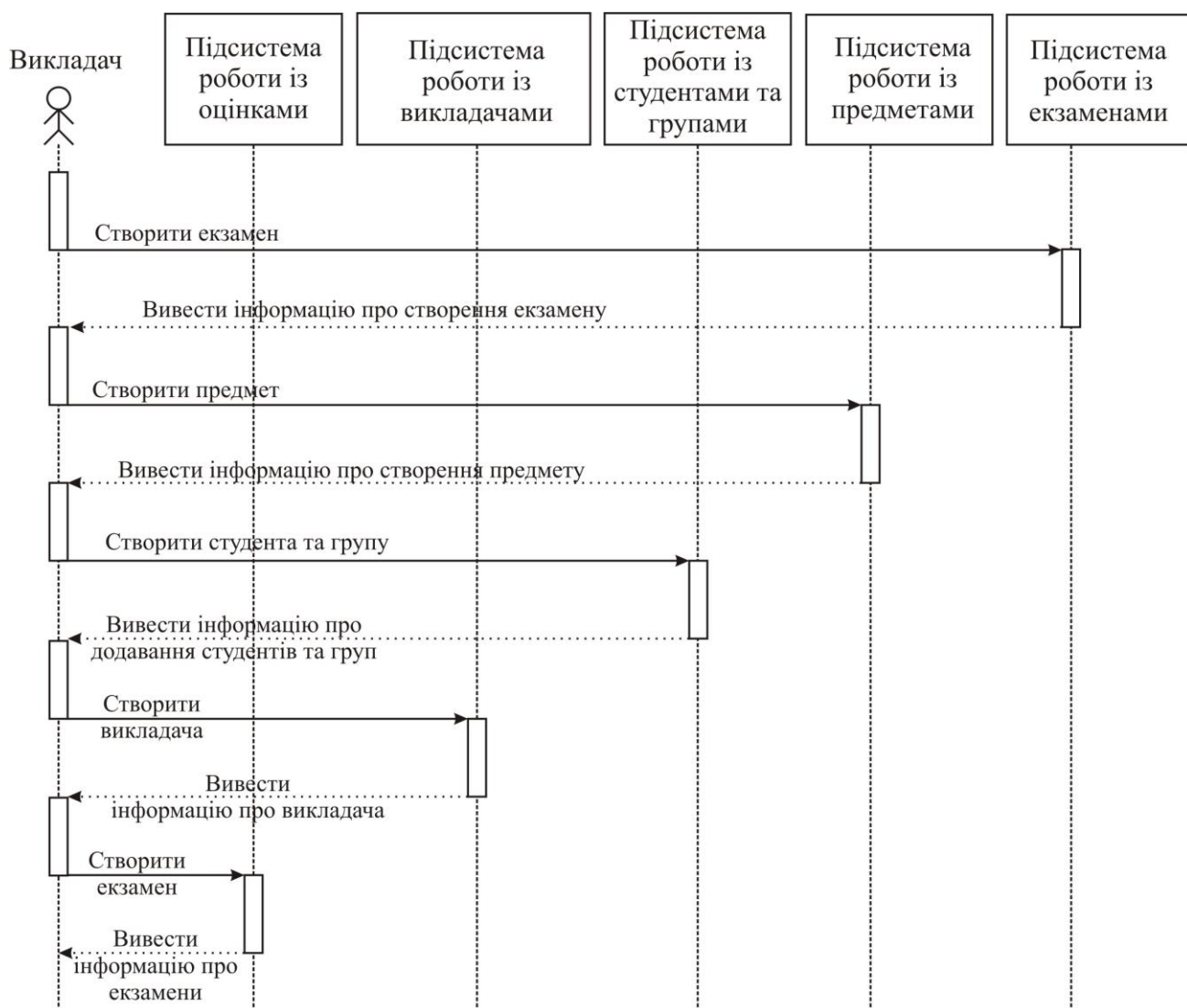


Рисунок 3.13 – UML-діаграма послідовності виконання

На початку роботи викладач створює екзамен, де відобразатимуться дані про дату, час та місце проведення. Далі здійснюється додавання предмету. Після цього здійснюється створення груп та студентів. Після цього здійснюється створення оцінок. На кожному кроці є можливість змінити та видалити вже додані записи, а також переглянути вже створені записи.

Для представлення структури запропонованої інформаційної системи використовується UML-діаграма компонентів. Згідно з цією діаграмою, інформаційна система складається з наступних компонентів: пристрій користувача, веб-сервер, сервер бази даних. UML-діаграму компонентів наведено на рисунку 3.14.

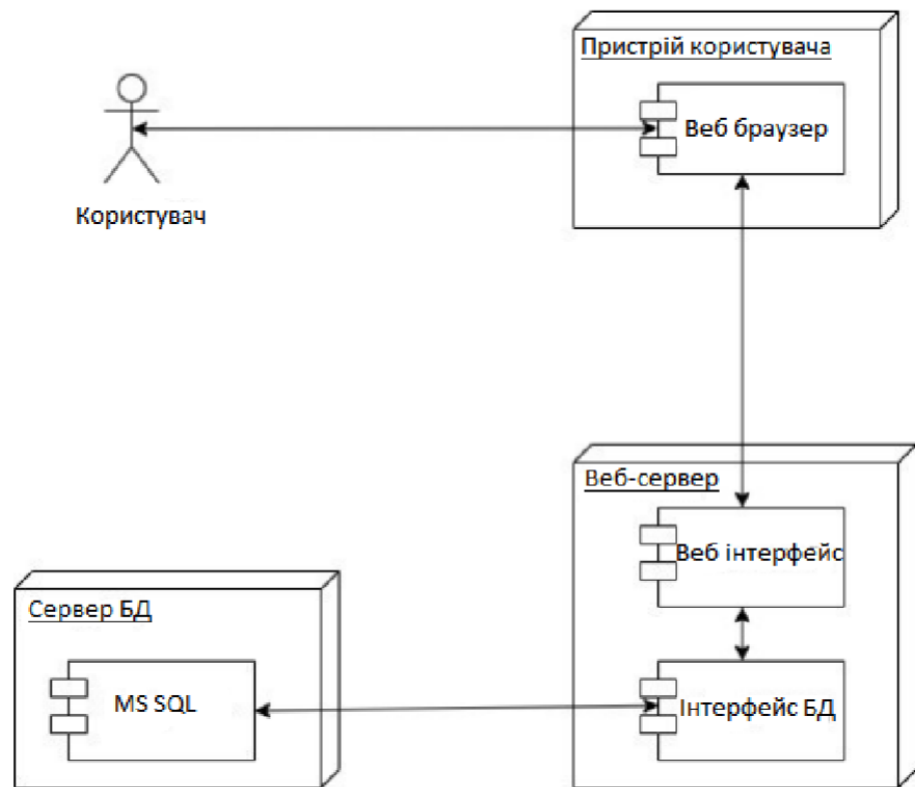


Рисунок 3.14 – UML-діаграма компонентів

Висновки

Таким чином представлено структуру інформаційної системи управління навчальним процесом освітньої установи та здійснено її реалізацію у вигляді програмного забезпечення. Також для візуалізації функціонування розробленої системи представлені UML діаграм варіантів використання, послідовності та компонентів.

ВИСНОВКИ

Інформаційна система управління навчальним процесом освітньої установи є надзвичайно важливою в сучасному освітньому середовищі. Зростаюча складність та обсяги даних, пов'язані з навчальними програмами, студентами, викладачами, розкладами та іншими аспектами управління, вимагають ефективного та організованого підходу до керування навчальними процесами. Такі системи допомагають автоматизувати та оптимізувати рутинні адміністративні процеси, забезпечуючи зручний доступ до інформації для адміністраторів, викладачів, студентів та інших учасників навчального процесу. Вони сприяють управлінню ресурсами, плануванню розкладу занять, моніторингу академічної успішності студентів, організації екзаменаційних сесій та багато іншого. Тому в результаті виконання даної кваліфікаційної роботи було спроектовано та реалізовано інформаційну систему управління навчальною діяльністю освітньої установи.

В першому розділі розглянуто концепції та архітектури інформаційних систем. Здійснено огляд відомих освітніх платформ управління навчальною діяльністю. Виокремлено їх переваги та недоліки. Досліджено етапи створення інформаційних систем.

В другому розділі проведено аналіз технологій та засобів для розробки інформаційної системи управління навчальним процесом освітньої установи, розглянуто їх переваги та недоліки. Визначено функційні та нефункційні вимоги до інформаційної системи управління навчальним процесом освітньої установи. Обрано стек технологій для реалізації інформаційної системи. Проведено проектування концептуальної бази даних, визначено сутності та їх атрибути. Створення концептуальної моделі бази даних дало змогу описати структуру та взаємозв'язки в предметній області, а також підготувати основу для подальшого розроблення та інформаційної системи. Спроектовано ERD діаграму.

В третьому розділі наведено структуру запропонованої інформаційної системи управління навчальним процесом освітньої установи. Відповідно до цієї

структури було спроектовано та реалізовано у вигляді програмного забезпечення інформаційну систему управління навчальним процесом освітньої установи. Реалізована інформаційна система має наступні функції: відображення даних у вигляді звітів про студентів, викладачів, оцінки, предмети, а також екзамени. Передбачено можливість додавання, видалення та оновлення інформації про відповідні сутності, а також реалізовано фільтрацію даних за різними полями. Додано функцію авторизації та визначено дві ролі – студент та викладач. Окрім того варто відзначити досить легкий та зручний інтерфейс, що дозволяє легко зорієнтуватись у функціях інформаційної системи. Також для візуалізації функціонування системи представлені UML діаграм варіантів використання, послідовності та компонентів.

В якості недоліків спроектованої інформаційної системи можна виокремити відсутність локалізації (вибір мови інтерфейсу) та відсутність налаштувань інтерфейсу під конкретного користувача.

					КвРІСТ. 200182.20.01.06 ПЗ	Арк.
						60
Зм.	Арк.	№докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Томашевський О.М., Цегелик Г.Г., Вітер М.Б., Дубук В.І. Інформаційні технології та моделювання бізнес процесів: навч. посіб. для студ. ВНЗ. К.: *Центр уч. ліри*, 2012., 295 с.
2. Якість та тестування інформаційних систем. Навчальний посібник для самостійної роботи студентів вищих навчальних закладів. Київ: *ННІТ ДУТ*, 2020. 128 с.
3. Азарова А.О., Поплавський А.В. Інформатика та комп'ютерна техніка навч. посіб., Вінниц. нац. техн. ун-т., Вінниця: *ВНТУ*, 2012. Ч. 1, 360 с.
4. Лагута В. В. Підвищення якості кібернетичної безпеки в інформаційно-телекомунікаційній системі підприємства. *Сучасний захист інформації*. 2020. № 1. С. 37-41.
5. Галахов Є. М. Стратегічні пріоритети системи інформаційної безпеки підприємства, що залучає фріланс-ресурс. *Сучасний захист інформації*. 2019. № 3. С. 30-35.
6. Медушевський С. Автоматизовані інформаційні системи як елементи системи управління якістю фармацевтичного підприємства. *Технічні науки та технології*. 2020. № 4. С. 65-71.
7. Прищепа Є. А. Метод розрахунку часу транзакцій у розподілених комп'ютерних мережах при побудові інформаційних систем підприємств. *Вісник університету "Україна"*. 2016. № 1. С. 175-180.
8. Павлик В. П. Систематизація методів у інформаційному забезпеченні управління підприємствами. *Економіка АПК*. 2020. № 1. С. 95-100.
9. Мінухін С. В. Методи і моделі проектування на основі сучасних CASE– засобів. Навчальний посібник, Харків: Вид. ХНЕУ, 2008. 272 с.
10. Гафіяк А. М. Інноваційні методи використання CRM-систем для підвищення рівня підготовки фахівців з інформаційно-комунікаційних технологій. *Духовність особистості: методологія, теорія і практика*. 2019. Вип. 3. С. 53-61.

					КВРІСТ. 200182.20.01.06 ПЗ	Арк. 61
Зм.	Арк.	№докум.	Підпис	Дата		

11. Черняк О. І. Нечіткий підхід до оцінювання рівня інформаційних ризиків у CRM-системах. *Нейро-нечіткі технології моделювання в економіці*. 2016. № 5. С. 199-232.

12. Павлиш В. А. Основи інформаційних технологій і систем, Львів : Львівська політехніка, 2018. 619 с.

13. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем: навчальний посібник, Черкаси: *Черкаський національний університет ім. Б. Хмельницького*, 2017. 434 с.

14. Мулеса О.Ю. Інформаційні системи та реляційні бази даних. Навч.посібник. Електронне видання, 2018. 118 с

15. Степанов М. М. Криптографічний захист інформації, що циркулює в інформаційних ресурсах ERP-систем. *Зв'язок*. 2016. № 2. С. 60-63.

16. Костенко О. Б. Організація баз даних та знань : конспект лекцій (для студентів денної та заочної форм навчання першого (бакалаврського) рівня вищої освіти за спеціальністю 126 – Інформаційні системи та технології), Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. Харків : ХНУМГ ім. О. М. Бекетова, 2021. 92 с.

17. Берко А. Ю. Системи баз даних та знань : підручник Львів : Магнолія 2006, 2015. Книга 1. *Організація баз даних та знань*. 440 с.

18. Шаховська Н. Б. Системи штучного інтелекту : навч. посібник, Львів : Львівська політехніка, 2018. 391 с.

19. Костріков С. В. Інформаційні технології в БД. Навчально-методичний посібник, Харків : РВВ ХНУ, 2015. 56 с.

20. Antoniou J. Quality of Experience and Learning in Information Systems, *Springer Cham*, 2019. 110 p.

21. Mikalsen, M., Moe, Nils, B., Wong, Sut I., Stray, Viktoria. Agile Information System Development Organizations Transforming to Large-Scale Collaboration. *In proceedings of the Forty-Second International Conference on Information Systems (ICIS 2021)*, Austin, USA. 2021.

					КВРІСТ. 200182.20.01.06 ПЗ	Арк. 62
Зм.	Арк.	№докум.	Підпис	Дата		

22. Muhammad K., Maryati M.. Lean IT Transformation Plan for Information Systems Development. *International Journal of Advanced Computer Science and Applications*. 2020. Vol 11 (8).

23. What is Google Classroom? URL: <https://www.techlearning.com/features/what-is-google-classroom>

24. Schoology's main features: a complete educational platform, URL: https://polaridad.es/en/schoology-features/#google_vignette

25. Getting Started on Schoology - For Instructors, URL: <https://uc.powerschool-docs.com/en/schoology/latest/getting-started-on-schoology-for-instructors>

26. A. Bajaj, et al. Information Technology and Systems - I Systems Analysis and Design: Should We Be Researching What We Teach? *Commun. Assoc. Inf. Syst.* 15 (2005), 27.

27. Zetter K. Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon. *Broadway Books*, 2015. – 448 p.

28. Solomon M.G. Fundamentals of Communications and Networking. *Jones & Bartlett Learning*, 2014. 512 p.

29. Geeksforgeeks, Architecture of a System, URL: <https://www.geeksforgeeks.org/architecture-of-a-system/>

30. Antoniou J. Quality of Experience and Learning in Information Systems, *Springer Cham*, 2019. 110 p.

31. Isyaku I. Iterative and Incremental Development Analysis Study of Vocational Career Information Systems. *International Journal of Software Engineering & Applications*. 11. 2020. pp. 13-24.

32. Chan O. SQL: Learn SQL (using MySQL) in One Day and Learn It Well. SQL for Beginners with Hands-on Project, 2018. 166p.

33. EdApp, 13 Educational Platforms, URL: <https://www.edapp.com/blog/educational-platforms/>

34. Turing, Software Architecture Patterns: What Are the Types and Which Is the Best One for Your Project, URL: <https://www.turing.com/blog/software-architecture-patterns-types/>

35. Hassanien A. E., Elhoseny M. Cybersecurity and Secure Information Systems. Challenges and Solutions in Smart Environments, *Springer Cham*, 2019. p. 314.

36. Medium, Types of System Architectures, URL: <https://medium.com/@abhijitgunjal1648/types-of-system-architectures-eb37dd496971>

37. Марголін О. UML для бізнес-моделювання: для чого потрібні діаграми процесів, URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html>

38. Катрич Д. В. Захист інформації в ERP-системі підприємства. *Адаптивні теми автоматичного управління*. 2017. № 2. С. 17-25.

39. Мищенко А. В. Нечітка модель оцінки ризиків інформаційної безпеки та підтримки рівня захищеності ERP-систем. Телекомунікаційні та інформаційні технології. 2020. № 1. С. 142-151.

40. Mikalsen, M., Moe, Nils, B., Wong, Sut I., Stray, Viktoria. Agile Information System Development Organizations Transforming to Large-Scale Collaboration. *In proceedings of the Forty-Second International Conference on Information Systems (ICIS 2021)*, Austin, USA. 2021.

41. Nachenberg, C. Understanding heuristics: Symantec's bloodhound technology, *Symantec White Paper Series*. 1998. Vol. 34. P. 17.

42. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій) [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська. – Київ : КПІ ім. Ігоря Сікорського, 2020. 192с.

43. Hassanien A. E., Elhoseny M. Cybersecurity and Secure Information Systems. Challenges and Solutions in Smart Environments, *Springer Cham*, 2019. p. 314.

44. Nachenberg, C. Understanding heuristics: Symantec's bloodhound technology, *Symantec White Paper Series*. 1998. Vol. 34. P. 17.

45. Perriot F. Principles and practise of xraying, *The 14-th Virus Bulletin International Conference*, September 12, 2004: Proceedings. – 2004. – Pp. 51-56.
46. Kim D. Fundamentals of Information Systems Security 3rd Edition, *Jones & Bartlett Learning*, 2016. 571 p.
47. Zetter K. Countdown to Zero Day: Stuxnet and the Launch of the World’s First Digital Weapon. *Broadway Books*, 2015. 448 p.
48. Solomon M.G. Fundamentals of Communications and Networking. *Jones & Bartlett Learning*, 2014. – 512 p.
49. TeachTarget, Information systems, URL: <https://www.techtarget.com/whatis/definition/IS-information-system-or-information-services>
50. Medium, Build a Basic FullStack app using C#(.NET Core), MySQL and Webpack/React. (1/3)(Updated), URL: <https://alialhaddad.medium.com/build-a-basic-fullstack-app-using-c-net-core-mysql-and-webpack-react-1-3-73cfb64daeb6>
51. .NET 7.0 + Dapper + MySQL - CRUD API Tutorial in ASP.NET Core, URL: <https://jasonwatmore.com/net-7-dapper-mysql-crud-api-tutorial-in-aspnet-core>
52. Chapple M. CISSP Certified Information Systems Security Professional, *Sybex*, 2018. 1616 p.
53. Mayer-Schönberger V. Big Data: A Revolution That Will Transform How We Live, Work, and Think, *Eamon Dolan.Mariner Books*, 2014. 272 p.
54. Antoniou J. Quality of Experience and Learning in Information Systems, *Springer Cham*, 2019. 110 p.
55. Jena A. K. Automated Software Testing: Foundations, Applications and Challenges (Services and Business Process Reengineering), *Springer* 1st ed., 2020. 179 p.
56. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій) [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп’ютерні науки» / КПІ ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська. Київ : КПІ ім. Ігоря Сікорського, 2020. 192с
57. Адміністрування баз даними URL: <http://firebirdsql.org/manual/ru/migration-mssql-db-admin-ru.html>.

58. Ben Forta. SQL in 10 Minutes a Day, *Sams Teach Yourself*: Sams Publishing; 5th edition. 18 Aug. 2020. 256p.

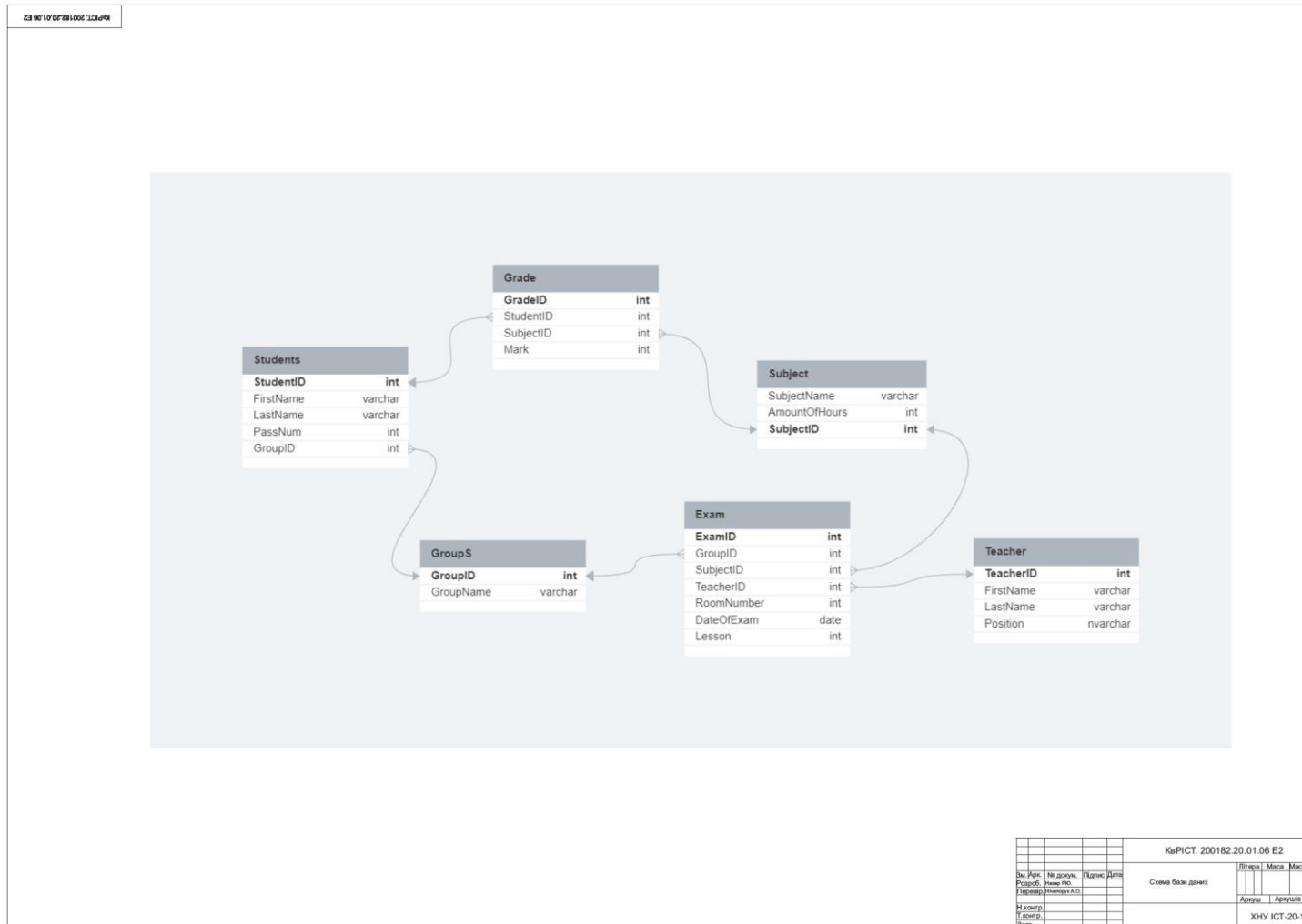
59. Гайдаржи В., Ізварін І. Бази даних в інформаційних системах: Навчальний посібник. Тернопіль: Навчальна книга. 2018. 418 с.

60. Create MySQL Database: A Step-by-Step Guide, URL: <https://www.singlestore.com/blog/create-mysql-database/>

					КВРІСТ. 200182.20.01.06 ПЗ	Арк.
						66
Зм..	Арк.	№докум.	Підпис	Дата		

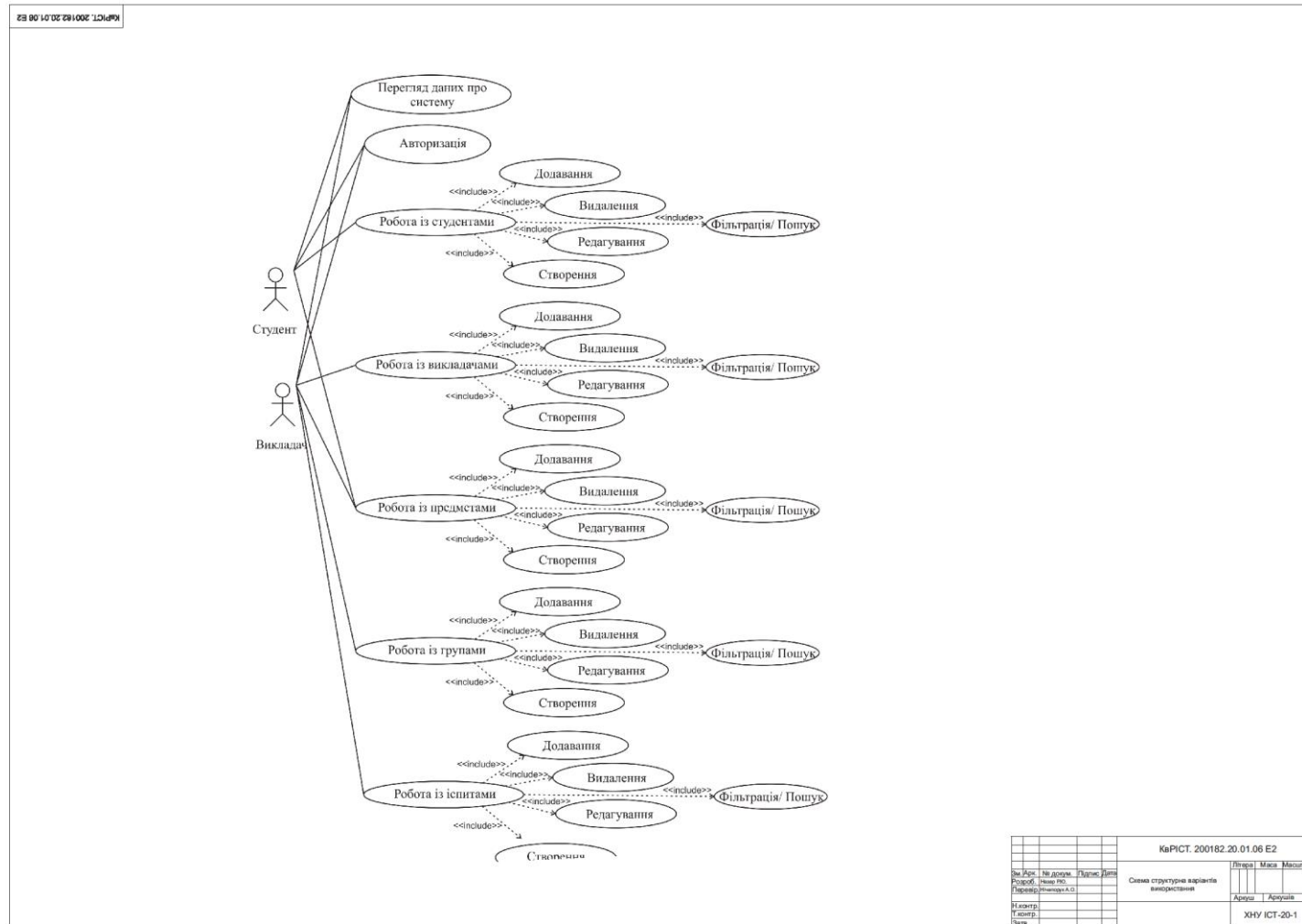
ДОДАТОК А

Копія креслення «Схема бази даних»



ДОДАТОК Б

Копія креслення «Схема структурна варіантів використання»



Ім'я користувача:
Кафедра КІ

ID перевірки:
1016235096

Дата перевірки:
07.05.2024 19:54:48 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
07.05.2024 20:51:30 EEST

ID користувача:
100005591

Назва документа: Назар_Інформаційна система управління навчальним процесом освітньої установи

Кількість сторінок: 68 Кількість слів: 10899 Кількість символів: 85423 Розмір файлу: 4.19 MB ID файлу: 1016016233

14.6% Схожість

Найбільша схожість: 2.65% з Інтернет-джерелом (<http://uadoc.zavantag.com/text/10589/index-1.html>)

14.2% Джерела з Інтернету

284

Сторінка 70

3.87% Джерела з Бібліотеки

62

Сторінка 73

0% Цитат

Цитати

1

Сторінка 74

Посилання

1

Сторінка 74

0% Вилучень

Немає вилучених джерел

Tue May 07 17:48:24 EEST 2024, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 3.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилоч в документах: 12%

ID: 125822 Назва: БКР Інформаційна система управління навчальним процесом освітньої установи Додано в БД: 2024-05-07 Автора: Р.Ю. Назар Керівники: А.О. Нічепорук Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	77528	669	5649 (7%)	69 (10%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Назар Роман Юрійович

Тема: Інформаційна система управління навчальним процесом освітньої установи

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг дипломної роботи:

Кількість листів креслень 3; кількість сторінок записки 55

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано інформаційну систему управління навчальним процесом освітньої установи

2. Висновок про відповідність роботи дипломному завданню _____

Дипломний проект відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз відомих засобів та інформаційних систем для навчальної діяльності. У другому розділі спроектовано інформаційну систему управління навчальним процесом освітньої установи. Розроблено концептуальну модель даних. У третьому розділі реалізовано інформаційну систему управління навчальним процесом освітньої установи

4. Позитивні сторони роботи: Спроектовано інформаційну систему та здійснено її реалізацію прототипу у вигляді програмного забезпечення. Спроектована система має досить зручний та простий інтерфейс.

5. Негативні сторони роботи: Доцільно було б реалізувати функцію, яка виконувала б фільтрацію розкладу іспитів по викладачам.

6. Оцінка графічного оформлення та пояснювальної записки роботи: пояснювальна записка та листи креслення виконані згідно діючих вимог

7. Відгук про роботу в цілому: В загальному робота виконана на достатньому рівні.

8. Інші зауваження: —

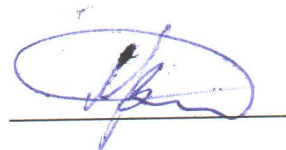
9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «добре» 3,75 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) —

Завідувач кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки, д.т.н., професор Мартинюк В. В

“ 13 ” 05 2024р.



Завідувачу кафедри КПС
д-р.техн.наук, проф. Говорущенко Т. О.

Назара Романа Юрійовича

ІІІБ здобувача вищої освіти

ФІТ, 4 курсу, групи ІСТ-20-1


ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

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

07 травня 2024 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна система управління навчальним процесом освітньої установи

Автор: Назар Роман Юрійович

Спеціальність: 126 – Інформаційні системи та технології

Освітня програма: освітньо-професійна

Науковий керівник: Нічепорук Андрій Олександрович, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

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

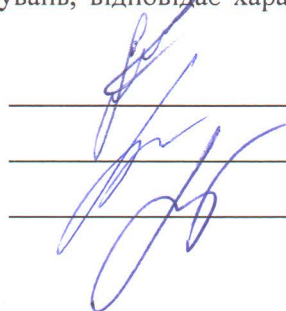
- 1) запозичення розміщені в розділі аналізу існуючих аналогів та відомих рішень, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) перелік використаних джерел було розпізнано як плагіат

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості Unicheck, складає 14,6 % і адресується до 394 першоджерела; та системою Anti-Plagiarism складає 3%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІІС



А.О. Нічепорук

Є.Г. Гнатчук

Т. О. Говорущенко