

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці**

ОСВІТНЬО ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ПРОЕКТ

на тему

**ІНФОРМАЦІЙНА СИСТЕМА НАВЧАЛЬНО-ОСВІТНЬОГО ЦЕНТРУ
РОЗВИТКУ**

здобувача Мозгового Дмитра Анатолійовича _____

Науковий керівник:

к.е.н., доцент

_____ Гордієнко І.В.

**Проект допущений до захисту перед
екзаменаційною комісією з атестації
здобувачів вищої освіти**

завідувач кафедри:

к.е.н., доцент

_____ Тішков Б.О.

Київ 2024

Міністерство освіти і науки України
Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО_ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проектної групи(гарант)
освітньо-професійної програми

Іванченко Г.Ф.
“ ____ ” _____ 2024 р.

ЗАТВЕРДЖУЮ:
Завідувач кафедри

Тішков Б.О.
“ ____ ” _____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти *Мозгового Дмитра Анатолійовича*
очної (денної) форми навчання

на підготовку кваліфікаційного бакалаврського проекту

на тему: «Інформаційна система навчально-освітнього центру розвитку»

Тему затверджено наказом ректора Університету від « 11 » березня 2024 р.
№ 529-ст.

Кваліфікаційний бакалаврський проект виконується на матеріалах
наукових публікацій, інтернет-ресурсів, технічної документації

План кваліфікаційного бакалаврського проекту

Розділ I Характеристика та аналіз предметної галузі

Розділ II Розробка вимог і моделювання інформаційної системи

Розділ III Проектування та реалізація компонентів системи

Об'єкт дослідження процеси ефективного управління та документообігу в галузі надання послуг дистанційної освіти

Предмет дослідження сукупність теоретичних, методичних і прикладних аспектів з моделювання, проектування та програмування модулів для підвищення ефективності управління дистанційною освітою

Мета кваліфікаційного бакалаврського проекту систематизація і узагальнення сучасного досвіду з розробки нових рішень в галузі дистанційної освіти та розробка власного прототипу інформаційної системи

Конкретні завдання, які студент повинен виконати для досягнення поставленої мети:

У розділі I Описати предметну галузь управління дистанційною освітою. Навести характеристику об'єкта дослідження. Подати результати аналізу літературних джерел та існуючих варіантів інформаційних систем і технологій для дистанційної освіти

У розділі II Визначити бізнес-вимоги, функціональні та нефункціональні вимоги до проєктованої системи та подати їх у формі діаграм і проєктної специфікації, розроблених з використанням CASE-засобів. Подати постановку та алгоритм розв'язання задачі надання послуг дистанційної освіти. Розробити моделі системи у специфікації мови UML

У розділі III Спроектувати базу даних та елементи інформаційного забезпечення інформаційної системи навчально-освітнього центру розвитку. Обґрунтувати вибір та розташування комплексу технічних засобів на об'єкті управління. Описати структуру та складові програмного забезпечення системи дистанційної освіти. Висвітлити результати реалізації інформаційної системи дистанційної освіти

Завдання підготував
науковий керівник _____

Гордієнко Ірина Василівна
“ _____ ” _____ 2024 р.

Завдання одержав
студент _____

Мозговий Дмитро Анатолійович
“ _____ ” _____ 2024 р.

Відгук
про кваліфікаційний бакалаврський проєкт
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми
«Комп'ютерні науки»

Мозгового Дмитра Анатолійовича

на тему: «Інформаційна система навчально-освітнього центру розвитку»

1. **Актуальність теми:** Останнім часом дистанційне навчання здобуває все більш вагомe місце у галузі освіти. Існують різноманітні підходи до проєктування інформаційних систем дистанційного навчання та інформаційні технології для їх реалізації. Тому тематика кваліфікаційного бакалаврського проєкту Мозгового Д.А., присвяченого проєктуванню сучасної інформаційної системи навчально-освітнього центру розвитку, є актуальною.

2. **Позитивні риси кваліфікаційного бакалаврського проєкту:** Кваліфікаційний проєкт містить результати дослідження здобувачем предметної області дистанційного навчання, відображає сучасні підходи до розробки інформаційної системи навчально-освітнього центру розвитку, демонструє володіння здобувачем інформаційно-технологічним інструментарієм розробки ІС, зокрема, Visual Studio Code 2022, СУБД MS SQL Server, мовами програмування Typescript і C#, фреймворками Next.js, ASP.NET Core.

3. **Наявність самостійних розробок автора:** Здобувач самостійно глибоко проаналізував і визначив вимоги до ІС навчально-освітнього центру розвитку, спроектував відповідні діаграми і специфікації, підготував постановку і алгоритм розв'язання задачі автоматизації дистанційного навчання, здійснив моделювання та розробку основних видів забезпечення інформаційної системи.

4. **Цінність теоретичних висновків та практичних рекомендацій:** Розроблена здобувачем інформаційна система за умови відповідного розвитку і удосконалення може бути рекомендована для практичного використання в освітній галузі.

5. **Наявність недоліків:** у тексті роботи залишились не виправлені помилки і описки.

6. **Загальна оцінка кваліфікаційного бакалаврського проєкту та його допущення до захисту перед ЕК:** Кваліфікаційний бакалаврський проєкт відповідає методичним вимогам, заслуговує позитивної оцінки і може бути допущений до захисту перед ЕК.

Науковий керівник _____ доцент, к.е.н. Гордієнко І.В.
“ ” _____ 2024 р.

Рецензія

на кваліфікаційний бакалаврський проєкт

Мозгового Дмитра Анатолійовича

тема

«Інформаційна система навчально-освітнього центру розвитку»

Актуальність теми кваліфікаційного бакалаврського проєкту і доцільність його розроблення. Тематика проєкту є досить актуальною, розробка інформаційної системи для навчально-освітнього центру розвитку має важливе значення в сучасних умовах. Тенденції розвитку технологій, все частіше вимагають від освітніх закладів впровадження цифрових рішень, тому розробка такої платформи є цілком доцільною та підходить під потреби сьогодення.

Якість проведеного дослідження. Дослідження проведене на високому високому рівні, з дотриманням усіх поставлених завдань. Проведений аналіз наявних джерел якісний, з певним урахуванням функціоналу вже наявних систем.

Позитивні риси кваліфікаційного бакалаврського проєкту. Тема проєкту є досить актуальною в сучасних умовах та має значний практичний потенціал. Варто також відзначити детальний аналіз предметної галузі та існуючих рішень, на основі яких і проводилось проєктування.

Зауваження. У проєкті можна було детальніше описати та провести мінімальну порівняльну характеристику з розробленою системою.

Практична значимість висновків і рекомендацій. Надані висновки цілком описуються переваги розробленої системи. Вони стануть корисними при розробці платформ з подібним функціоналом або розширенням наявного функціоналу вже у існуючих реалізаціях.

Керівник
відділу розробки та тестування
ТОВ «АДВЕРТАЙЗІНГ»



(підпис)

Рибак Д.В

АНОТАЦІЯ

Кваліфікаційного бакалаврського проекту студента 4 курсу
Навчально-наукового інституту «Інститут інформаційних технологій в економіці»

Мозгового Дмитра Анатолійовича, виконаний на тему:

«Інформаційна система навчально-освітнього центру розвитку» Київ: кафедра
інформаційних систем в економіці, 2024 р.

Кваліфікаційний бакалаврський проект присвячений актуальній проблемі автоматизації навчального процесу у дистанційному форматі, яку пропонується розв'язувати та удосконалювати з використанням сучасних інформаційних технологій.

У першому розділі подана характеристика предметної галузі й об'єкта дослідження, наведено аналіз задач, що розв'язуються, а також наводиться перелік існуючих програмних засобів даної предметної галузі.

Другий розділ присвячений обґрунтуванню методу проектування системи, розробленню її архітектури, виконанню постановки та розробленню алгоритму розв'язання задач.

У третьому розділі наведено інформаційне, технічне, програмне забезпечення для розробленої інформаційної системи. Продемонстровано та аргументовано комплекс засобів який використовувався при реалізації частин системи.

Висновки містять рекомендації щодо доцільності розроблення та впровадження інформаційної системи навчально-освітнього центру розвитку.

РЕФЕРАТ

Даний кваліфікаційний бакалаврський проект містить 81 сторінку, 40 рисунків, 13 таблиці, перелік використаних джерел у кількості: 24, та 1 додаток.

ІНФОРМАЦІЙНА СИСТЕМА НАВЧАЛЬНО- ОСВІТНЬОГО ЦЕНТРУ РОЗВИТКУ

Предметом дослідження є сукупність теоретичних, методичних і прикладних аспектів з моделювання, проектування та програмування модулів для підвищення ефективності управління дистанційною освітою.

Об'єктом дослідження є процеси ефективного управління та документообігу в галузі надання послуг дистанційної освіти

Мета кваліфікаційного бакалаврського проекту є систематизація і узагальнення сучасного досвіду з розробки нових рішень в галузі дистанційної освіти та розробка власного прототипу інформаційної системи.

Апаратні та програмні засоби, що використовувались при проектуванні: інтернет-ресурс Figma, Lucid, середовища розробки Visual Studio Code 2022 та Visual Studio 2022, мови програмування: Typescript, C#, фреймворки: Next.js, ASP.NET Core та мови розмітки HTML та стилів CSS.

Розроблена інформаційна система навчально-освітнього центру розвитку спроектована для виконання поставлених задач. Реалізовані рівні доступів у системі дозволяють ефективно керувати процесами у системі. Студенти мають змогу безкоштовно доєднуватись до обраних навчальних матеріалів, за якими вже закріплені викладачі-практики. Вчителі мають достатній функціонал для створення повноцінних освітніх програм та повний набір інструментів, для перевірки та оцінки знань студентів. Також, під час кваліфікаційного бакалаврського проектування було визначено та присвоєно загальну назву інформаційній системі - «**StudyEra**»

Отримані результати можуть бути використані компаніями, які планують організувати та проводити безкоштовні онлайн заняття для своїх майбутніх студентів.

Рік виконання кваліфікаційного бакалаврського проекту: 2024 р.

Рік захисту кваліфікаційного бакалаврського проекту: 2024 р.

Перелік скорочень у а кваліфікаційному бакалаврському проекті

Абревіатура	Роз'яснення
PK	Primary key
FK	Foring key

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	5
1.1 Характеристика предметної галузі та об'єкта дослідження.....	5
1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі.....	6
РОЗДІЛ 2 РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	12
2.1 Аналіз і специфікація вимог до інформаційної системи	12
2.1.1 Бізнес-вимоги до системи.....	12
2.1.2 Функціональні вимоги до системи.....	15
2.1.3 Нефункціональні вимоги до системи.....	21
2.2 Постановка задачі та алгоритм розв'язання задачі	21
2.2.1 Характеристика задачі.....	26
2.2.2 Вихідна інформація.....	28
2.2.3 Вхідна інформація	28
2.2.4. Математичне забезпечення та алгоритм функціонування системи.....	29
2.3 Моделювання інформаційної системи.....	30
РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ	35
3.1 Інформаційне забезпечення	35
3.2 Технічне забезпечення.....	42
3.3 Програмне забезпечення	43
3.4 Результати реалізації інформаційної системи.....	46
ВИСНОВКИ.....	57
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
ДОДАТКИ А.....	61
ДОДАТКИ Б.....	76

ВСТУП

Актуальність теми. У сучасному світі дистанційна освіта набуває все більшої популярності та значущості, особливо в умовах глобальних викликів, таких як пандемія COVID-19 та війна у нашій країні. Розвиток інформаційних технологій відкриває нові можливості для забезпечення безперебійного навчального процесу незалежно від фізичного місцезнаходження учасників освітнього процесу. Інформаційні системи відіграють ключову роль у забезпеченні ефективної організації та управління освітніми процесами, сприяючи підвищенню якості надання онлайн освіти та доступності навчальних матеріалів для усіх охочих.

Предмет дослідження. Предметом дослідження є сукупність теоретичних, методичних і прикладних аспектів моделювання, проектування та програмування модулів для підвищення ефективності управління дистанційною освітою.

Об'єкт дослідження. Об'єктом дослідження є процеси ефективного управління та документообігу в галузі надання послуг дистанційної освіти.

Методи та інструментарій дослідження. Методи дослідження предметної області, за для підвищення ефективності проектування системи, включають аналіз літературних джерел, моделювання з використанням UML, а також методи проектування та розробки програмного забезпечення. Інструментарій дослідження включає середовища розробки Visual Studio Code та Visual Studio, інструменти моделювання Figma та Lucid, мови програмування TypeScript та C#, а також фреймворки Next.js та ASP.NET Core.

Мета дослідження. Мета кваліфікаційного бакалаврського проекту полягає у систематизації і узагальненні сучасного досвіду з розробки нових рішень у галузі дистанційної освіти та розробці власного прототипу інформаційної системи.

Були визначені наступні завдання:

- Описати предметну галузь управління дистанційною освітою.
- Навести характеристику об'єкта дослідження.
- Подати результати аналізу літературних джерел та існуючих варіантів інформаційних систем і технологій для дистанційної освіти.

- Визначити бізнес-вимоги, функціональні та нефункціональні вимоги до проєктованої системи та подати їх у формі діаграм і проєктної специфікації.
- Розробити моделі системи у специфікації мови UML.
- Спроекувати базу даних та елементи інформаційного забезпечення інформаційної системи навчально-освітнього центру розвитку.
- Описати структуру та складові програмного забезпечення системи надання дистанційної освіти.
- Висвітлити результати реалізації інформаційної системи дистанційної освіти.

РОЗДІЛ 1 ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Характеристика предметної галузі та об'єкта дослідження

Дистанційна освіта стає все більш популярною у сучасному світі, забезпечуючи доступ до навчальних матеріалів незалежно від місця перебування студентів. Вона відкриває нові можливості для тих, хто не має можливості відвідувати традиційні навчальні заклади. Основні переваги дистанційної освіти включають гнучкість у виборі часу та місця навчання, доступ до широкого спектру навчальних ресурсів та можливість навчатись у власному темпі.

Дистанційна освіта може бути представлена у різних формах: онлайн-курси, вебінари, відеолекції, інтерактивні платформи та інші засоби електронного навчання. Вона забезпечує студентам можливість отримувати знання та навички з різних галузей науки та техніки, не виходячи з дому. Особливо актуальною дистанційна освіта стала під час пандемії COVID-19, коли багато навчальних закладів були змушені перейти на онлайн-формат.

Навчально-освітні центри є важливими інституціями у сфері освіти, що забезпечують навчання та професійний розвиток різних категорій населення. Вони можуть бути як традиційними, так і дистанційними. Традиційні навчально-освітні центри надають послуги у формі очних занять, семінарів, тренінгів та майстер-класів. Вони спрямовані на підвищення кваліфікації, професійний розвиток та перекваліфікацію фахівців у різних галузях.

Дистанційні навчально-освітні центри, у свою чергу, забезпечують навчання онлайн, без необхідності фізичного відвідування занять. Для залучення більш широкої аудиторії користувачів на інформаційну систему необхідно постійно оновлювати кількість навчального матеріалу та підвищувати його якість, залучаючи додаткових професійних спікерів, організовувати та проводити оффлайн зустрічі. Приклади навчально-освітніх центрів включають великі університети, що пропонують онлайн-програми, такі як Harvard Online Learning, Coursera, Udacity, та спеціалізовані платформи, які надають безкоштовні освітні ресурси для широкої аудиторії. Об'єктом

дослідження кваліфікаційного бакалаврського проекту є процеси ефективного управління та документообігу в галузі надання послуг дистанційної освіти. Включає аналіз і оптимізацію організаційних та технічних аспектів, що забезпечують ефективну роботу дистанційних навчально-освітніх центрів, таких як управління навчальними матеріалами, зручну взаємодію між учасниками освітнього процесу, оцінювання успішності студентів та підтримка адміністрування курсів.

1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі

У цьому підрозділі проведемо аналіз існуючих на даний момент платформ конкурентів. Розглянемо певний функціонал, який передбачений на платформах та в цілому реалізацію процесу навчання на освітніх платформах. Для проведення порівняльної характеристики, було обрано наступні інформаційні системи: Kwiga [1], Prometheus [2], SendPulse [3], Google Classroom [4],

Kwiga (рис. 1.1) – платформа, на якій можна створювати та керувати кількома незалежними проектами, що мають окремі команди, бази контактів, свої сайти та продукти. Kwiga також пропонує рішення для дистанційного навчання. На перший погляд, не зрозуміло, для яких конкретних цілей використовується платформа, але провівши більш поглиблений аналіз, стає зрозумілим, що вона також дозволяє організувати повноцінний формат онлайн навчання, починаючи від перегляду уроків у записі до проведення живих уроків, а також різні формати домашніх завдань та тестів для перевірки знань. Також, дана платформа має відкриту API документацію для розробників – що є великим плюсом для проведення дослідження та визначення методів проектування, які використовує компанія.

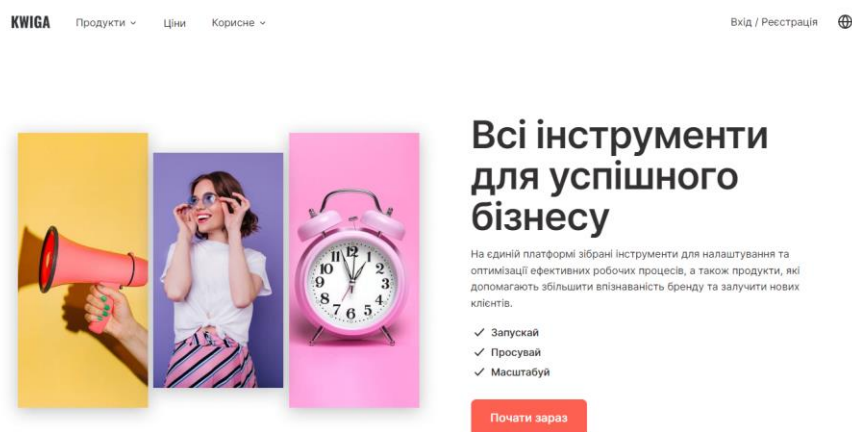


Рисунок 1.1 – Головна сторінка ресурсу «Kwiga»

Prometheus (рис. 1.2) – це найбільша платформа онлайн-курсів в Україні, яка пропонує широкий спектр курсів для різних напрямків і рівнів. Вона включає ІТ-курси (тестування, програмування, Front-end), бізнес-курси, курси англійської мови, підготовку до ЗНО, курси для HR та багато інших. Платформа надає як безкоштовні, так і платні курси, які підтримуються провідними викладачами з України та світу, забезпечуючи високий рівень освіти для всіх бажаючих.

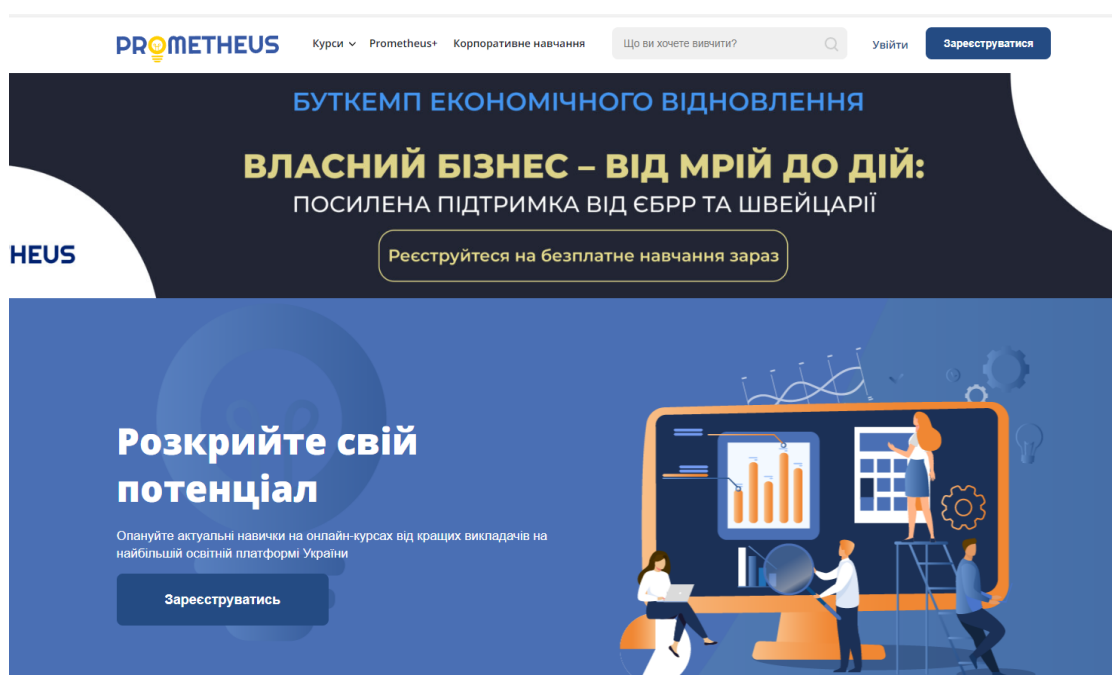


Рисунок 1.2 - Головна сторінка ресурсу «Prometheus»

SendPulse (рис.1.3) пропонує універсальні інструменти для створення онлайн-курсів. Вона дозволяє користувачам створювати, продавати та адмініструвати навчальні програми з використанням різноманітних каналів комунікації, таких як email, SMS, месенджери та веб-пуш повідомлення. Інструменти включають конструктор курсів та лендінгів, CRM для автоматизації продажів, можливість додавання модераторів та моніторинг успішності студентів. Мобільний додаток дозволяє проходити курси з будь-якої точки світу.

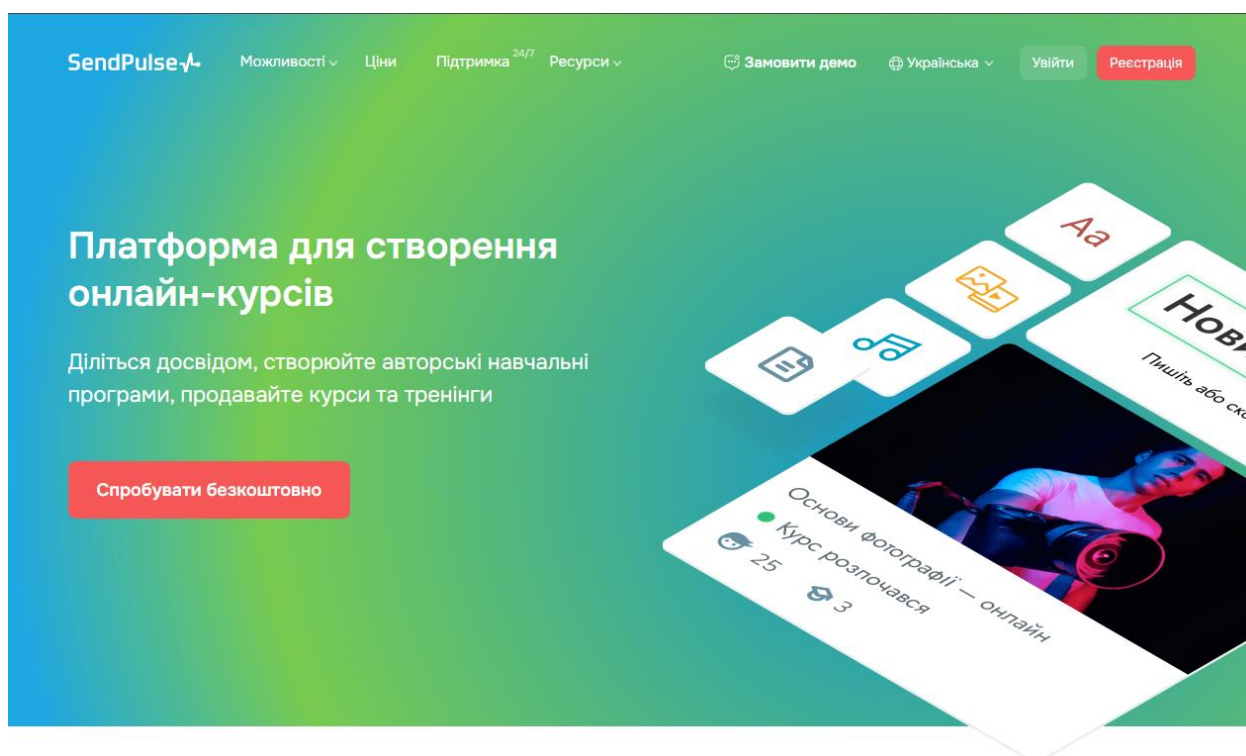


Рисунок 1.3 - Головна сторінка ресурсу «SendPulse»

«Google Classroom» (рис. 1.4) – це безкоштовний веб-сервіс, створений Google для шкіл, який спрощує створення, розповсюдження та оцінку завдань. Його метою є спрощення процесу обміну файлами між викладачами та студентами. Google Classroom інтегрується з іншими додатками Google, такими як Google Docs, Google Drive та Gmail, що дозволяє зручно керувати матеріалами та комунікацією.

У період пандемії яка почалась у 2019 році кожен студент/школяр та сотні вчителів почали знайомство з абсолютно новою для себе формою дистанційного навчання. Враховуючи обставини, які спіткали нашу планету – усі ми були вимушені працювати та навчатись віддалено.

Google Classroom безкоштовний для всіх користувачів, як для викладачів так і для студентів, взаємодіяти з інформаційною системою можна як і (через веб-версію) так і з мобільного телефону, за допомогою додатків одноіменного додатка.

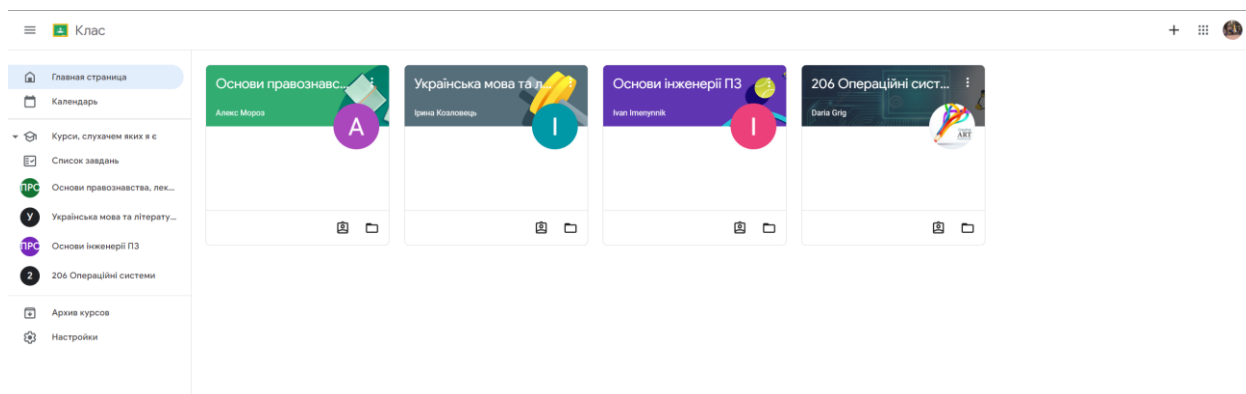


Рисунок 1.4 – Сторінка користувача з переліком класів

Провівши поверхневий аналіз, було створено певну порівняльну характеристику інформаційних систем, яка продемонстрована у Таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика програмних систем

Функціонал	Prometheus	Kwiga	SendPulse	Google Classroom
Можливість вибору декількох навчальних програм ІС	Так	Так	Так	Так
Отримання сертифіката по завершенню навчання на ІС	Так	Так	Ні	Так
Окрема людина – викладач, який закріплений за освітньою програмою	Так	Так	Так	Так
Особистий кабінет студента	Так	Так	Так	Так
На ІС передбачена здача домашніх завдань	Так	Так	Так	Так

Продовження Таблиці 1.1

Функціонал	Prometheus	Kwiga	SendPulse	Google Classroom
Відстеження прогресу учнів % переглянутого матеріалу	Так	Так	Так	Так
Хмарне сховище навчальних файлів	Так	Так (платне)	Так (Платне)	Так
Спільний чат студентів навчального курсу	Ні	Ні	Ні	Ні
Можливість відправити виконання завдання	Так	Так	Так	Так
Можливість перевірити виконане учнем завдання	Так	Так	Так	Так
Встановлення відкриття часу відкриття завдань	Так	Так	Так	Так
Функціонал безкоштовної версії обмежений	Так	Так	Так	Ні

Проведений аналіз показує, що кожна з розглянутих платформ має свої унікальні особливості та функціональні можливості, які роблять їх придатними для різних типів навчання та користувачів. Prometheus виділяється своїм широким вибором курсів та високою якістю викладання. Kwiga дозволяє створювати та керувати кількома незалежними проектами, надаючи гнучкі можливості для організації навчального процесу. SendPulse забезпечує багатоканальну комунікацію та інтеграцію з іншими сервісами, що робить її зручною для користувачів, які потребують ефективного адміністрування та маркетингу курсів. Google Classroom, завдяки своїй інтеграції з іншими сервісами Google, спрощує навчальний процес та забезпечує зручне управління матеріалами та комунікацією. Кожна з платформ має свої сильні сторони, що робить вибір конкретної системи залежним від специфічних потреб та умов навчання.

Проаналізувавши функціонал вже робочих систем, було зафіксовано основний ключовий функціонал, який варто реалізувати та інтегрувати схожі модулі в створену інформаційну систему.

РОЗДІЛ 2 РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Аналіз і специфікація вимог до інформаційної системи

2.1.1 Бізнес-вимоги до системи

Бізнес-вимоги визначають цілі та задачі, які повинна виконувати інформаційна система для підтримки навчально-освітнього центру розвитку. Вони формують основу для подальшого технічної розробки системи. Для розробленої системи, було визначено основних стейкхолдерів, а саме: студенти; вчителі; системні адміністратори.

Під час визначення бізнес вимог, варто виокремити основні аспекти, а саме:

- 1. Покращення якості освіти:** Забезпечити якісний доступ до навчальних матеріалів, курсів та ресурсів.
- 2. Збільшення охоплення аудиторії:** Розширити доступ до освіти для різних груп населення, включаючи тих, хто не має можливості навчатися офлайн.

Також, варто врахувати основні задачі системи:

- 1. Надання доступу до навчальних матеріалів:** Забезпечити студентам безперебійний доступ до лекцій, практичних занять, тестів та інших навчальних матеріалів.
- 2. Організація навчального процесу:** Підтримувати створення та управління курсами, оцінювання знань студентів.
- 3. Забезпечення комунікації:** Інтегрування зовнішніх платформ для ефективною комунікації між студентами та викладачами (а саме посилення на додаткові чати).

Розглянемо бізнес вимоги кожного стейкхолдера до проектованої інформаційної системи. Почнемо з вчителя, який працювати на розробленій платформі (Рис.2.1)

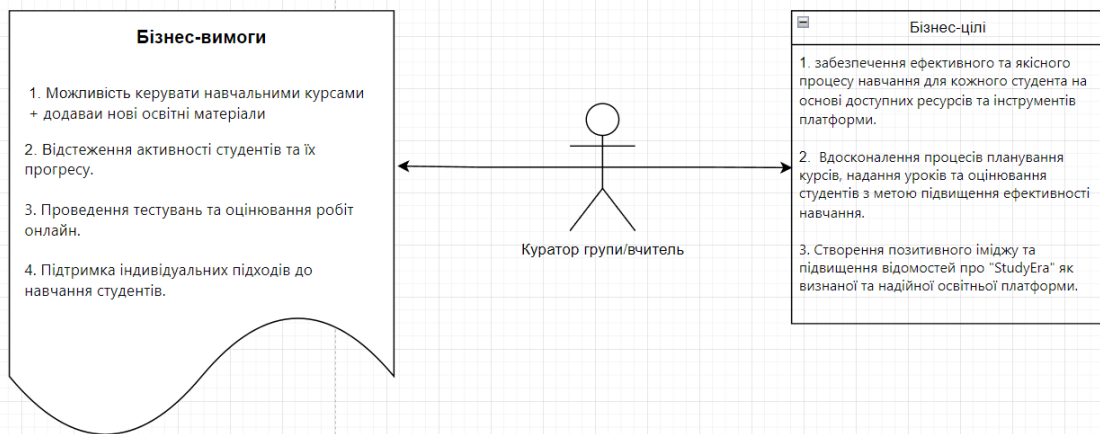


Рисунок 2.1 – Бізнес-вимоги та бізнес-цілі вчителя закріпленого за навчальним курсом

До переліку бізнес вимог, варто віднести такі ключові можливості взаємодії вчителя у інформаційній системі, а саме:

1. Однією з ключових можливостей, якій потрібно приділити увагу – надати можливість вчителю формувати навчальну програму, згідно з його побажань та бачення. Алгоритм взаємодії досить простий, але дуже ефективний – для того, щоб почати взаємодію з новоствореним курсом, адміністратор закріплює керуючого користувача (Вчителя), який має змогу заповнювати навчальний курс. Варто загадати і те, що навчальна програма має поділитись на окремі модулі та уроки за якими і закріплюється той чи інший матеріал.
2. Також для коректного налагодження процесу навчання та відстеження активності та успішності студентів, обов'язковим є введення системи оцінення виконаних завдань. Це у свою чергу, буде додатковим елементом мотивації для учнів.
3. На цей час, онлайн навчання в Україні набуло особливої важливості під час війни, оскільки забезпечує безперервність освітнього процесу незалежно від фізичних обмежень та безпеки. Воно також дозволяє студентам і викладачам залишатися в безпеці, зберігаючи доступ до необхідних ресурсів і матеріалів через інтернет.

4. Підтримка та індивідуальний підхід – показник якісного навчального продукту. Для розуміння потреб учнів та вибору під них оптимального рішення, було прийнято рішення про додавання джерела комунікації, а саме – чати у середовищах телеграм.¹

Розглянемо бізнес вимоги стейкхолдера під назвою студент (рис.2.2), який виступає у ролі активного користувача системи.

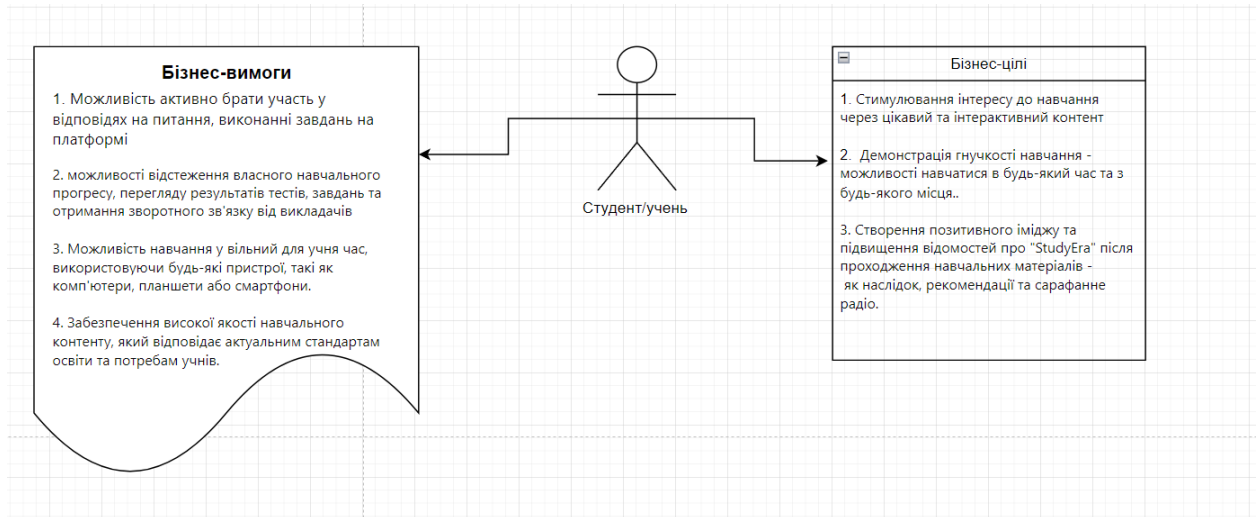


Рисунок 2.2 – Бізнес-вимоги та бізнес-цілі студентів на навчальному курсі

Розглянемо кожну з вимог окремо:

1. Важливим аспектом, будь-якого навчального процесу є взаємодії між ключовими користувачами [учень:вчитель]. Своєю чергою перший, повинен мати можливість бути залученим у навчальний процес, виконувати домашні завдання та отримувати за це справедливую оцінку від вчителя.
2. Кожен учень, котрий доєднується на навчальній програмі, повинен мати змогу, переглянути поточну успішність у % еквіваленті.
3. Користувачі [студенти] не прив'язані, не прив'язані до місця та можуть переглядати освітні матеріали з різних пристроїв, таких як: планшети на базі android чи iOS або смартфон з відповідними програмними забезпеченнями.
4. Матеріал поданий у відео роликах має бути корисним та змістовним, можливість вибору якості відео для перегляду може сягати до 1080р.

2.1.2 Функціональні вимоги до системи

Функціональні вимоги, які були визначені для розробленої інформаційної системи «StudyEra», її складові представлені на рис.2.3, опис функціональних вимог у таблиці 2.1.

Таблиця 2.1 – Функціональні вимоги до системи

Функціональна вимога	Опис вимоги	Реалізація в інформаційній системі
Управління користувачами інформаційної системи	Адміністратор повинен мати можливість створювати та видаляти облікові записи усіх учнів та усіх вчителів.	В адміністративній панелі передбачено перегляд наявних користувачів у системі, з подальшою можливістю додавання/видалення користувача з системи.
Реєстрація користувачів	Новий користувач інформаційної системи може з легкістю зареєструватись на платформі, в якій він одразу може заповнити дані про себе, такі як: ім'я, прізвище, номер телефону, дату народження та освітній ступінь.	Реєстрація особистого профілю на навчальній платформі має бути доступна абсолютно усім. Кожен студент повинен мати змогу створити свій унікальний обліковий запис, заповнити його своїми персональними даними. Також, він повинен мати можливість ці дані оновлювати.

Продовження Таблиці 2.1

Аутентифікація та авторизація	Захист доступу до системи враховуючи різні права доступів для різних користувачів системи	При створенні облікового запису, у розробленій системі різні типи облікових записів не можуть мати однаковий функціонал. Передбачена система ієрархії, має автоматично визначати тип користувача ще на процесі авторизації у систему шляхом введення логіну та пароля
Керування обліковими записами користувачів системи	Кожен користувач системи повинен мати можливість керувати своїм обліковим записом, змінювати та оновлювати власні дані.	Інформаційна система повинна передбачати особистий "профіль" користувача системи, в якому, він зможе взаємодіяти зі своїми даними, в залежності від потреби. Інтерфейсом облікового запису має бути передбачено можливість зміни персональних даних, таких як: фотографії користувача, номер телефону, прізвища, тощо..

Продовження Таблиці 2.1

<p>Редагування та оновлення навчальних курсів</p>	<p>Користувачі з рівнем доступу [вчитель], можуть самостійно здійснювати наповнення навчального курсу згідно з обраною методологією. Додаючи такі поля взаємодії як модулі, уроки, навчальні відео матеріали, додаткові матеріали тощо</p>	<p>У інформаційній системі потрібно передбачати такий функціональний набір, за допомогою якого, користувач з рівнем доступу [вчитель], зможе здійснювати маніпуляції з навчальним курсом до якого він закріплений. Мають бути передбачені наступні поля, такі як: додавання нового модуля, додавання уроку, зміна назви навчального курсу, зміна логотипа навчального курсу. Крім цього, користувач повинен мати можливість закріплення унікальних посилань на чати, які пов'язані з іншим додатком.</p>
<p>Управління навчальними курсами які відображені на головній панелі інформаційної системи</p>	<p>Платформа має дозволяти користувачам рівнем доступу: - [адміністратор] створювати навчальні програми та закріплювати за ними створеного вчителя. - [вчитель]: можливість повного управління навчальним курсом.</p>	<p>В інформаційній системі має бути передбачений алгоритм створення нових навчальних курсів, а також інструменти для подальшого редагування наповнення їх.</p>

Продовження Таблиці 2.1

Планування розкладу занять	Користувачі з рівнем доступу [вчитель] має змогу визначати час та дату проведення навчальної лекції за допомогою додаткових каналів комунікації, таких як спільний чат у зовнішньому додатку "Google calendar"	У інформаційній системі, користувач [вчитель] має змогу закріплювати графік розкладу занять за спеціальною кнопкою, при натисканні на яку, кожен з учнів може отримати графік майбутніх лекцій які будуть проходити у режимі онлайн.
Взаємодія користувачів курсу	Для підвищення ефективності навчального процесу варто також і визначити середовище для комунікації учнів та вчителя. Для цього і буде використовуватись зовнішній додаток, який не пов'язаний з розброленою системою	У інформаційній системі має бути передбачена функціональна кнопка, при натисканні на яку кожен з учасників курсу зможе доєднуватись до спільноти студентів та вчителів даного курсу.
Обговорення навчального матеріалу	У процесі навчання також важливим аспектом відіграє зворотний зв'язок учнів на викладений матеріал. За допомогою зовнішнього додатка (чату), кожен зі студентів матиме змогу поділитись своїми враженнями від уроки або задати питання напряду вчителю	Платформа має надати можливість студенту безперешкодно доєднатись до спільного чату зовнішнього додатку

Продовження Таблиці 2.1

Групові завдання	Студенти у загальному чаті для комунікації, матимуть змогу виконувати практичні завдання об'єднуючись в пари. (У випадку, якщо такі завдання передбачені навчальною програмою)	Платформа має надати можливість студент безперешкодно доєднатись до спільного чату зовнішнього додатку для виконання групового завдання
Сповіщення на навчальній платформі	Усі користувачі системи мають отримувати сповіщення про успішне доєднання до того чи іншого курсу.	Платформа має повідомляти студент про активність на платформі, яка була виконанна. Усі сповіщення повинні зберігатись у окремому відповідному блоці, куди кожен з учнів повинен мати доступ
Відстеження прогресу користувачів	Впродовж навчального процесу на обраному курсі, після перегляду відео, студент повинен отримувати оновлену інформацію про % пройденого навчального матеріалу на курсі.	На платформі потрібно передбачити лічильник у %, який закріплюється за кожним учнем, котрий доєднався до курсу. Цей показник має відображати відсоток переглянутого матеріалу на платформі.

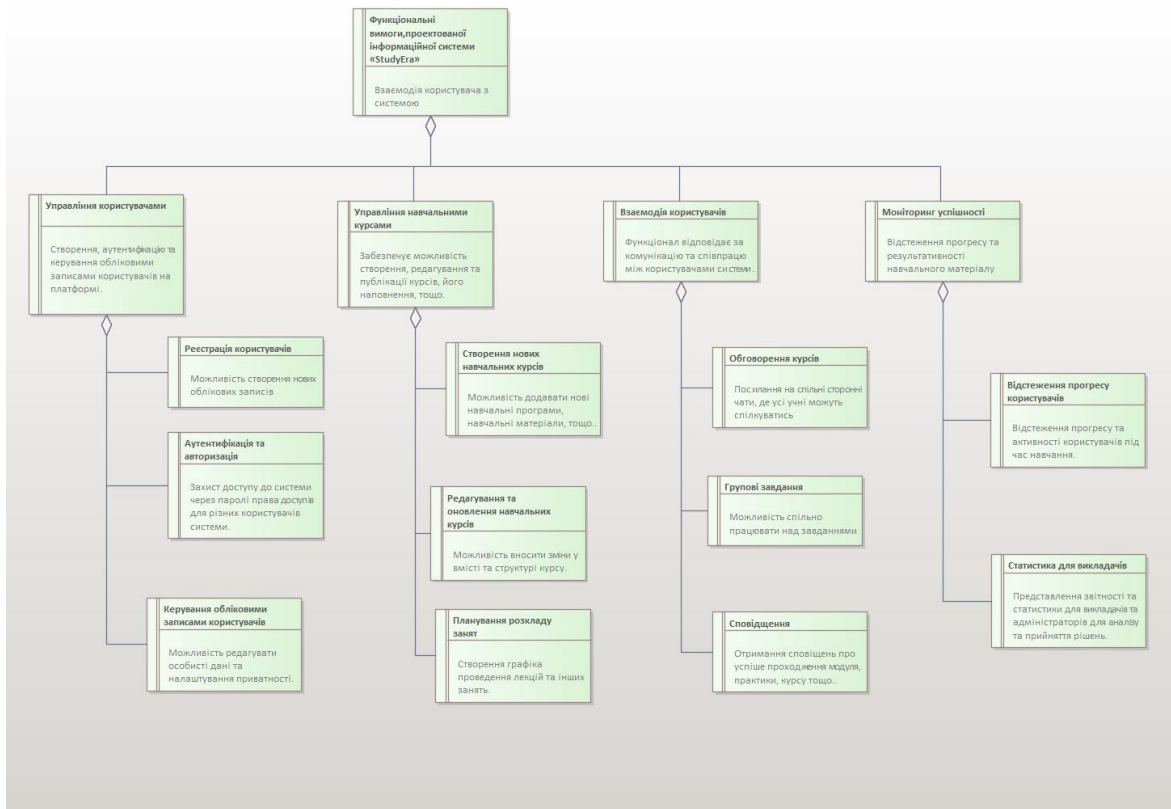


Рисунок 2.3 - Функціональні вимоги до розробленої інформаційної системи

2.1.3 Нефункціональні вимоги до системи

Забезпечення ефективності роботи інформаційної системи "StudyEra", також варто розглядати і нефункціональні вимоги. Ці вимоги орієнтовані на забезпечення стабільної та безперебійної роботи платформи, збереження конфіденційності користувачів та ефективне використання ресурсів. У елементах нефункціональних вимог було виділено наступні елементи:

1. Швидкість системи.

Платформа має працювати ефективно та швидко, навіть при великій кількості одночасних користувачів.

2. Масштабованість.

Система повинна бути легко масштабованою, щоб забезпечити зручну роботу при збільшенні обсягів користувачів та курсів.

3. Доступність.

Платформа повинна бути доступною для користувачів з будь-якого пристрою з Інтернет-підключенням, незалежно від їхнього місця розташування.

4. Відмовостійкість.

Система повинна мати механізми відновлення та резервне копіювання даних для запобігання втраті інформації у випадку виникнення непередбачених ситуацій

5. Інтерфейс користувача.

Інтерфейс та інші елементи системи мають бути зручними та інтуїтивно зрозумілими для користувачів будь-якого рівня навичок.

6. Мовна підтримка

Система повинна підтримувати різні мови для зручного користування користувачами з різних країн та культур. [Окрім мови агресора]

7. Сумісність із браузером.

Платформа повинна бути сумісною з різними веб-браузерами, такими як Google Chrome[5], Mozilla Firefox [6], Safari [7] та Microsoft Edge [8], а також для пристроїв на базі android чи iOS для забезпечення доступності для широкого кола користувачів.

Описані елементи вище, були представлені на діаграмі нефункціональних вимог до проектованої інформаційної системи на рис. 2.4

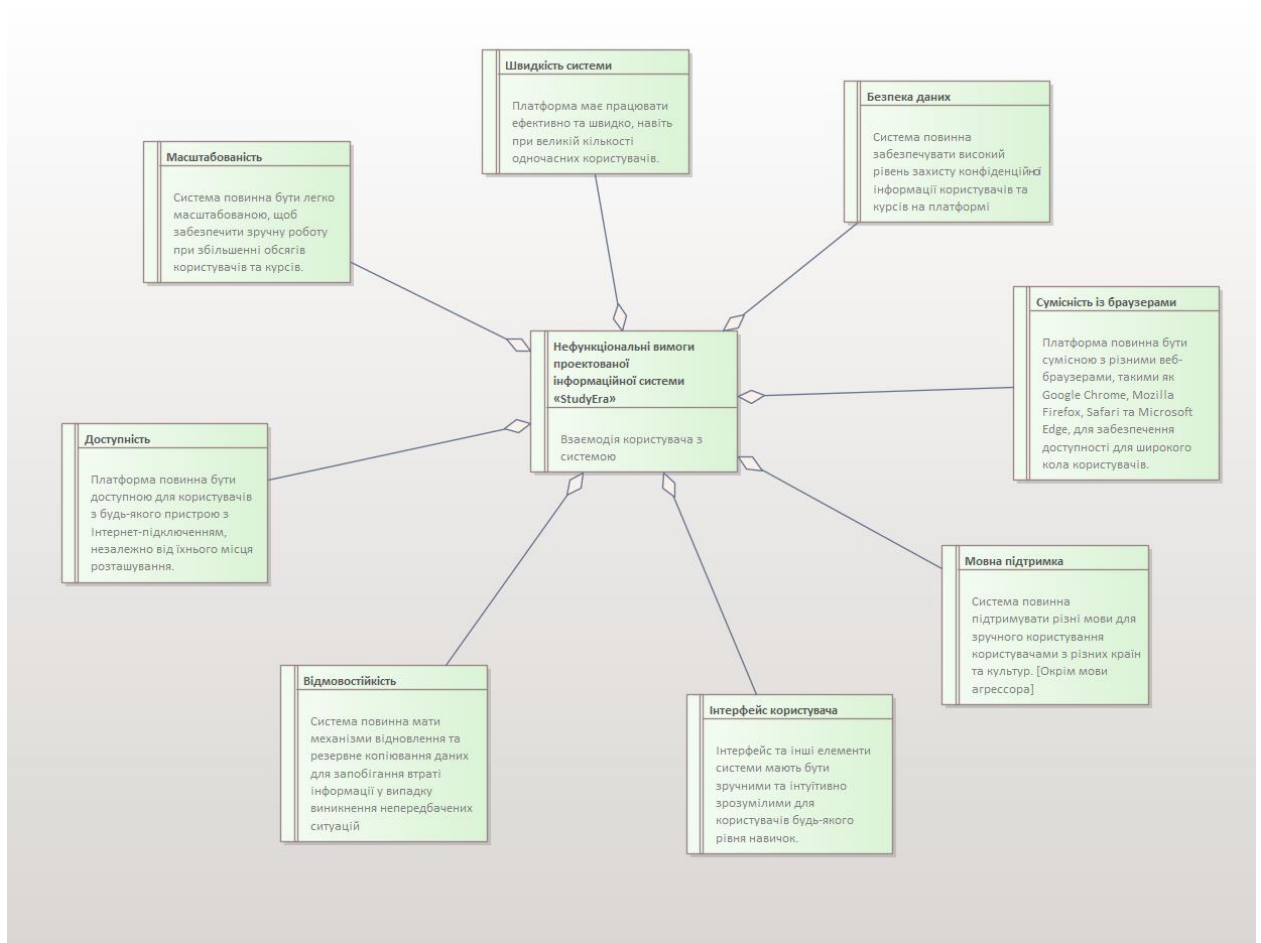


Рисунок 2.4 – Нефункціональні вимоги до розробленої інформаційної системи

2.2 Постановка задачі та алгоритм розв'язання задачі

У контексті автоматизації дистанційного навчально-освітнього центру розвитку», обрана методологія проєктування може бути порівняна із концепцією об'єктноорієнтованого програмування, де кожен тип користувача, функція та взаємодія представлені як окремі об'єкти.

1. Студент (Об'єкт Студент):

- Атрибути: ім'я, прізвище, дата народження, обраний навчальний курс, фотографія профілю.

- Методи: Доєднання до курсів, перегляд уроків, здача домашніх завдань, доєднання до зовнішніх додатків (чаті).

2. Вчитель (Об'єкт Вчитель):

- Атрибути: Ім'я, доступ до активності студентів закріпленої групи.
- Методи: Перевірка домашніх завдань та оцінювання, додавання нових навчальних матеріалів.

3. Адміністратор (Об'єкт Адміністратор):

- Атрибути: Права адміністратора, повний контроль платформи, включаючи внесення змін, додавання нових викладачів, учнів.
- Методи: Додавання нових освітніх програм, повний контроль та управління особистими обліковими записами вчителів та студентів.

Використання принципів ООП дозволяє структурувати систему, розглядаючи кожен тип користувача як окремий об'єкт і надаючи їм необхідні властивості та методи для ефективної взаємодії в навчальному середовищі. Такий підхід сприяє гнучкості, розширюваності самої системи. Задля демонстрації роботи ІС, було розроблено декілька діаграм.

На діаграмі (рис.2.5) розроблена система передбачає взаємодію трьох типів користувачів, а саме студентів, вчителів та головного адміністратора. Студент має змогу доєднатись на доступні курси на платформі, переглядати уроки, здавати домашні завдання. По завершенню проходження курсу — студент отримує змогу завантажити сертифікат якості про проходження курсу. Вчитель має змогу переглядати та оцінювати домашнє завдання студента. Адміністратор додає нову освітню програму, закріплює вчителів за конкретними програмами, змінює дані про студента та має

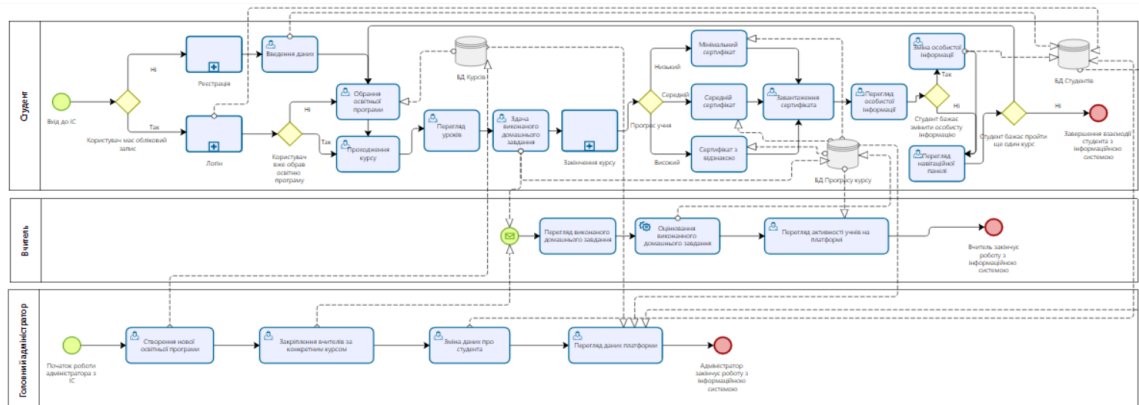


Рисунок 2.5 – Діаграма взаємодії

2.2.1 Характеристика задачі

Розроблена система призначена для оптимізації навчального процесу у дистанційному форматі для усіх осіб. Завданням якої є надання якісної безкоштовної освіти для її користувачів, з метою підготовки майбутніх спеціалістів та людей, які бажають перекваліфікуватись у сфері ІТ

З врахуванням швидкого розвитку сучасних технологій та зростання інтересу до дистанційної освіти, створення ефективної інформаційної системи для надання послуг цієї форми освіти стає актуальним завданням. У даному підрозділі розглядається комплекс заходів, спрямованих на вирішення ключових завдань в реалізації інформаційної системи, а також представлено інформаційної задачі (рис 2.6).

Розв'язування задач здійснюватиметься за допомогою об'єктів, таких як:

- Чітка структурна ієрархія користувачів, зі своїм унікальним функціоналом.
- База даних усіх користувачів системи. Включаючи окремі групи учнів, які закріплені за певною освітньою програмою та які підпорядковуються певному вчителю.
- Дані навчальних матеріалів котрі зберігаються у відповідній таблиці, надають змогу адміністратору інформаційної системи швидко реагувати на непередбачені помилки щодо надання доступів чи виникнення помилок.

Вихідна інформація яка надається користувачеві, залежить виключно від запиту. У випадку, якщо студент бажає доєднатись до курсу, натискаючи відповідний елемент, який передбачений системою, вихідною інформацією він отримує повний доступ до навчального матеріалу, така ж процедура проходить і з завантаження домашнього завдання та сертифікату по проходженню курсу – система буде сповіщати користувача про те, що операція була успішно виконана.

У інформаційній системі передбачена швидка відповідь на запит користувача. Також, у випадку взаємодії двох користувачів «Вчителя» та

«Студента», при моделюванні ситуації оцінення роботи користувачем «Вчитель» -> «Учень», кінцевий отримає результат з деякою затримкою, але не суттєвої у межах розробленої інформаційної системи. Така сама взаємодія, відбувається у зворотному напрямку коли «Учень» -> «Вчитель», коли перший користувач завантажує виконане завдання для другого користувача на оцінювання.

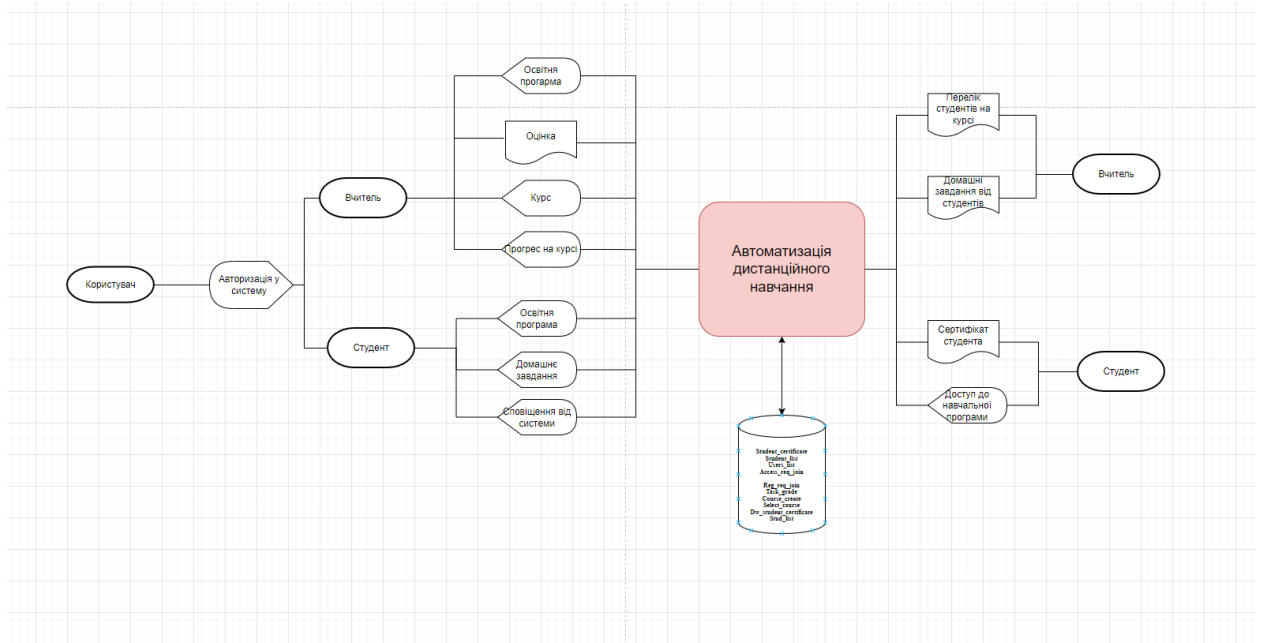


Рисунок 2.6 - Інформаційна модель задачі

2.2.2 Вихідна інформація

У проєктованій інформаційній системі вихідна інформація відіграє ключову роль взаємодією її користувачів. Ця інформація використовується для формування вихідних повідомлень, необхідних для подальшого використання. Було виділено декілька з них (табл. 2.2).

Таблиця 2.2 – Перелік і опис вихідних повідомлень

№	Назва вихідного повідомлення	Ідентифікатор в базі даних	Форма подання і вимоги до неї	Періодичність видання	Термін видання і допустимий час затримки	Користувач інформації
1.	Сертифікат студента	Student_certificate	Електронний документ/масив	Після завершення курсу	Після завершення курсу	Студент
2.	Список студентів на курсі	Student_list	Екранна форма	При запиті вчителя	За запитом	Вчитель
3.	Перелік усіх користувачів ІС	Users_list	Екранна форма/масив	При запиті адміністратора	За запитом	Адміністратор
4.	Надання доступу до навчального курсу	Access_req_join	Екранна форма	При запиті студента	За запитом	Студент

2.2.3. Вхідна інформація

Вхідна інформація як правило надається за запитом користувача. У нашому випадку, в інформаційній системі є можливість ініціювати запити, які своєю чергою звертаються до бази даних, для отримання певних результатів, щодо навчального процесу. Основні вихідні дані було виділено у (таб 2.3)

Таблиця 2.3 - Перелік і опис вхідних повідомлень

№ з/п	Назва вихідного повідомлення	Ідентифікатор в базі даних	Форма подання	Термін видання і частота надходження	Джерело
1.	Реєстрація студента на курс	Reg_req_join	Електронна форма	При реєстрації нового студента	Вебсайт
2.	Оцінка вчителя за домашнє завдання	Task_grade	Електронна форма/масив	Після завершення оцінювання	Вчитель
3.	Додавання нової освітньої програми	Course_create	Електронна форма	При створенні нової програми освітньої програми у адміністративній панелі	Адміністратор
4.	Обрати навчальний курс	Select_course	Електронна форма	За запитом	Учень
5.	Сертифікат студента	Dw_student_certificate	Електронний документ/масив	За запитом	Учень
6.	Переглянути перелік студентів	Stud_list	Електронний документ	За запитом	Вчитель

2.2.4. Математичне забезпечення та алгоритм функціонування системи

При розробці інформаційної системи, яка позиціонується як платформа для навчання, варто враховувати результативність та залученість студентів у навчальний процес для подальшого аналізу, та/або зміни стратегії подання

навчального матеріалу, для підвищення цієї залученості, спрощена схема алгоритму представлена нижче (рис.2.7). У проєктованій ІС «StudyEra», передбачено розрахунок середнього балового показника, а саме:

- Середній бал кожного студента по проходженню практики

1. Середній бал можна розрахувати, склавши всі оцінки, отримані студентом, і поділивши їх на кількість оцінок за формулою (2.1):

$$S = \frac{R}{T} \quad (2.1)$$

Де:

- S – середній бал студента;
- R – сума оцінок за усі виконані домашні завдання;
- T – кількість можливостей отримання оцінки в цілому.

Цей показник надалі впливатиме на тип сертифіката, який по завершенню освітньої програми може отримати учень.

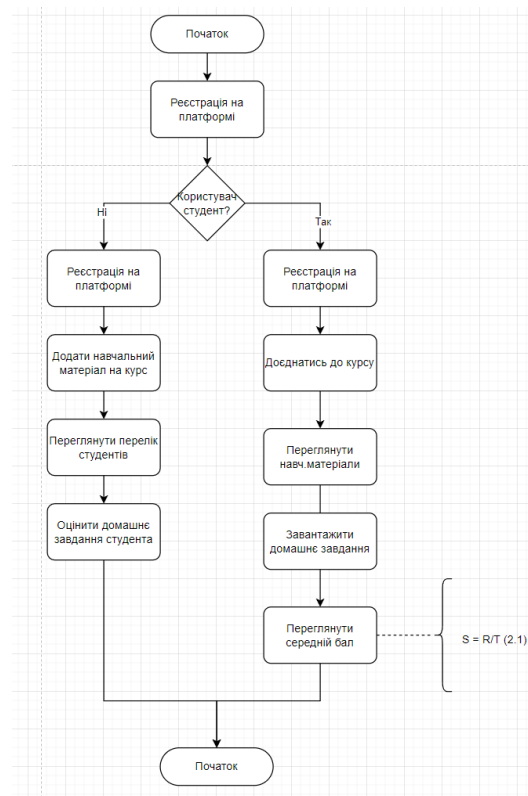


Рис. 2.7– Спрощена схема алгоритму

2.3 Моделювання інформаційної системи

При моделюванні системи, було виділено 3-х ключових осіб, та визначення можливостей системи за допомогою UML діаграми прецедентів на рис 2.8, що відображає взаємодію акторів з розробленою системою. Нижче, зафіксуємо поведінку розробленої інформаційної системи.

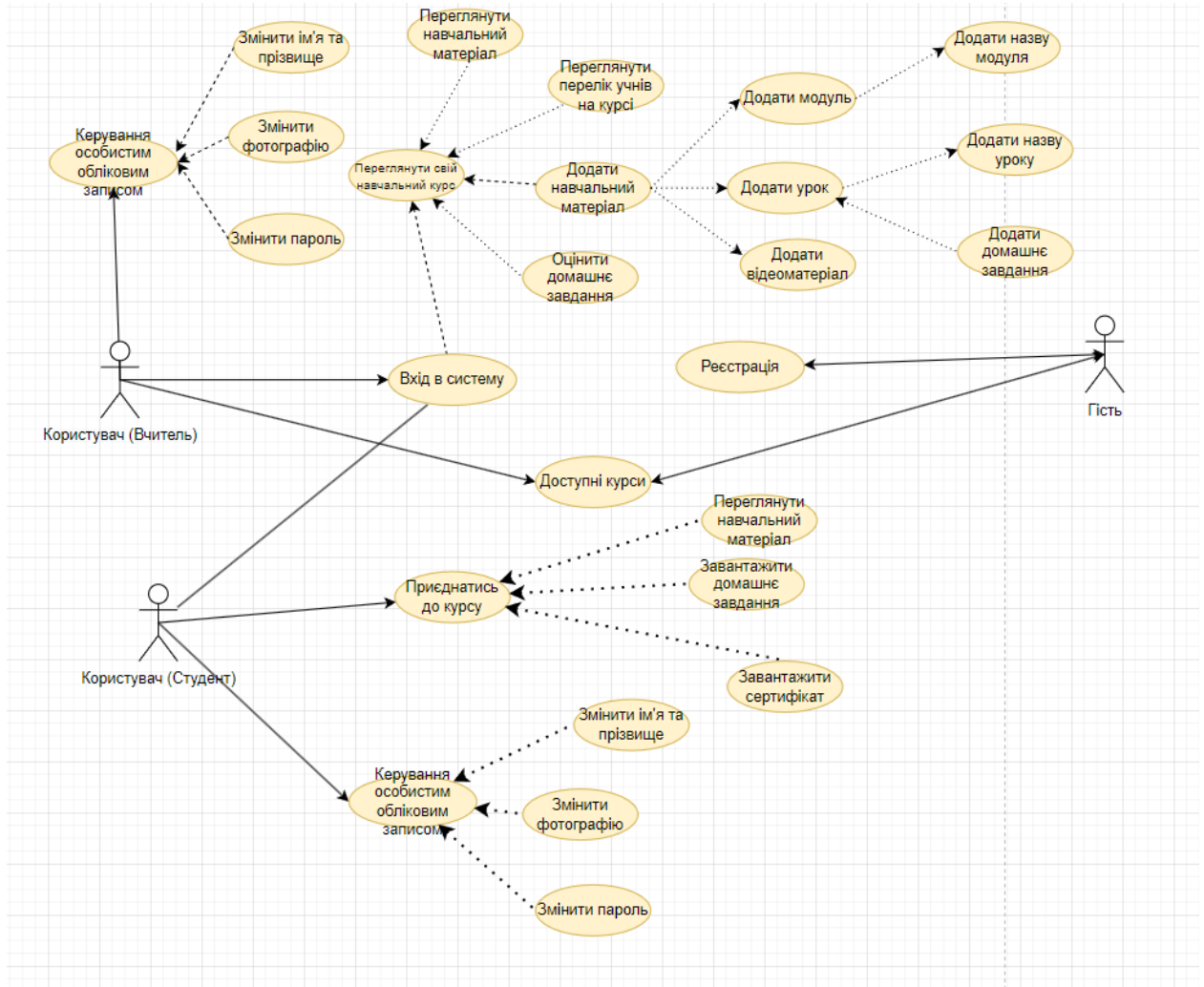


Рисунок 2.8 – Діаграма прецедентів з ключовими особами

Користувач з рівнем доступу – Вчитель.

Користувач (вчитель) має змогу увійти у свій обліковий запис та одразу працювати зі своїм навчальним курсом.

- Переглянути свій навчальний курс:

користувач (вчитель), може переглянути курси, за якими він закріплений. Це досить зручне рішення для навігації по інформаційній системі, так як кількість курсів на платформі буде збільшуватись.

- Переглянути навчальний матеріал:

користувач (вчитель) має змогу переглянути наявність відповідного матеріалу у програмі курсу та отримати розуміння про актуальність або наявність матеріалу.

- Переглянути перелік учнів на курсі:

користувач (вчитель) завжди має можливість переглянути кількість учні на його курсі.

- Додати навчальний матеріал:

1. Користувач (вчитель) завжди має можливість створити новий модуль на навчальній програмі, та задати для нього назву.

2. Користувач (вчитель) також може додавати уроки у новостворені модулі. Задавати їм назву, та закріплювати домашнє завдання для виконання.

3. Користувач (вчитель) у новоствореному уроці, який закріплюється за певним модулем, може додати відео матеріал з зовнішнього хмарного джерела, такого як YouTube [12].

- Переглядати та оцінювати домашні завдання студентів:

Користувач (вчитель) має змогу перевіряти виконання домашніх завдань студентами. До виконаного домашнього завдання вчитель може закріпити текстове повідомлення а також виставити оцінку по 5-ти бальній шкалі.

- Керування обліковим записом

Користувач (вчитель) має змогу змінювати фотографію свого особистої сторінки, змінювати пароль, а також дані про себе.

Користувач з рівнем доступу – учень/студент.

Користувач (учень) має змогу увійти у власний обліковий запис та одразу та одразу обрати бажаний навчальний курс.

- Приєднатись до навчального курсу:

користувач (студент) має можливість доєднуватись до усіх навчальних програм, які представлені на платформі.

- Перегляд навчального матеріалу:

користувач (студент), після доєднання на курс має можливість одразу ознайомлюватись та переглядати структуру подання модулів у обраному курсі.

- Завантажити домашнє завдання:

користувач (студент), одразу ж після доєднання має змогу розпочати виконання домашнього завдання, яке передбачено навчальним уроком.

- Завантажити сертифікат:

по завершенню навчального курсу, студент матиме можливість завантажити сертифікат, що свідчить про завершення обраного навчального курсу.

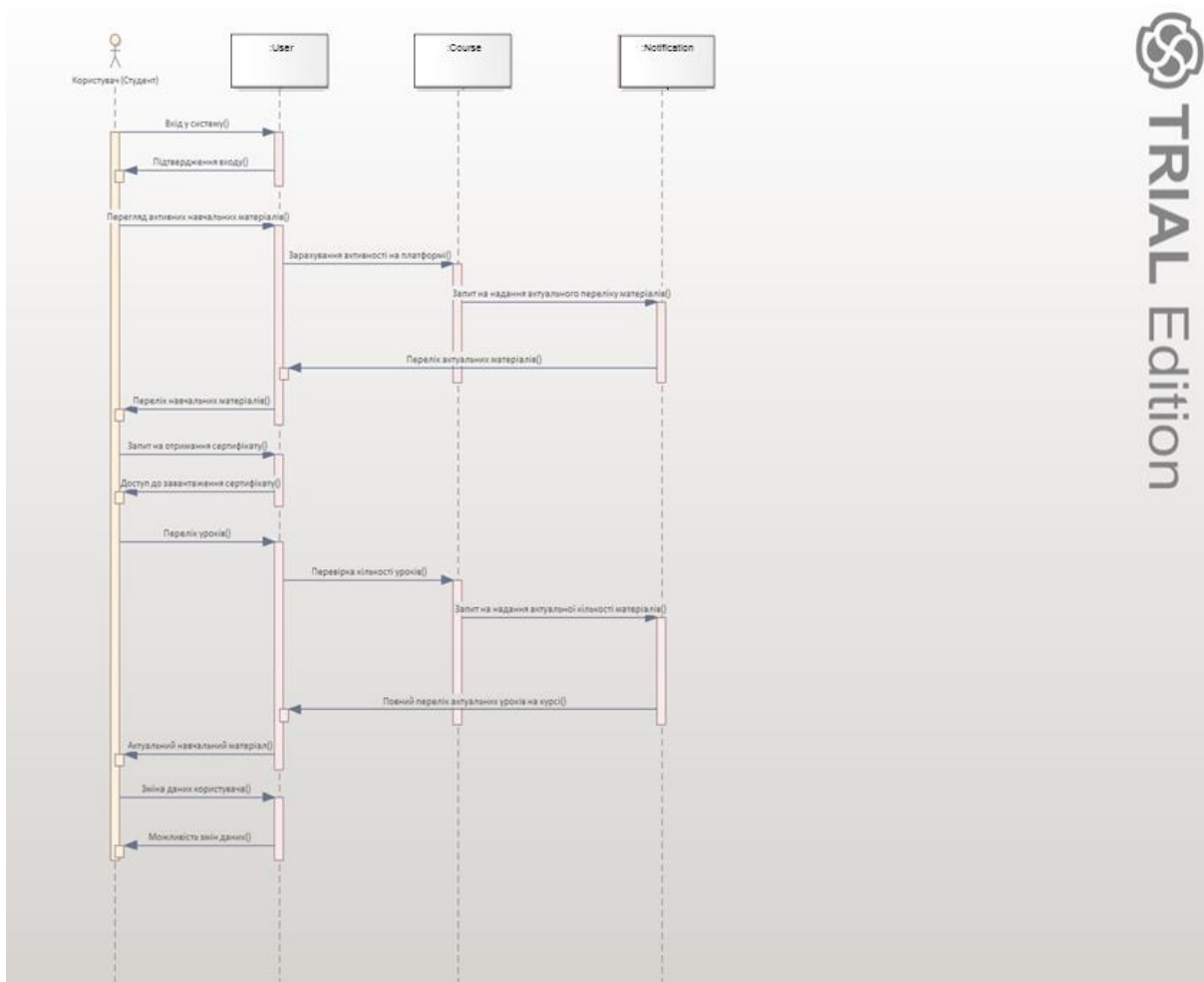


Рисунок 2.9 – Діаграма послідовності розробленої системи

На діаграмі послідовності (рис.2.9) зображено процес послідовності взаємодії користувача "студент" з інформаційною системою. Для початку роботи користувач повинен пройти етап входу до системи шляхом введення своїх облікових даних. Після успішної автентифікації, користувач отримує доступ до функціонала системи. Серед операцій, доступних для виконання, у користувача є можливість перегляду переліку активних навчальних програм, запити результатів навчання та подання запити на зміну власних даних, завантаження сертифіката (по завершенню навчання), можливість доєднання до зовнішніх додатків (чату у Telegram[9] та перегляду графіку лекційних матеріалів у Google Calendar[10])

На діаграмі (рис.2.10) зображено процес послідовності взаємодії користувача "Адміністратор" з інформаційною системою де продемонстровано процес створення нового навчального курсу та закріплення за створеним курсом вчителя.

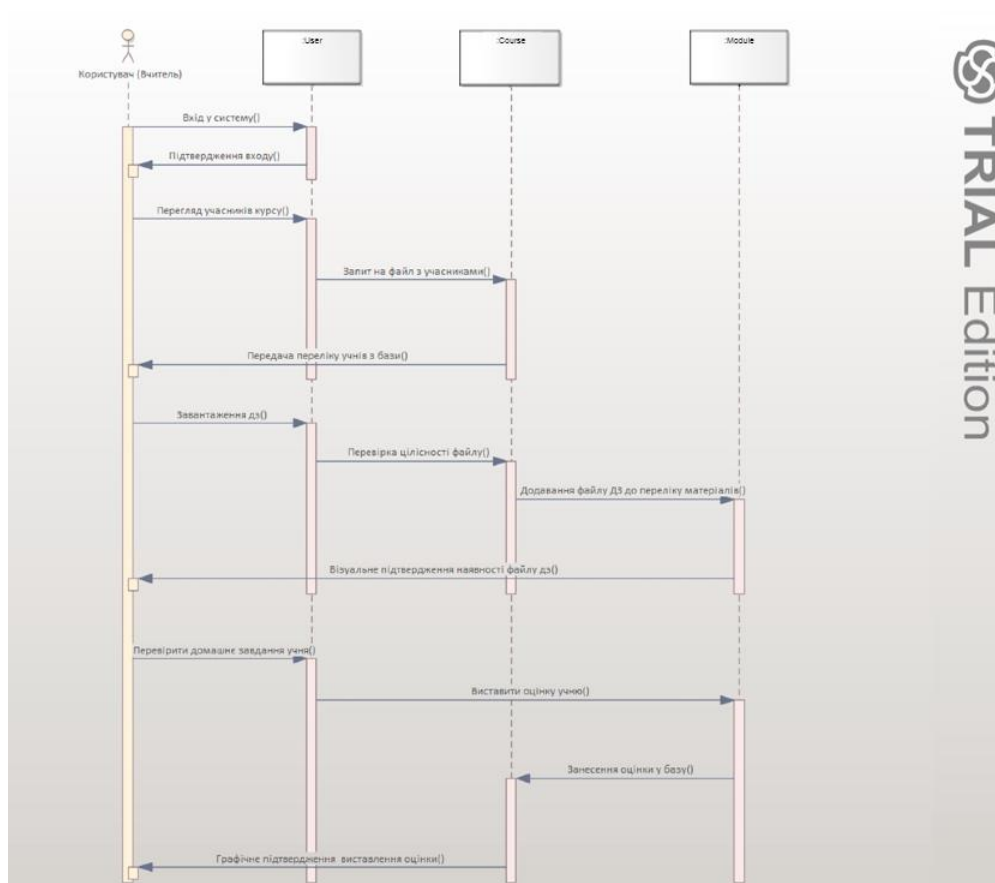


Рис. 2.10 – Діаграма послідовності розробленої системи сценарій адміністратора

На діаграмі (рис.2.11) зображено процес послідовності дій взаємодії користувача "Вчитель" з інформаційною системою. Продемонстровані ключові дії на платформі, які користувач з даним типом прав доступу має змогу виконувати. Варто зазначити, що не усі процеси представлені на даній діаграмі.

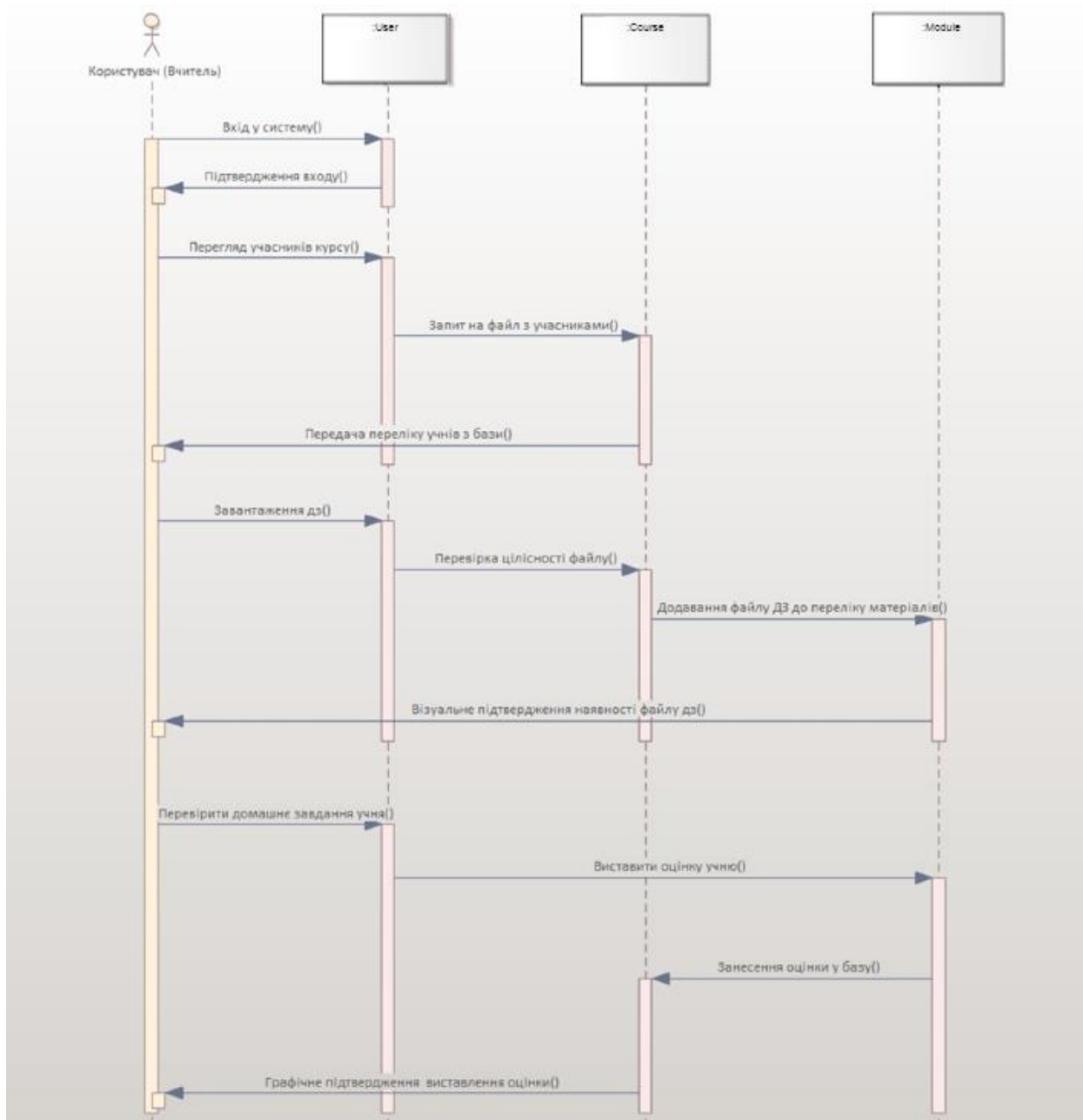


Рис. 2.11 – Діаграма послідовності розробленої системи сценарій користувача вчителя.

Для ефективного моделювання структури інформаційної системи, також варто враховувати і шляхи взаємодії користувача з системою. На діаграмі внутрішніх блоків (рис. 2.12), описаний шлях користувача інформаційної системи для отримання доступу до навчального курсу. Для початку, кожен користувач, має пройти валідацію даних, за допомогою логіну або реєстрації у систему, після чого, з переліку доступних навчальних курсів, має змогу обрати необхідний.

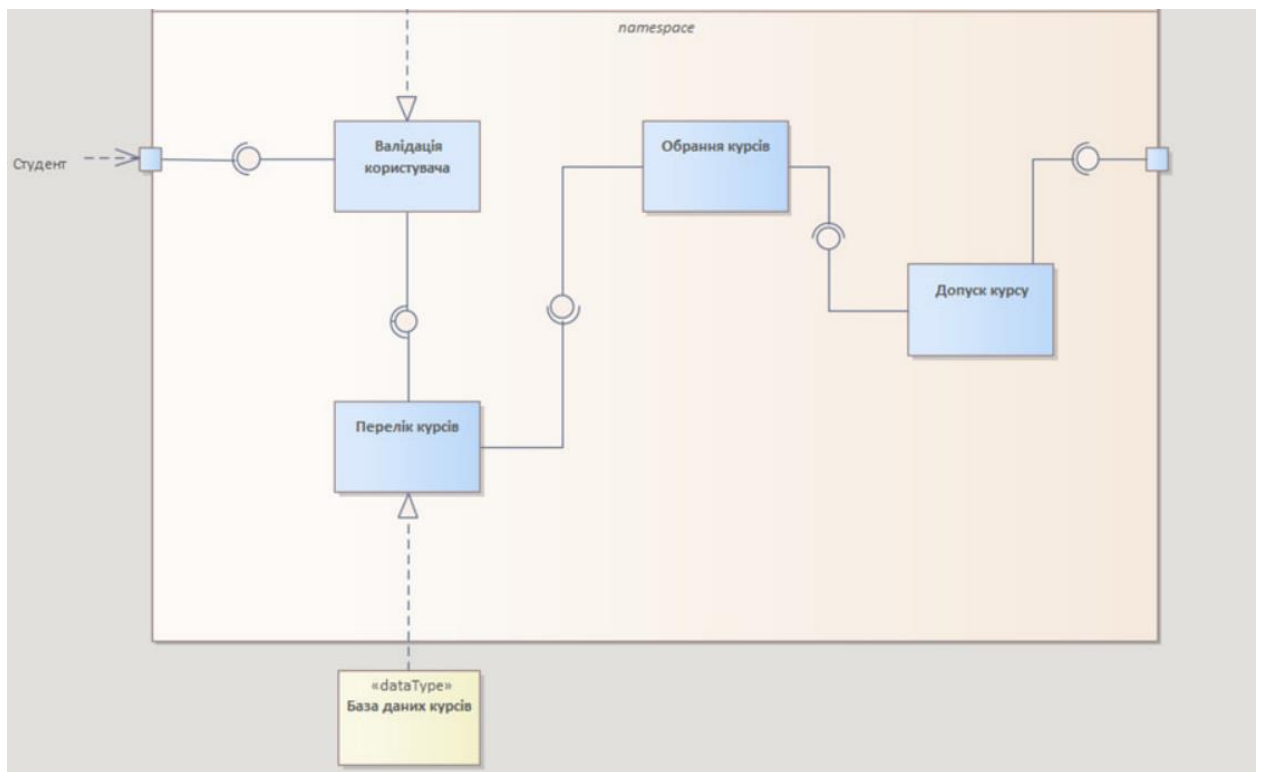


Рисунок 2.12 – Діаграма внутрішніх блоків проєктованої системи

РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1 Інформаційне забезпечення

Під час реалізації кваліфікаційного бакалаврського проекту, для створення бази даних системи, було обрано систему управління базами даних MsSQLServer [19]. Обране середовище забезпечує високу продуктивність, надійність і масштабованість, що є критично важливим для інформаційної системи навчально-освітнього центру розвитку.

Використання MsSQL[20] дозволяє ефективно керувати великим обсягом даних, забезпечуючи швидкий доступ до навчальних матеріалів та стабільну роботу системи навіть при значному навантаженні.

Для доступу та управління даними в MsSQL[20] було використано Entity Framework Core (EfCore)[21]. EfCore є сучасним ORM (Object-Relational Mapping) фреймворком для .NET, який дозволяє розробникам працювати з базою даних за допомогою об'єктно-орієнтованого підходу.

Основною перевагою обраного підходу є висока продуктивність, надійність та масштабованість MsSQL, це означатиме, що система буде працювати стабільно навіть при значних навантаженнях. Використання EfCore[19] забезпечує зручний підхід до управління даними, спрощуючи розробку та підтримку коду за рахунок зменшення необхідних SQL-запитів та прозорого відображення об'єктів на базу даних. Завдяки розробленому механізму аутентифікації та авторизації користувачів, досягається безпека даних усіх користувачів системи. Крім того, система передбачає регулярне резервне копіювання бази даних, що дозволяє відновити її у випадку збоїв чи втрати даних. Зручне адміністрування бази даних, забезпечене комбінацією MsSQL[19] та EfCore[20], що значно спрощує управління системою та підтримку її роботи на високому рівні. Побудовану діаграму класів представлено на рис 3.1, інфологічну модель БД наведено у (додату Б).

Описи структур масивів подані у вигляді таблиць (3.1 – 3.9).

Таблиця 3.1 – Масив «Користувачі системи»

Найменування	Ідентифікатор у програмі	Умовне позначення формули	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний	Умова на значення	Обов'язкове поле	Індексне поле	
Id	Id	ID	int	PK	-	Так	ІНД	Сертифікат судедн та
Ім'я	FirstName	FN	nvarchar	-	-	Так	-	
Прізвище	LastName	LN	nvarchar	-	-	Ні	-	
Електронна пошта	Email	ME	nvarchar	-	Довжина від до 0-200	Так	-	
Захист паролю	PasswordHash	HS	nvarchar	-	-	Так		
Роль	RoleID	RI	int	Fk	-	Так	ІДД	Ролі системи
Дата народження	Birthday	BD	datetime 2	-	-	Так	-	
Рівень освіти	EducatoionalStatus	ES	nvarchar	-	-	Ні	-	
Аватар користувача	AvatarUrl	AU	nvarchar	-	-	Ні	-	

Таблиця 3.2 – Масив «Ролі» у системі

Найменування	Ідентифікатор у програмі	Умовне позначення формули	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний	Умова на значення	Обов'язкове поле	Індексне поле	
id	Id	ID	int	PK	-	Так	ІНД	Користувачі системи
Назва	Name	NM	nvarchar	-	-	Так	-	

Таблиця 3.3 – Масив «Уроки»

Найменування	Ідентифікатор у програмі	Умовне позначення формули	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний	Умова на значення	Обов'язкове поле	Індексне поле	
id	Id	ID	int	PK	-	Так	ІНД	
Назва	Title	TL	nvarchar	-	-	Так	-	
Опис	Description	Ds	nvarchar	-	Довжина від до 0-200	Так	-	
Опис домашнього завдання	HomeworkDescription	HD	nvarchar	-	-	Ні	-	Домашнє завдання
Наявність домашнього в уроці	HaveHomework	HW	bit	-	-	Так	-	
id модуля	ModuleId	MI	int	FK	-	Так	ІДД	Навчальні ні курси Навчальні модулі
Посилання на відео	YoutubeVideoId	YT	int	-	Довжина від до 0-200	Ні	-	

Таблиця 3.4 – Масив «Навчальні модулі»

Найменування	Ідентифікатор у програмі	Умовне позначення формули	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний	Умова на значення	Обов'язкове поле	Індексне поле	
id	Id	ID	int	PK	-	Так	ІНД	
Назва	Title	TL	nvarchar	-	-	Так	-	
Опис	Description	Ds	nvarchar	-	Довжи на від до 0-200	Так	-	
id курсу	CourseId	CI	int	FK	-	Так	ІДД	Уроки, навчальні курси

Таблиця 3.5 – Масив «Переглянуті уроки»

Найменування	Ідентифікатор у програмі	Умовне позначення формули	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний	Умова на значення	Обов'язкове поле	Індексне поле	
id	Id	ID	int	PK	-	Так	ІНД	
id уроку	LessonID	LI	int	FK	-	Так	ІДД	Серфікат для студента, домашнє завдання
id студента	StudentID	SI	int	FK	-	Так	ІДД	

Таблиця 3.6 – Масив «Навчальні курси»

Найменування	Ідентифікатор у програмі	Умове позначення формули	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний	Умова на значення	Обов'язкове поле	Індексне поле	
id	Id	ID	int	PK	-	Так	ІНД	
Назва	Title	TL	nvarchar	-	-	Так	-	
Опис	Description	Ds	nvarchar	-	Довжина від до 0-200	Так	-	
Картинка курсу	ImageUrl	ImU	nvarchar	-	Довжина від до 0-200	Так	-	
Максимально кількість студентів	LimitOfStudents	LOS	int	FK	-	Так	-	
id вчителя	TeacherID	TI	int	FK	-	Так	ІДД	

Таблиця 3.7 – Масив «Серфікат для студента»

Найменування	Ідентифікатор у програмі	Умове позначення формули	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний	Умова на значення	Обов'язкове поле	Індексне поле	
id	Id	ID	int	PK	-	Так	ІНД	
id курсу	CourseId	CI	int	FK	-	Так	ІДД	Навчальні модулі
id студента	StudentID	SI	int	FK	-	Так	ІДД	Переглянути уроки, Домашнє завдання

Таблиця 3.8 – Масив «Домашнє завдання»

Найменування	Ідентифікатор у програмі	Умове позначення формули	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний	Умова на значення	Обов'язкове поле	Індексне поле	
id	Id	ID	int	PK	-	Так	ІНД	
Маркування виконання	Completed	CM	bit	-	-	Так	-	
Оцінка	Mark	MD	int	-	-	Ні	-	
Коментар вчителя	CommentFromTeacher	CFT	nvarchar	-	Довжина від до 0-200	Ні	-	
id уроку	LessonID	LI	int	FK	-	Так	ІДД	Уроки, Переглянуті уроки
id студента	StudentID	SI	int	FK	-	Так	ІДД	Користувачі системи, Серфікату для студента
Відповідь студента	Answer	AN	nvarchar	-	-	Так	-	

Таблиця 3.9 – Масив «Сповіщення»

Найменування	Ідентифікатор у програмі	Умове позначення формули	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний/вторинний	Умова на значення	Обов'язкове поле	Індексне поле	
1	2	3	4	5	6	7	8	9

Продовження Таблиці 3.9

1	2	3	4	5	6	7	8	9
id	Id	ID	int	PK	-	Так	ІНД	
Назва	Title	TL	nvarchar	-	-	Так	-	
Опис	Description	Ds	nvarchar	-	Довжина від до 0-200	Так	-	
Тип	Type	TP	nvarchar	-	-	Так	-	
id користувача	UserId	MI	int	FK	-	Так	ІДД	Користувачі системи
id Сповіщення	NotificationTypeId	NTN	int	FK	-	Так	ІДД	

Інтерфейс користувача (студент):

- На головній сторінці ІС, студент має змогу обрати будь-яку навчальну програму та доєднатись до неї.
- У функціональній панелі користувача, передбачені пункти:
 - Сертифікати: студент матиме змогу переглядати отримані сертифікати, завантажувати їх.
 - Сповіщення: у панелі, студент отримує системні сповіщення, такі як – отримання оцінки за практичне завдання.
 - Налаштування: студенти матимуть змогу у налаштуваннях, здійснювати повну зміну інформації про себе, але виключно один раз.
 - Вихід: вихід з облікового запису.
 - Можливість переглядати усі модулі та уроки на курсі, до якого було здійснено доєднання.

- Можливість відправляти виконані домашні завдання на перевірку вчителю та отримати зворотній зв'язок від нього у вигляді оцінки по 5-ти бальній шкалі та коментаря.
- У навігаційній панелі, користувач матиме змогу переглядати перелік навчальних курсів, до яких він вже доєднався.

Інтерфейс користувача (вчитель):

- Можливість додавання до свого навчального курсу (який попередньо було створено адміністрацією) навчальних уроків/модулів, та закріплення за ними домашніх завдань.
- Доступний перегляд усіх учнів, які були зареєстровані на навчальний модуль, за яким був закріплений саме цей викладач.
- Користувач має змогу переглядати та оцінювати роботи, які були надіслані користувачем (студент) у відповідь на відповідне поле домашнього завдання.

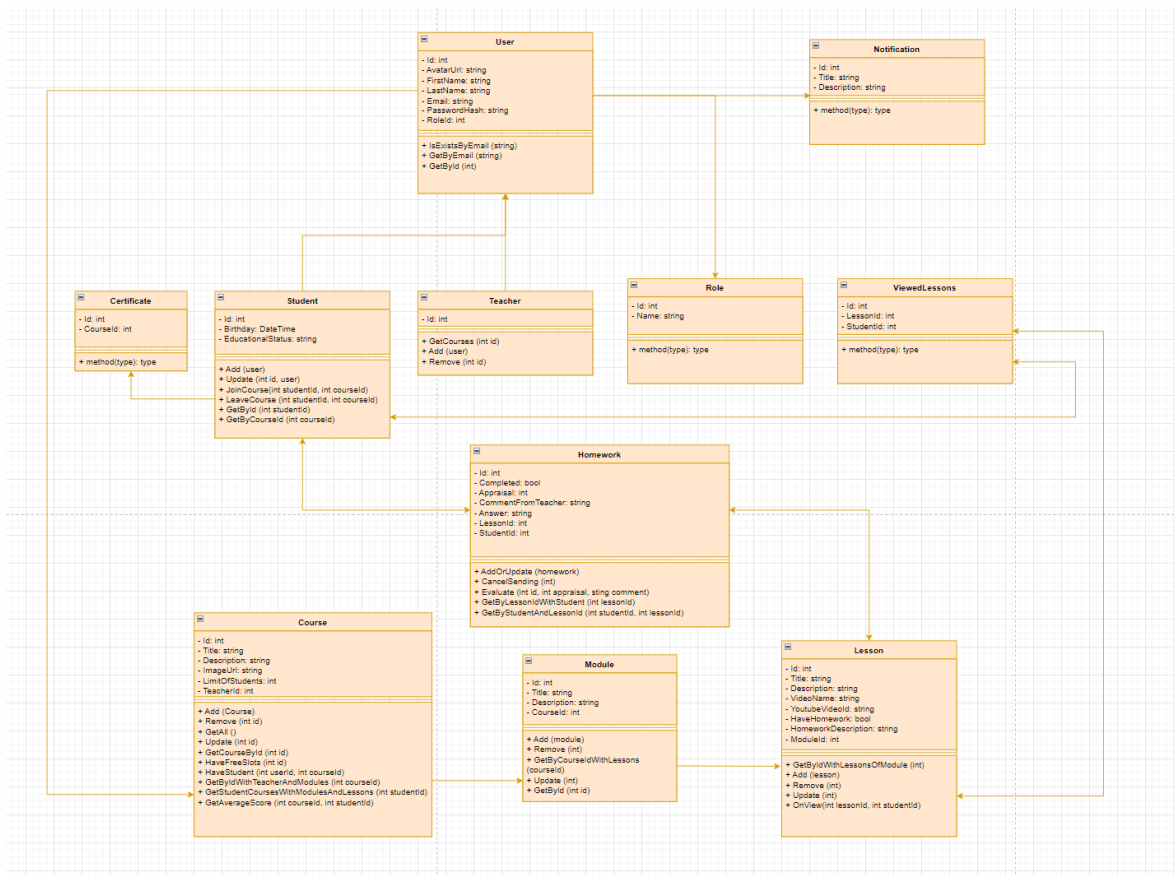


Рис. 3.1 – Діаграма класів проєктованої інформаційної системи

3.2 Технічне забезпечення

Для даної інформаційної системи було обрано клієнт-серверну архітектуру. Такий тип архітектури являє собою в розподіл завдань між клієнтською частиною, яка забезпечує інтерфейс для користувача, та серверною частиною, яка відповідає за зберігання та обробку даних (рис.3.2). Клієнт-серверна архітектура була вибрана через її численні переваги. Вона дозволяє легко масштабувати систему, додаючи нові сервери для обробки запитів або зберігання даних. Це забезпечує високу продуктивність і надійність системи навіть при значних навантаженнях. Також централізоване зберігання даних спрощує управління та підтримку системи, забезпечуючи високий рівень безпеки і конфіденційності. Основним інструментом для управління базами даних було обрано MsSQLServer. Це потужна система управління базами даних, яка забезпечує високу продуктивність, надійність та масштабованість. Використання MsSQLServer дозволяє ефективно керувати великим обсягом даних, забезпечуючи швидкий доступ до навчальних матеріалів та стабільну роботу системи навіть при значному навантаженні.

Таким чином, клієнт-серверна архітектура розподіляє завдання між клієнтами і серверами, забезпечуючи ефективну обробку даних і взаємодію користувачів з системою.

З недоліків такого підходу до реалізації системи є також необхідність у високоякісному технічному забезпеченні на серверній стороні, через високу кількість клієнтських запитів може призвести до значного навантаження на мережу і сервер. У разі відмови сервера, уся система може стати недоступною.

Також, варто зазначити і позитивні сторони такого підходу, вона полягає у легкості масштабованості системи, додавши більше серверів для обробки запитів або для зберігання даних. Також, централізоване зберігання та обробка даних спрощує управління та підтримку системи.

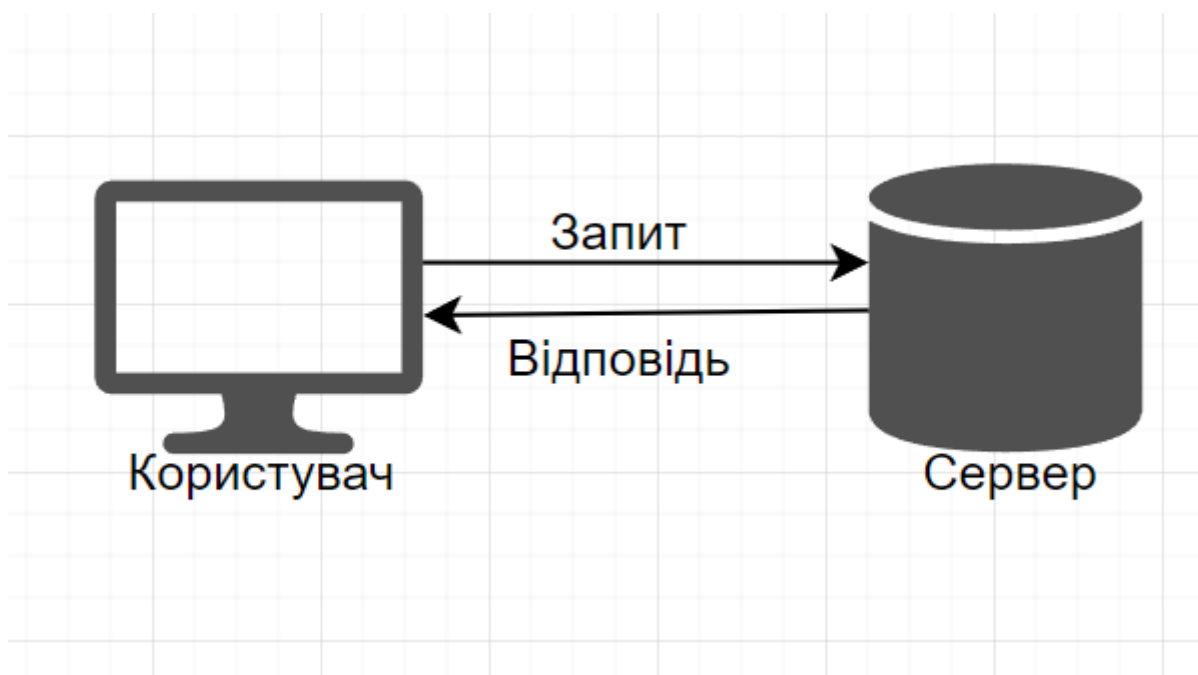


Рис. 3.2 – Спрощене представлення принципу роботи клієнт-серверної архітектура

3.3 Програмне забезпечення

Діаграма вказує на взаємозв'язок компонентів системи, який ілюструє процес, яким студент отримує доступ до конкретного курсу (рис 3.3).

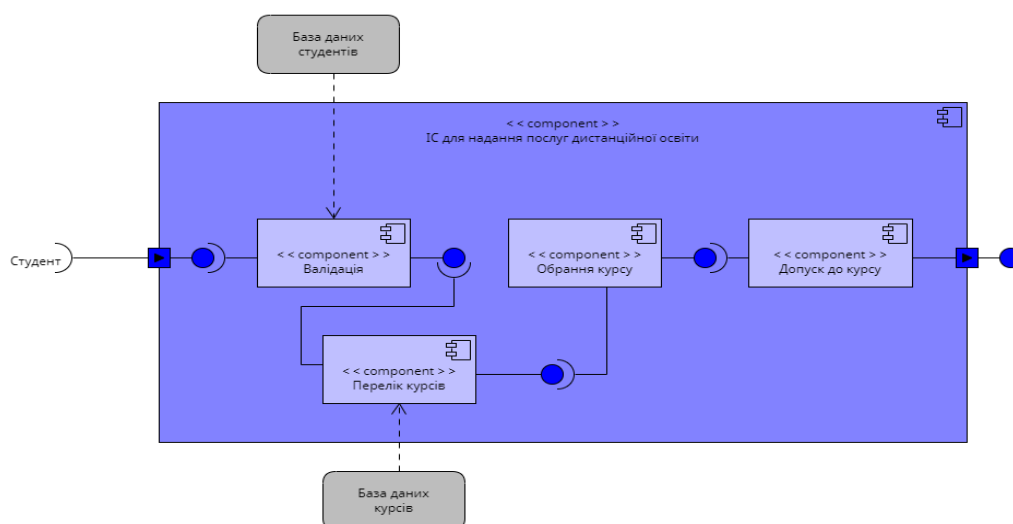


Рисунок 3.3 - Діаграма компонентів

При розробці інформаційної системи навчального-освітнього центру розвитку, було використано наступні програмні засоби: уся робота по моделюванню та створенню інформаційної системи проводилась на базі операційної системи Windows 10, на базі якої і працюватиме розроблена

інформаційна система. Програмним забезпеченням для роботи з базами даних було використано MSSqlServer[19] який дозволяє ефективно керувати даними та таблицями у базі даних розробленої системи.

Для написання Frontend частини, було використано середовище (Visual Studio Code 2022) та широкий спектр бібліотек і фреймворків, які забезпечують ефективну, динамічну взаємодію з користувачем. Нижче наведено основні технології та їхні переваги, які були застосовані:

@headlessui/react - Ця бібліотека забезпечує створення компонентів інтерфейсу користувача, які не залежать від стилів, що дозволяє максимально адаптувати їх під індивідуальні потреби проєкту.

Embla Carousel - надає легкий, гнучкий і продуктивний спосіб створення каруселей зображень або контенту.

Tailwind CSS - це утилітарний фреймворк для CSS, який дозволяє створювати індивідуальні стилі без необхідності написання кастомного CSS.

@react-input/mas - завдяки цьому, форми стають більш зручними та інтуїтивно зрозумілими для користувачів.

Redux, React-Redux, @reduxjs/toolkit - ці бібліотеки використовуються для ефективного управління станом у React-додатках.

React-YouTube - ця бібліотека дозволяє легко інтегрувати відео з YouTube у React-додатки.

Фреймворки які були використані у реалізації, такі як:

1. Next.js - це потужним фреймворком для React, який забезпечує серверний рендеринг та генерацію статичних сайтів. Його можливості, такі як оптимізація продуктивності, легка навігація між сторінками та автоматична генерація сторінок, значно покращують загальну продуктивність і SEO.
2. ASP.NET Core - цей фреймворк підтримує сучасні веб-технології і забезпечує масштабованість та інтеграцію з різними сервісами.

Для написання Backend частини, було використано середовище розробки (Visual Studio 2022) з підтримкою широкого спектру інструментів і бібліотек, таких як:

1. AutoMapper – бібліотека котра дозволяє полегшує взаємодію з класами та їх перетворенням
2. FluentValidation – бібліотека використовувалась для перевірки вхідних даних на коректність
3. Swagger – ця бібліотека використовувалась для генерації кінцевих точок у системи. А також, для генерації та взаємодії з панеллю адміністратора у розробленій інформаційній системі.

3.4 Результати реалізації інформаційної системи

Інформаційна система, яка була розроблена, представлена на знімках екрану (рис 3.4 – 3.24). Кожен зареєстрований користувач, котрий має доступ до мережі інтернет, має змогу входити у свій обліковий запис, ввівши свій «Логін» та «Пароль», які вказувались при реєстрації.

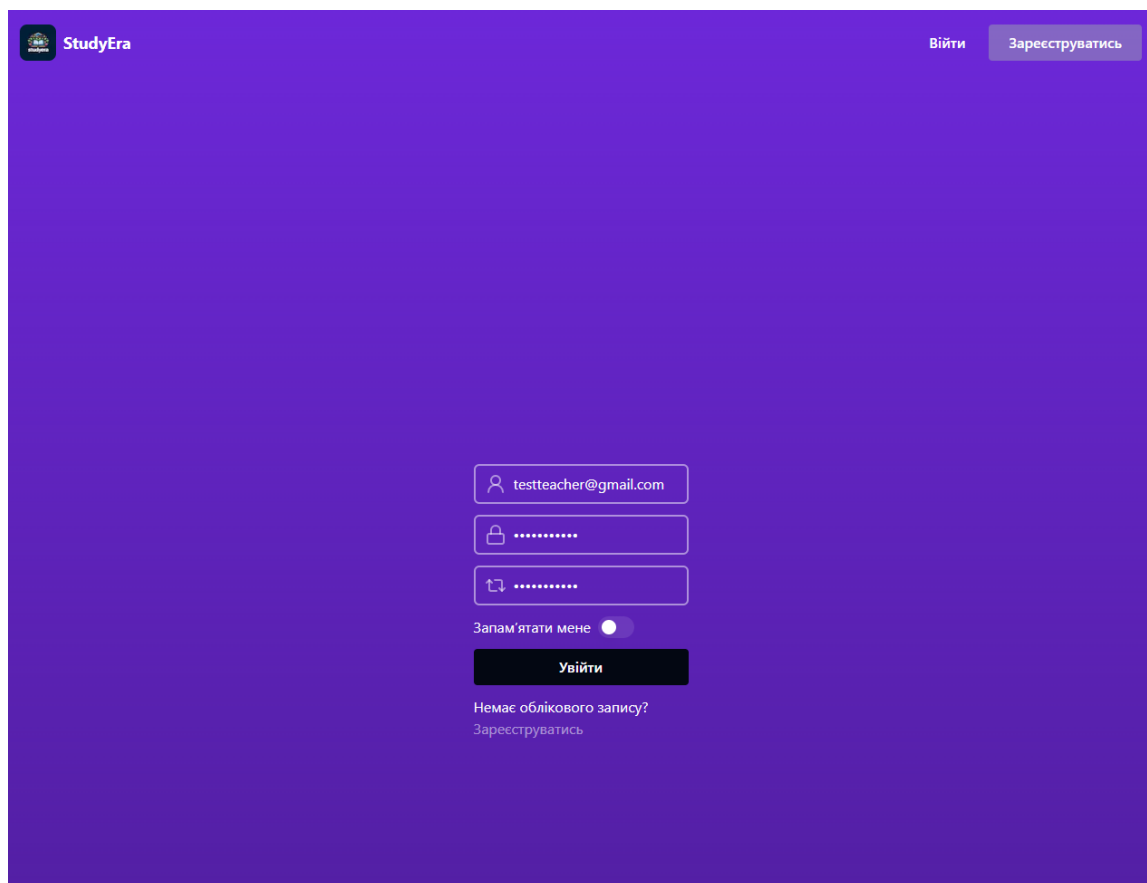


Рис. 3.4 – Вікно входу у обліковий запис користувача інформаційної системи.

У випадку, якщо у користувача ще не має облікового запису на платформі, він з легкістю може зареєструватись та розпочати навчання. На рис 3.5-3.6 продемонстровані навігаційні кнопки, які приведуть користувача до

форми реєстрації. На рис. 3.17 проілюстроване поле реєстрації, яке потрібно заповнити для того, щоб створити обліковий запис.

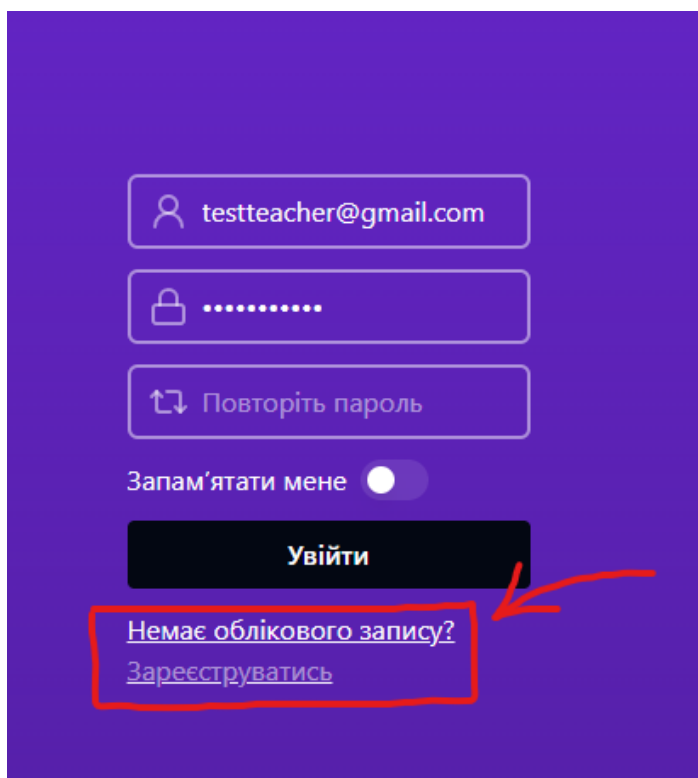


Рис. 3.5 – Навігаційна кнопка для переходу у форму реєстрації на платформі

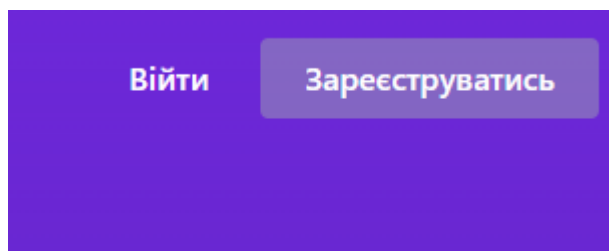


Рис. 3.6 - Навігаційна кнопка «Зареєструватись» для переходу у форму реєстрації на платформі

StudyEra Вийти [Зареєструватись](#)

Ім'я

Прізвище

Почта

Дата народження

Освітній статус

Пароль

Зареєструватись

[Вже є акаунт? Вхід](#)

Рис. 3.7 – Форма реєстрації користувача у систему з відповідними полями для заповнення

Авторизувавшись у систему усі користувачі системи (вчителі та студенти), мають змогу переглядати доступ курс для доєднання, а також, сміливо до них доєднуватись. На рис.3.8 продемонстровані перші 2 тестові курси, до яких вже є можливість доєднання.

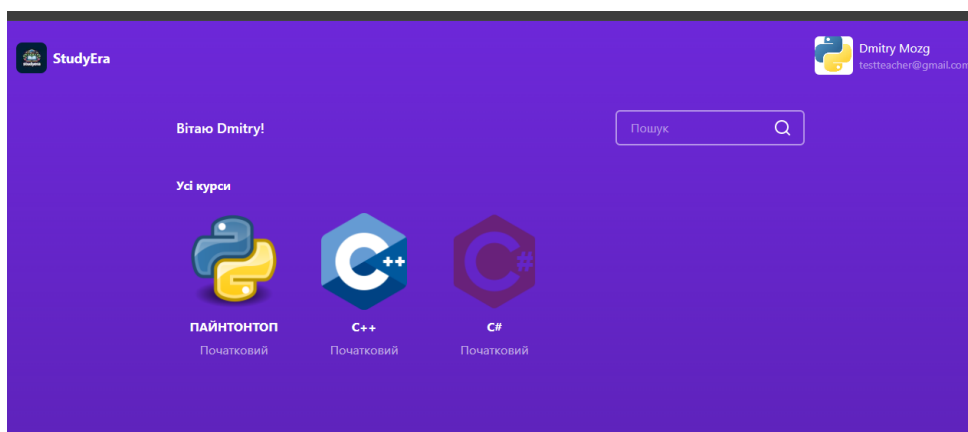


Рис. 3.8 – Перелік навчальних курсів до яких можна доєднатись

У кожного користувача системи є можливість переглядати інформацію свого облікового запису, шляхом натискання на свою фотографію у правому верхньому куті додатку (рис.3.9).

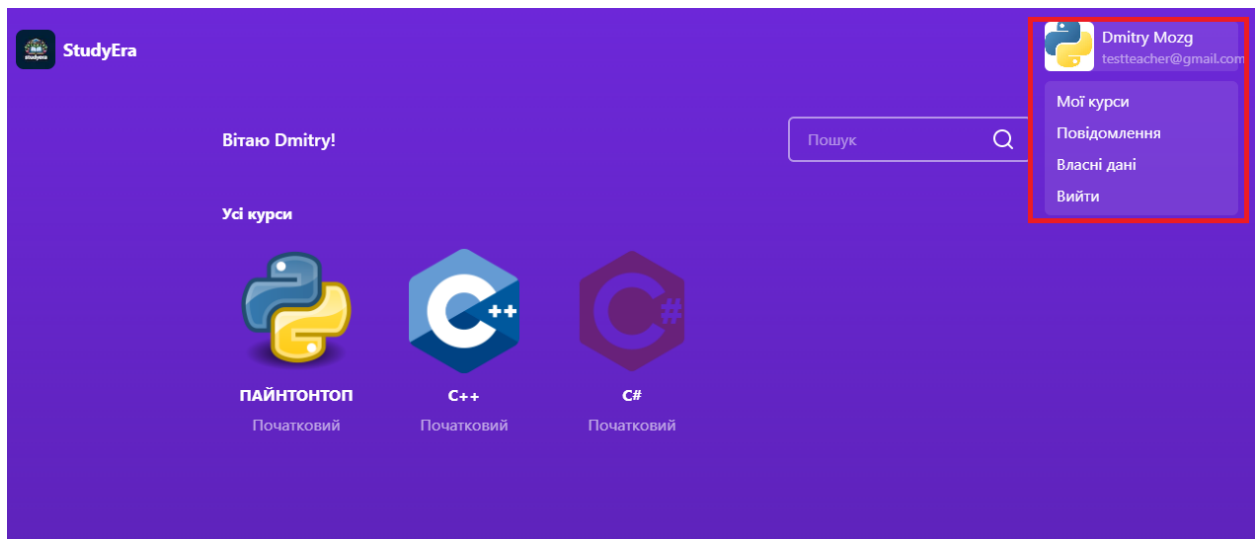


Рис. 3.9 – Пане керування обліковим записом

При натисканні на параметр «Мої курси» (рис.3.10), користувач вчитель має змогу переглянути навчальні курси, за якими його було закріплено адміністрацією. У нього є змога повністю переглядати інформацію щодо навчального курсу та здійснювати такі операції як:

- Переглянути сторінку курсу на панелі редактора (3.11)
- Переглянути кількість студентів на навчальному курсі (рис.3.12 – 3.13)
- Додати на обраний навчальний курс матеріал для студент (3.14 – 3.17)
- Закріплювати домашнє завдання за ними (3.18 - 3.20)
- Переглядати виконані студентами домашні завдання (3.21 - 3.22)
- Повністю редагувати сторінку курс: змінювати логотип, змінювати назву курсу та опис (3.23)
- Закріплювати посилання на спільний чат зі студентами, який реалізований у іншому додатку. (3.24)

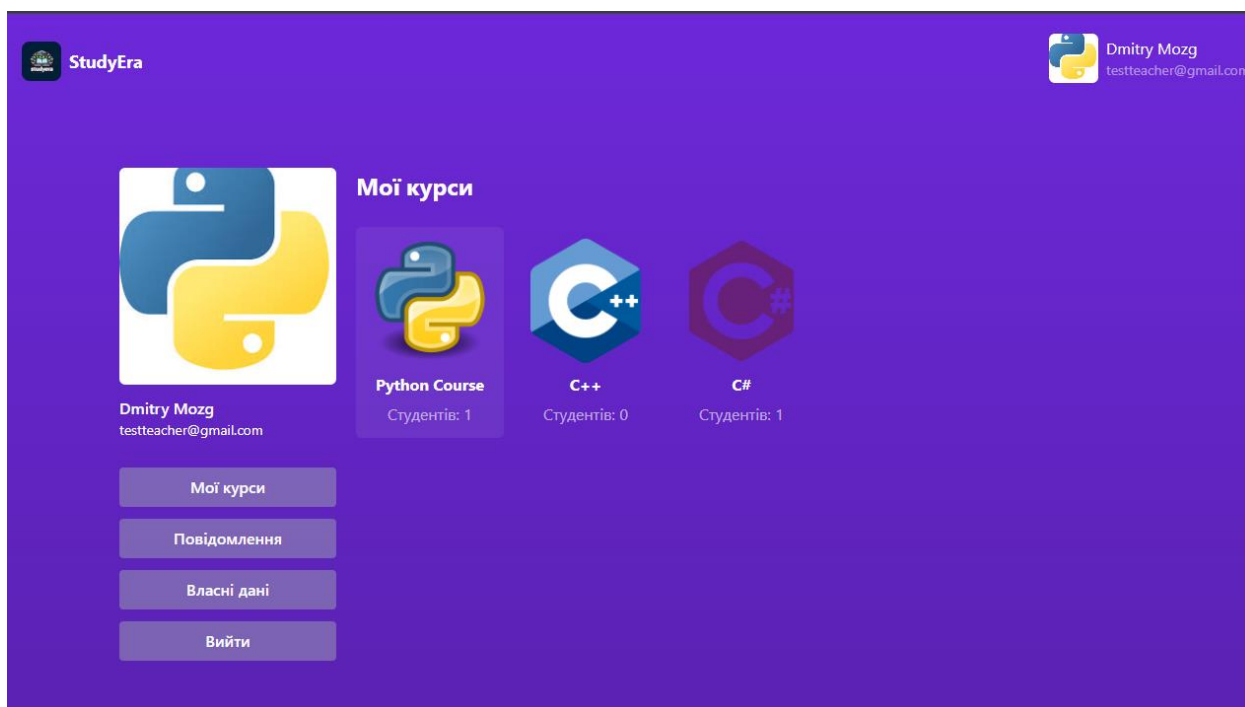


Рис. 3.10– вікно керування курсами, за якими закріплений вчитель

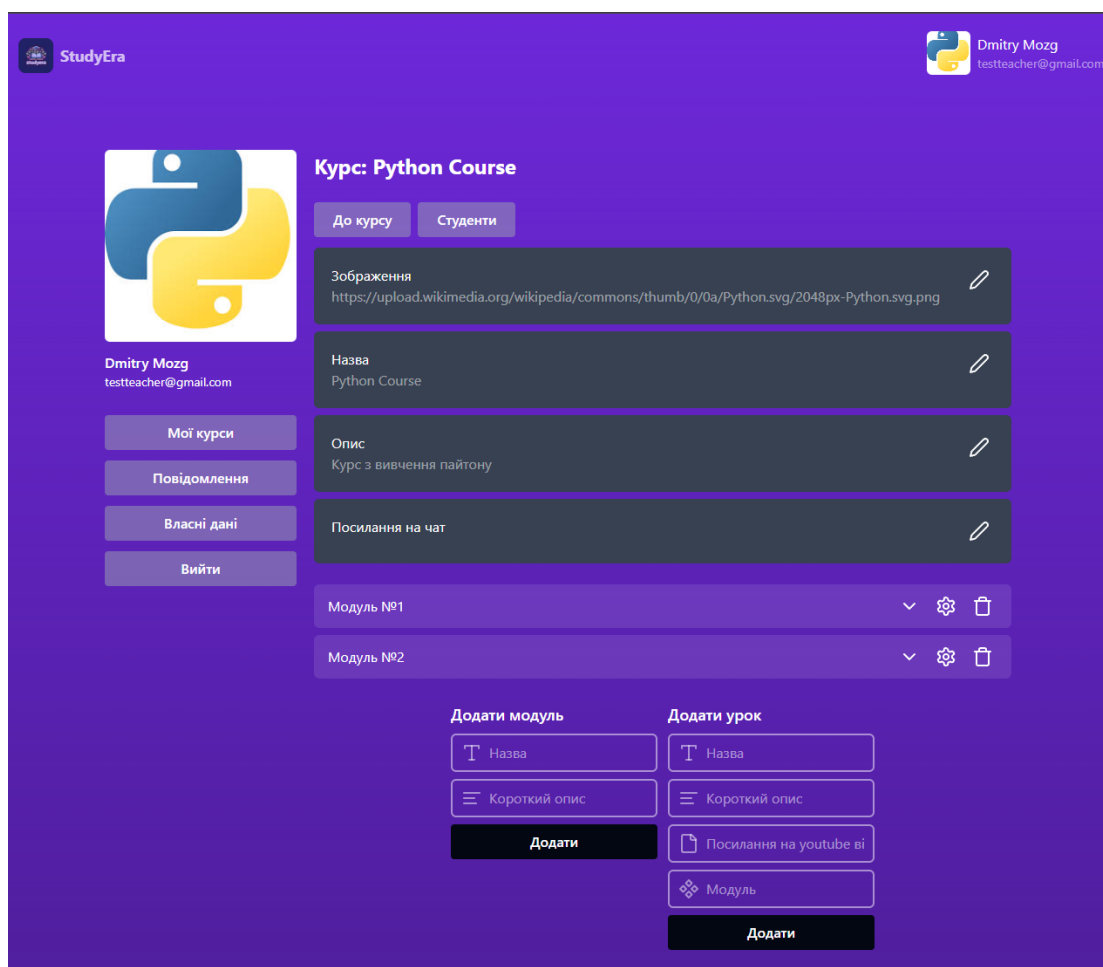


Рис. 3.11– Панель керування обраним курсом

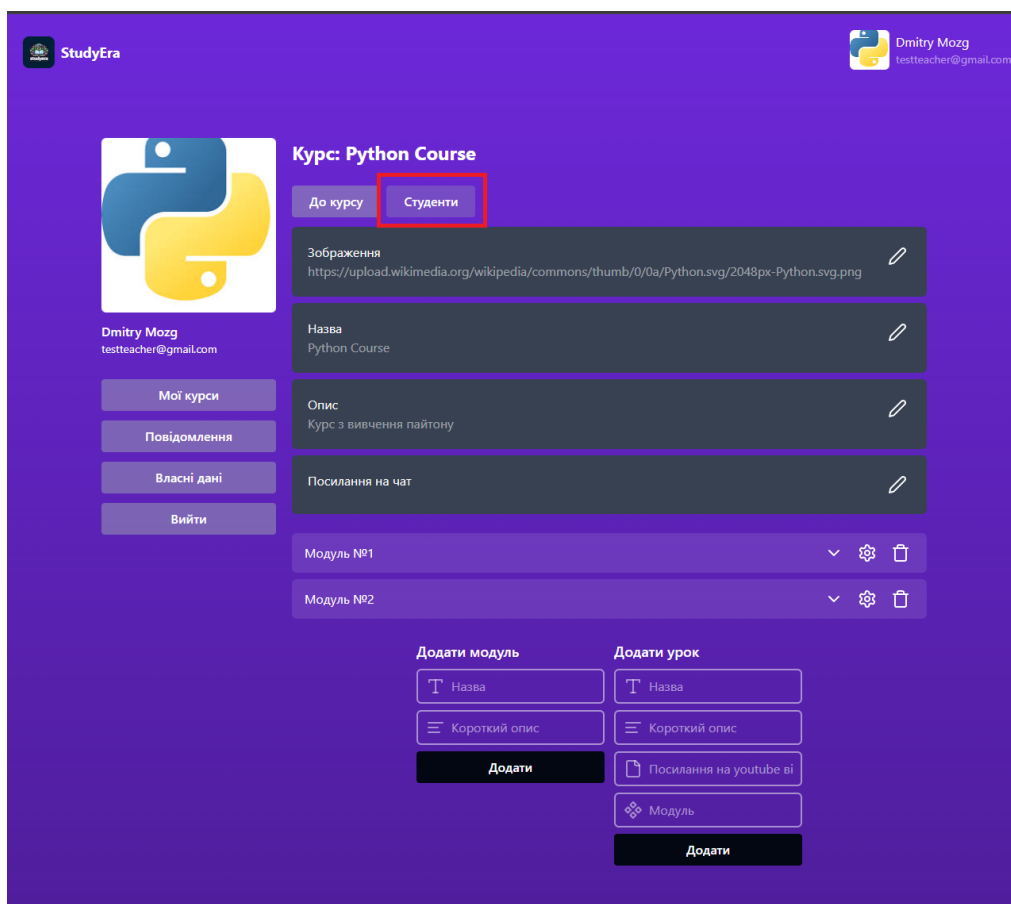


Рис. 3.12 – Кнопка для перегляду усіх користувачів зареєстрованих на обраний навчальний курс

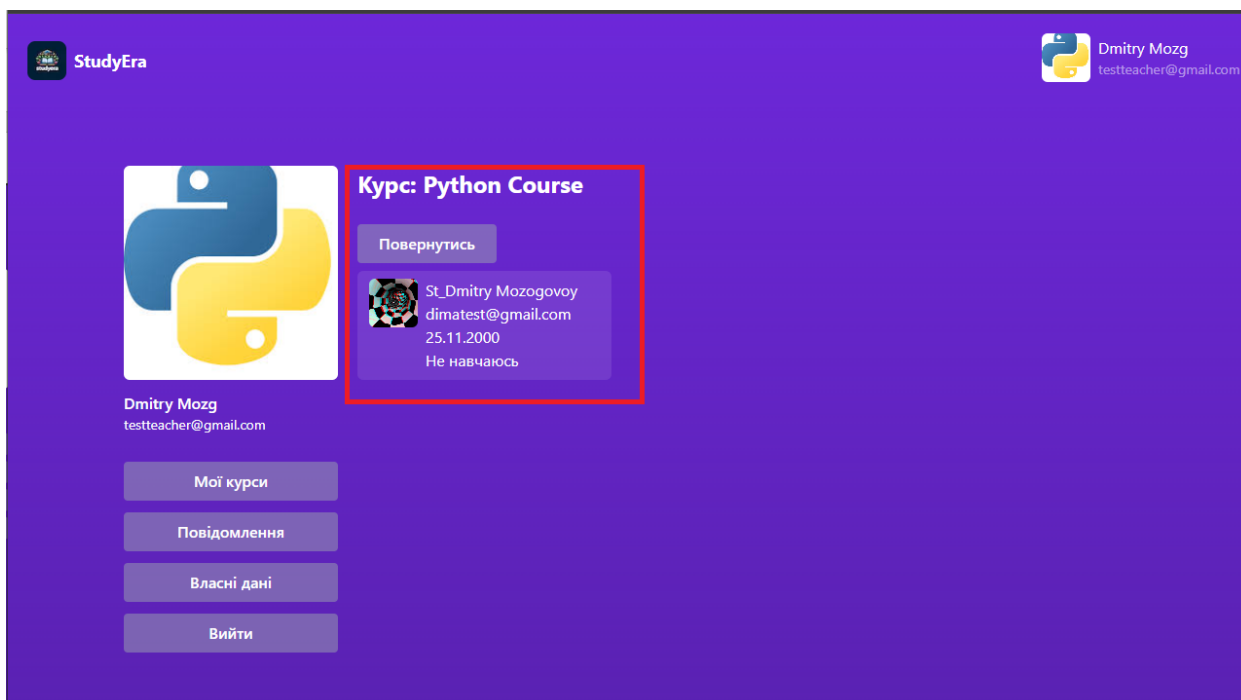


Рис. 3.13 – Перелік студентів, котрі зареєстровані на навчальний курс

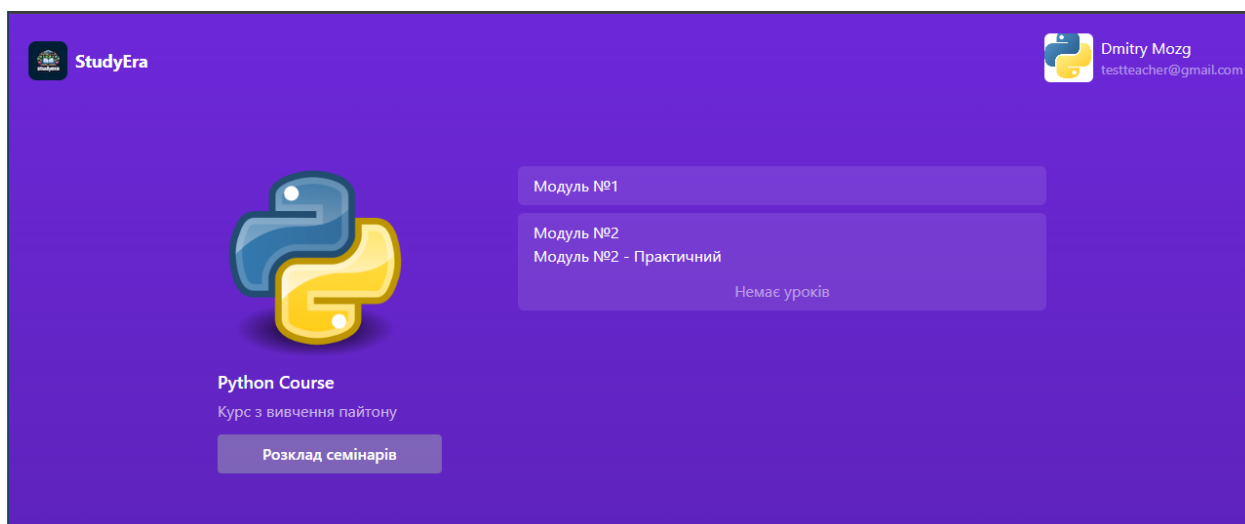


Рис. 3.14 – Навчальний модуль №2 до додавання нового матеріалу

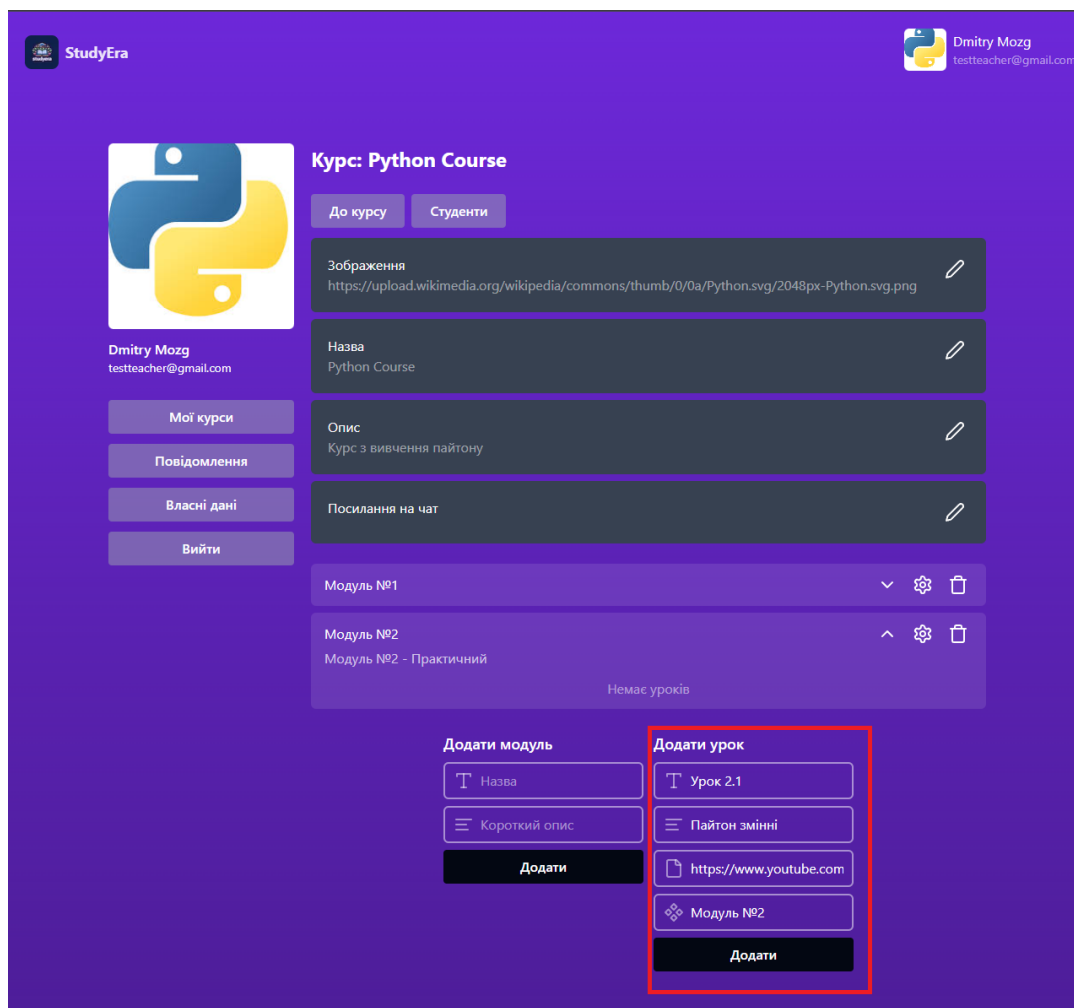


Рис. 3.15 – Додавання нового уроку до навчального модуля №2

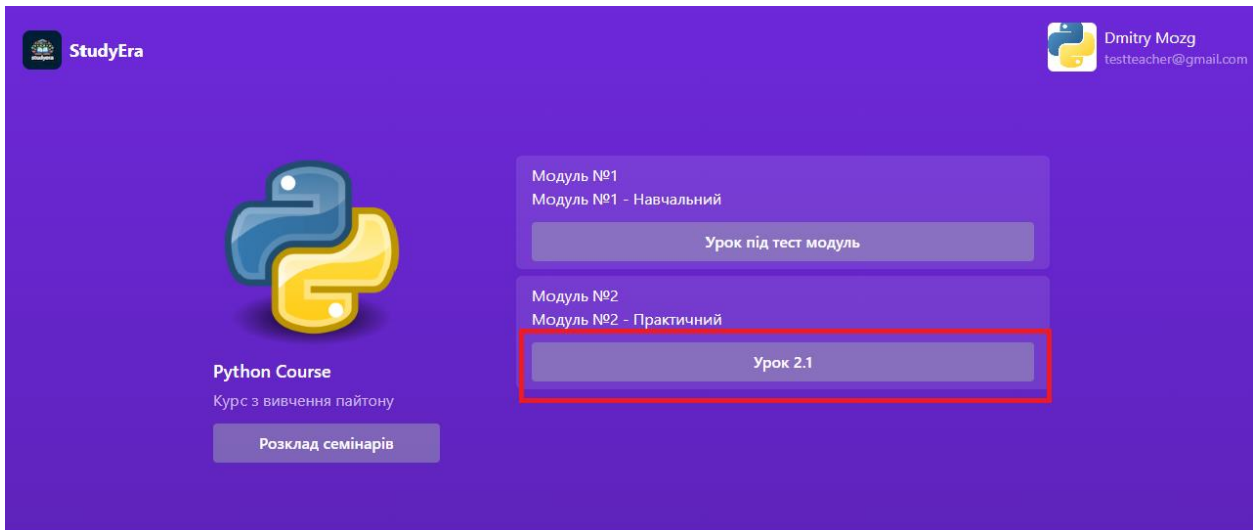


Рис. 3.16 – Результат додавання нового навчального уроку до курсу.

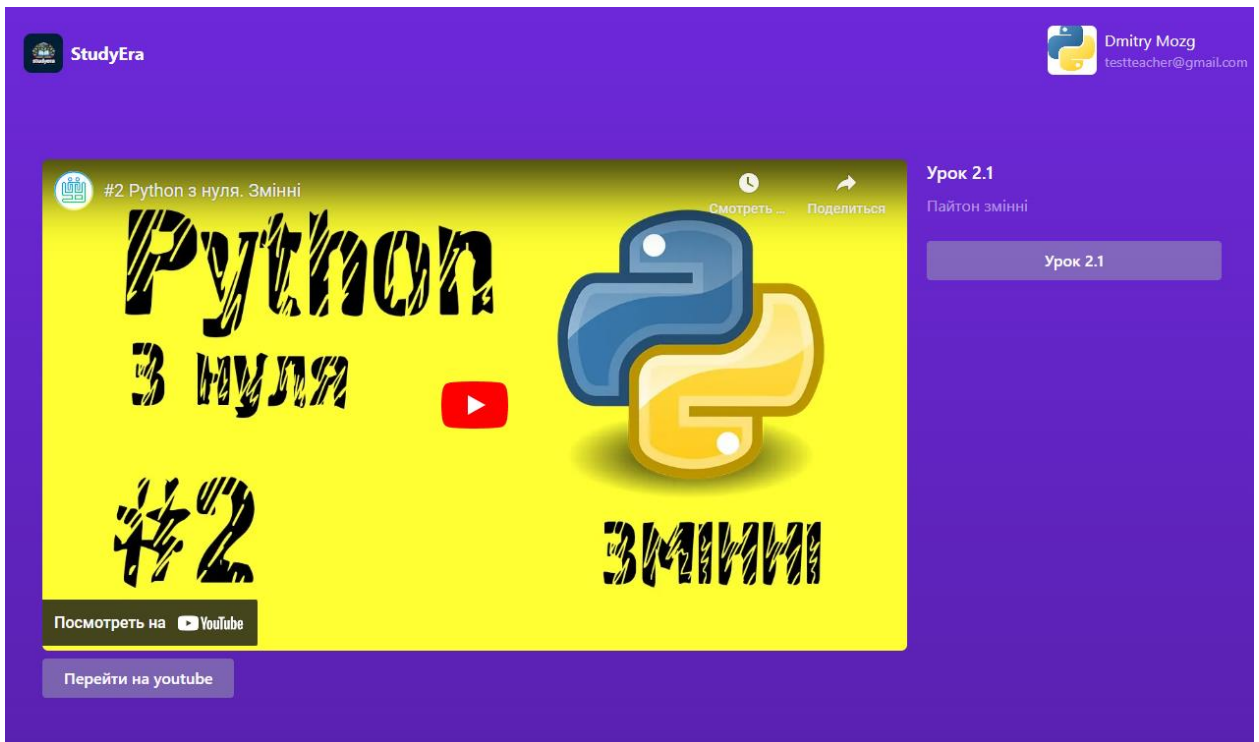


Рис. 3.17 – Візуальне підтвердження новоствореного уроку на навчальному курсі.

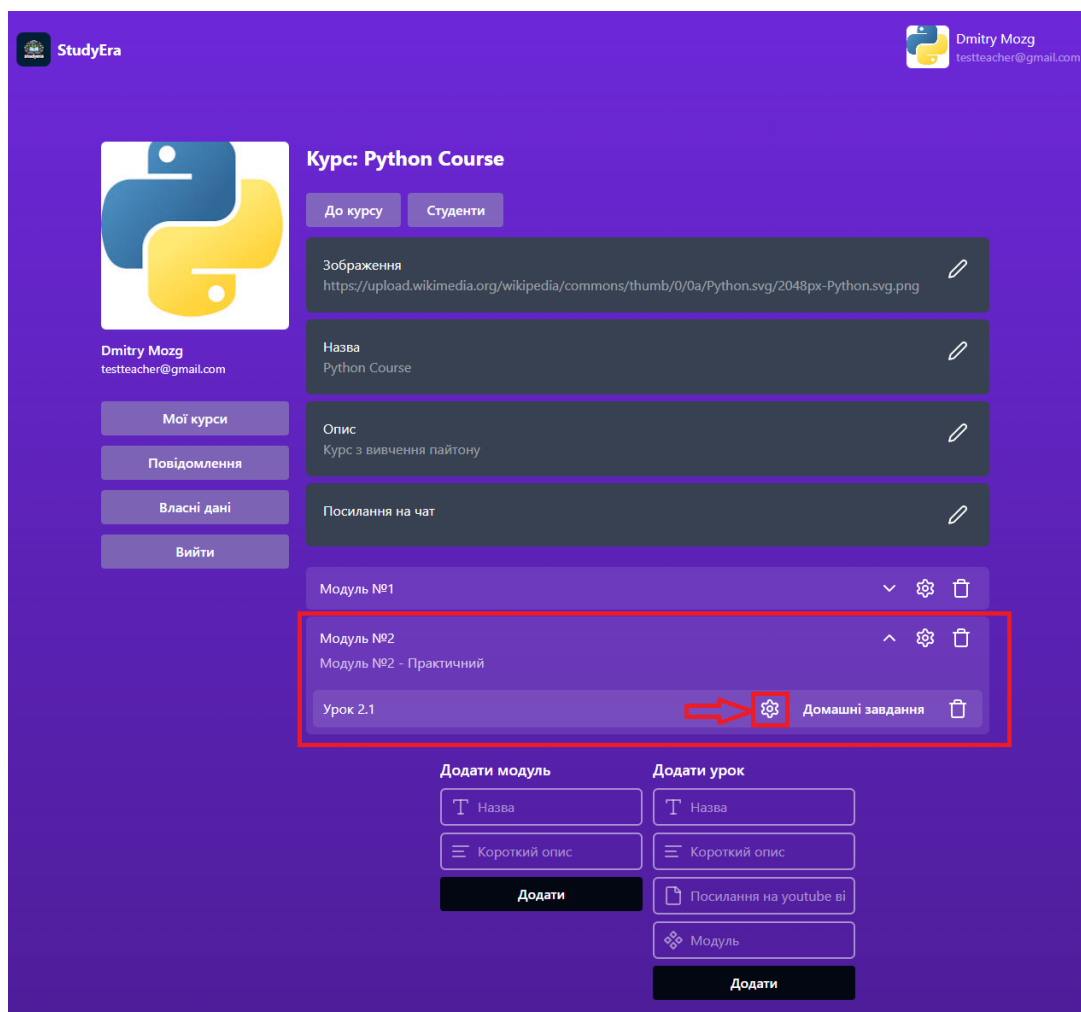


Рис. 3.18 – Навігаційна кнопка для налаштування домашнього завдання за обраним уроком

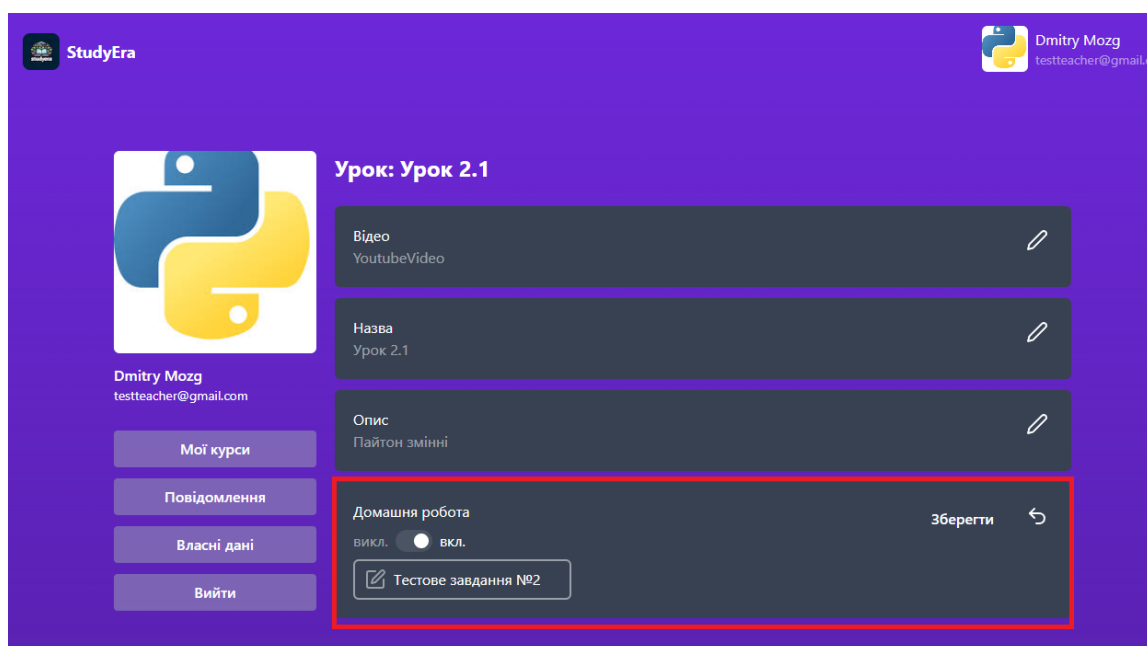


Рис. 3.19 – Панель редагування та закріплення домашнього завдання за певним уроком

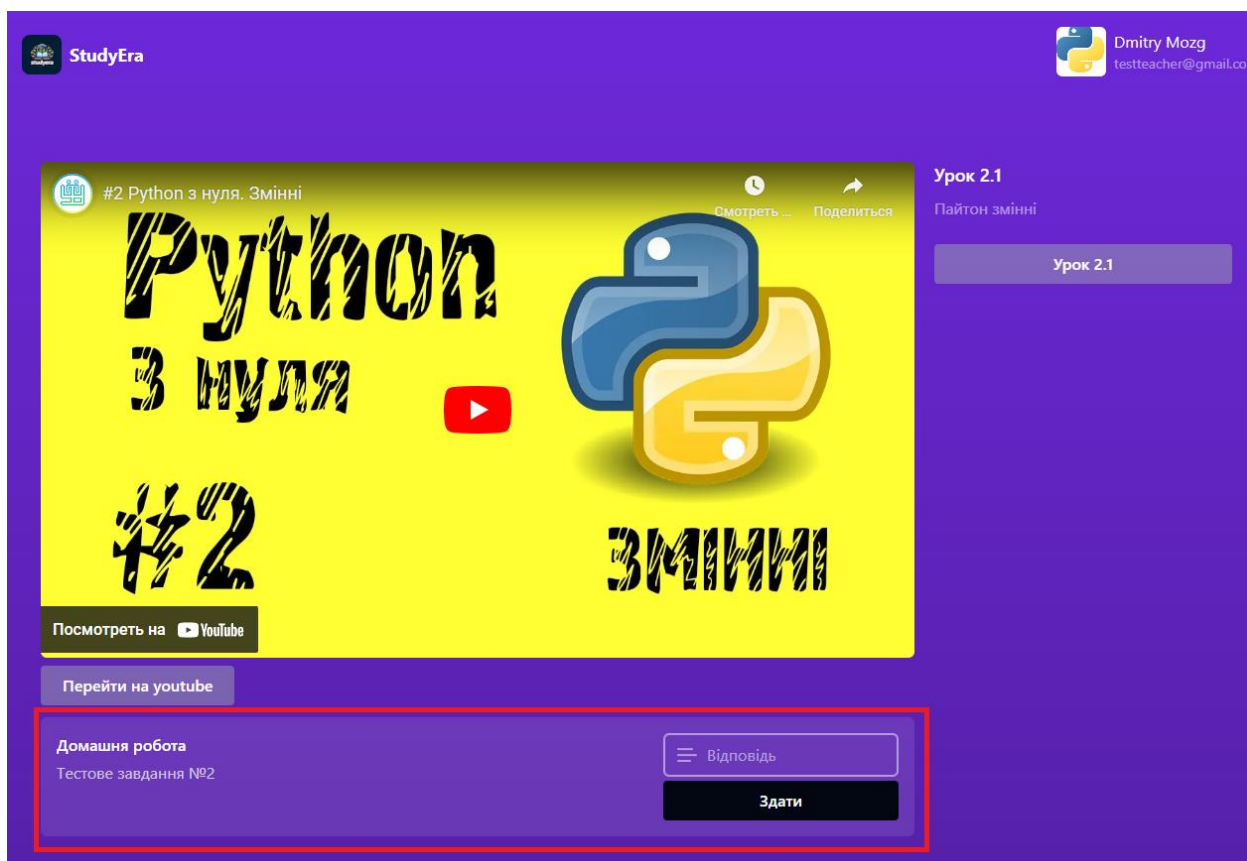


Рис. 3.20 - Візуальне підтвердження додавання та закріплення домашнього завдання за уроком

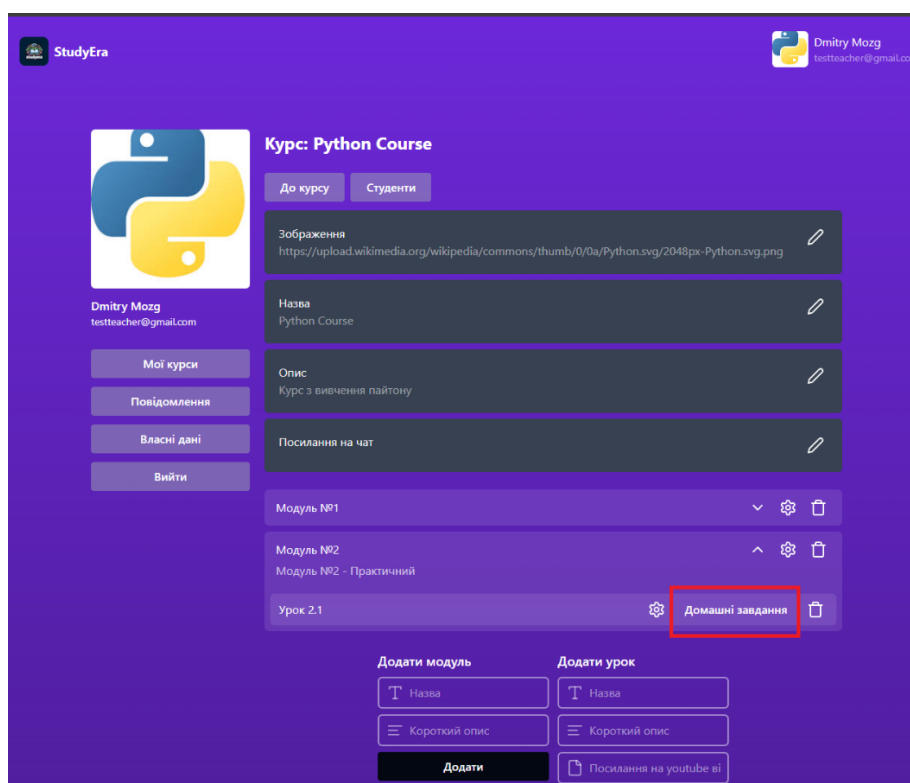


Рис. 3.21 – Спеціальна кнопка для перегляду та оцінки домашніх завдань виконаних студентами під уроком

При перегляді виконаного домашнього завдання, вчитель має змогу переглянути текст відповіді студента, оцінити його роботу та додати коментар з приводу виконання роботи (рис.3.22)

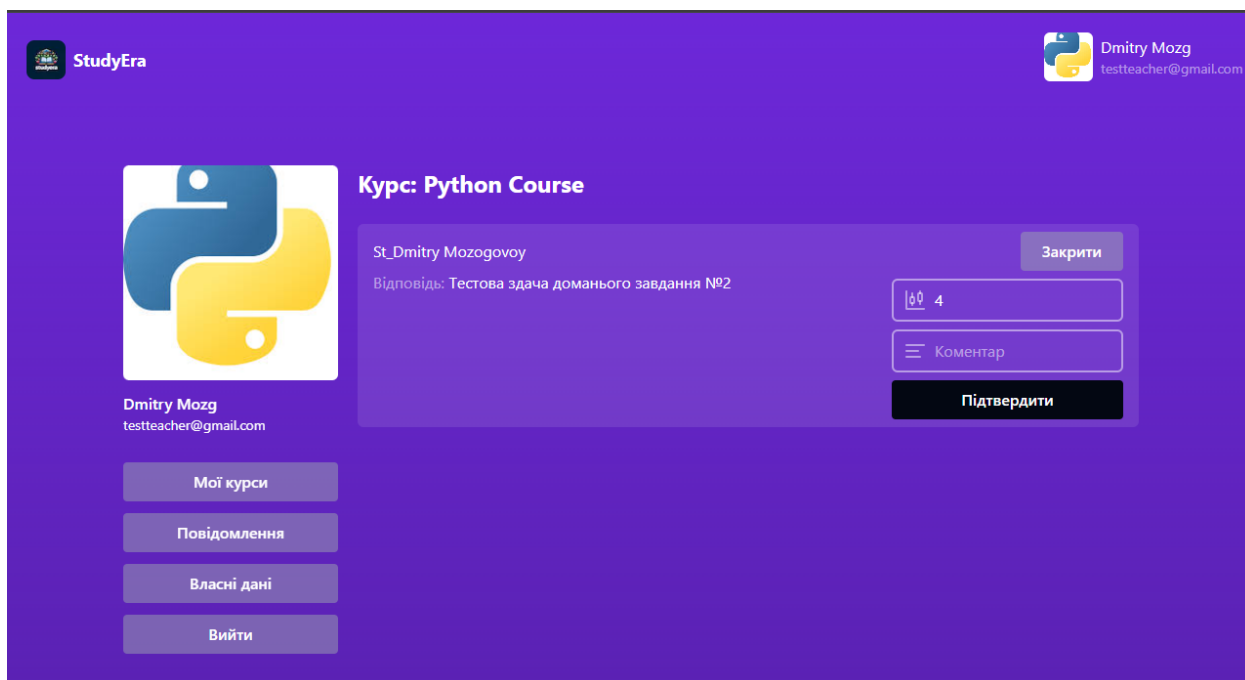


Рис. 3.22 – Панель перегляду та оцінки виконаних домашніх завдань

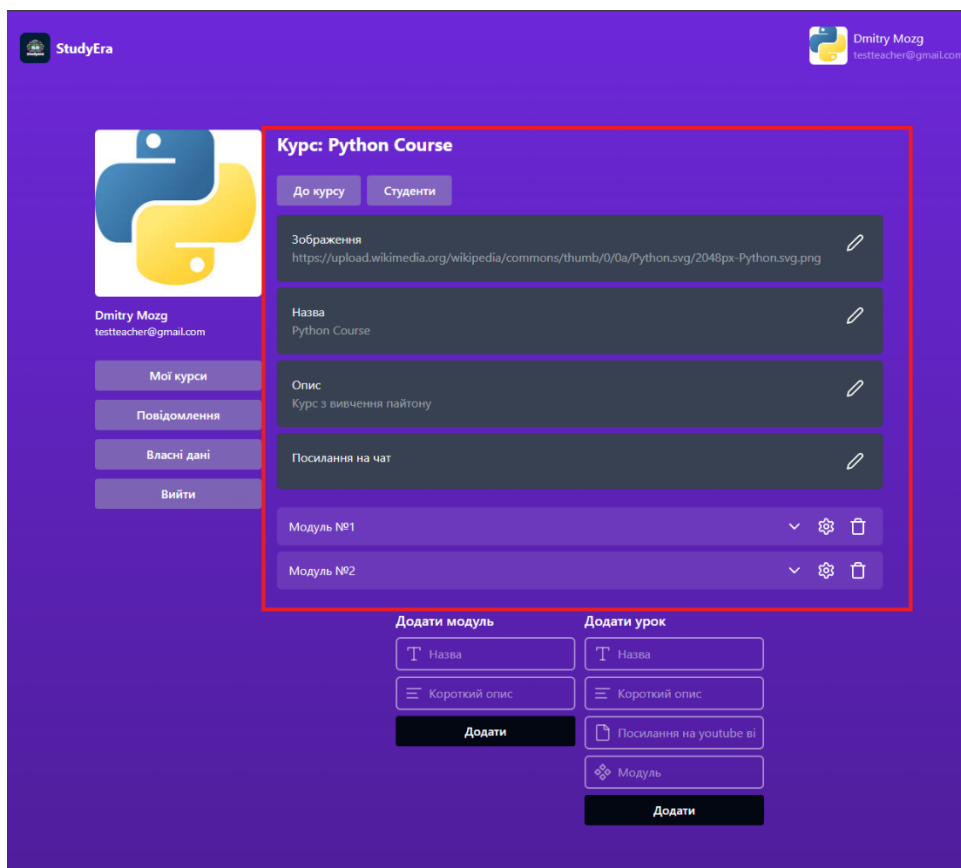


Рис. 3.23 – Панель керування навчальним курсом

The screenshot displays the StudyEra interface for a course titled "Курс: Python Course". The interface is primarily purple and dark grey. At the top left is the StudyEra logo, and at the top right is the user profile for "Dmitry Mozg" with the email "testteacher@gmail.com". The course profile includes a Python logo, the course name, and buttons for "До курсу" and "Студенти". A list of course details is shown, including "Зображення" (Image), "Назва" (Name), and "Опис" (Description). A "Посилання на чат" (Chat link) is highlighted with a red box, showing the URL "https://t.me/+4DlmbihVoI" and a "Зберегти" (Save) button. Below the chat link are two course modules: "Модуль №1" and "Модуль №2".

Рис. 3.24 – Закріплення посилання зовнішнього додатка на спільний чат студентів курсу

ВИСНОВКИ

У ході дослідження та детального аналізу обраної предметної області було переглянуто велику кількість практичних реалізацій використання інформаційних систем пов'язаних з онлайн-освітою. Завдяки цьому аналізу, було сформовано основні функціональні вимоги до системи, а також, визначено ключові функціональні елементи, котрі були частково інтегровані в розроблену систему.

Було розроблено бізнес-вимоги до системи, визначено функціональні та нефункціональні вимоги, такі як: швидкість, масштабованість, доступність, відмовостійкість та інтерфейс користувача. Створено UML діаграми для моделювання системи, що відображають взаємодію користувачів з системою та внутрішню структуру системи.

Проектування та реалізація компонентів системи включали кілька ключових етапів. Було спроектовано інформаційне забезпечення системи, враховуючи інтерфейси як для студентів, так і для вчителів. Розроблено організаційне забезпечення, що охоплює діаграми станів і дій для різних сценаріїв використання системи різними типами користувачів.

Розроблені та протестовані функціональні можливості системи включають реєстрацію та аутентифікацію користувачів, керування навчальними курсами, додавання навчальних матеріалів і перевірку домашніх завдань. Проведене тестування користувацького інтерфейсу забезпечило зручність використання системи для різних категорій користувачів. Взаємодія з зовнішніми додатками підвищила ефективність комунікації між студентами та викладачами.

Підсумовуючи, розроблена інформаційна системи "StudyEra" є досить зручним інструментом для дистанційної освіти, що відповідає сучасним вимогам, які були визначені при дослідженні практичних джерел. Система забезпечує ефективну організацію навчального процесу, зручність використання та можливість розширення, що підвищує якість освіти і доступність навчальних матеріалів для всіх користувачів. Створена інформаційної системи, може використовуватися для подальшого розвитку

дистанційної освіти в Україні. Результати дослідження та розробки можуть бути корисними для освітніх установ, розробників програмного забезпечення та всіх зацікавлених у розвитку онлайн навчання.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. KWIGA [Електронний ресурс] – доступ з: <https://kwiga.com/ua> Дата звернення: Червень 10, 2024 року.
2. PROMETHEUS [Електронний ресурс] – доступ з: <https://prometheus.org.ua/> Дата звернення: Червень 10, 2024 року.
3. SendPulse [Електронний ресурс] – доступ з: <https://sendpulse.com/ua> Дата звернення: Червень 10, 2024 року.
4. Google Classroom [Електронний ресурс] – доступ з: <https://classroom.google.com/> Дата звернення: Червень 10, 2024 року.
5. Google Chrome [Електронний ресурс] – доступ з: <https://www.google.com/chrome/> Дата звернення: Червень 10, 2024 року.
6. Mozilla Firefox [Електронний ресурс] – доступ з: <https://www.mozilla.org/firefox/> Дата звернення: Червень 10, 2024 року.
7. Safari [Електронний ресурс] – доступ з: <https://www.apple.com/safari/> Дата звернення: Червень 10, 2024 року.
8. Microsoft Edge [Електронний ресурс] – доступ з: <https://www.microsoft.com/edge> Дата звернення: Червень 10, 2024 року.
9. Telegram [Електронний ресурс] – доступ з: <https://telegram.org/> Дата звернення: Червень 10, 2024 року.
10. Google Calendar [Електронний ресурс] – доступ з: <https://calendar.google.com/>. Дата звернення: Червень 10, 2024 року
11. Next.js [Електронний ресурс] – доступ з: <https://nextjs.org/docs> Дата звернення: Червень 10, 2024 року.
12. ASP.NET Core [Електронний ресурс] – доступ з: <https://docs.microsoft.com/en-us/aspnet/core/> Дата звернення: Червень 10, 2024 року.

13. TypeScript (TS) [Електронний ресурс] – доступ з: <https://www.typescriptlang.org/docs/> . Дата звернення: Червень 10, 2024 року.
14. C# [Електронний ресурс] – доступ з: <https://docs.microsoft.com/en-us/dotnet/csharp/> Дата звернення: Червень 10, 2024 року.
16. AutoMapper [Електронний ресурс] – доступ з: <https://docs.automapper.org/en/latest/Getting-started.html> Дата звернення: Червень 10, 2024 року.
17. FluentValidation [Електронний ресурс] – доступ з: <https://docs.fluentvalidation.net/en/latest/> Дата звернення: Червень 10, 2024 року.
18. Swagger [Електронний ресурс] – доступ з: <https://swagger.io/docs/>. Дата звернення: Червень 10, 2024 року.
19. SQL Server [Електронний ресурс] – доступ з: <https://learn.microsoft.com/en-us/sql/sql-server/connect-to-database-engine?view=sql-server-ver16&tabs=sqlldb> Дата звернення: Червень 10, 2024 року.
20. Oracle MySQL Database [Електронний ресурс] – доступ з: <https://docs.oracle.com/en-us/iaas/mysql-database/doc/connecting-dbsystem1.html> Дата звернення: Червень 10, 2024 року.
21. Entity Framework Core [Електронний ресурс] – доступ з: <https://learn.microsoft.com/en-us/ef/core/managing-schemas/ensure-created>. Дата звернення: Червень 10, 2024 року.
22. Troelsen, A., Japikse, P. Pro C# 7: With .NET and .NET Core. – Apress, 2017.
23. Пістунов І.В. Інформаційні системи [Електронний ресурс] – доступ з: http://pistunovi.inf.ua/II_I_C.pdf. Дата звернення: Червень 10, 2024 року.

ДОДАТКИ А

Бекенд частина:

Бізнес логіка «CourseService» для взаємодії з курсами.

namespace Backend.Core.Futures.MaterialsForStudy.Courses;

```
public class CourseService : ICourseService
{
    private readonly ICourseGateway _courseGateway;
    private readonly IMapper _mapper;
    private readonly IValidator<CreateCourseDto> _createCourseValidator;
    private readonly IValidator<UpdateCourseImageByUrlDto>
_updateImageByUrlValidator;
    private readonly IUserContext _userContext;

    public CourseService(ICourseGateway courseGateway, IMapper mapper,
IValidator<CreateCourseDto> createCourseValidator,
IValidator<UpdateCourseImageByUrlDto> updateImageByUrlValidator,
IUserContext userContext)
    {
        _courseGateway = courseGateway;
        _mapper = mapper;
        _createCourseValidator = createCourseValidator;
        _updateImageByUrlValidator = updateImageByUrlValidator;
        _userContext = userContext;
    }

    public async Task<Response> CreateAsync(CreateCourseDto dto)
    {
        if (await _createCourseValidator.ValidateAsync(dto) is { IsValid: false }
validationResult)
            return Response.ValidationFailed(validationResult.Errors);
    }
}
```

```

    var course = _mapper.Map<Course>(dto);
    var isAdded = await _courseGateway.AddAsync(course);
    return Response.Result(isAdded);
}

```

```

public async Task<Response> RemoveAsync(RemoveCourseDto dto)
{
    var isRemoved = await _courseGateway.RemoveAsync(dto.Id);
    return Response.Result(isRemoved);
}

```

```

public async Task<Response<IEnumerable<ShortCourseDto>>>
GetAllCourses()
{
    var courses = await _courseGateway.GetAllAsync();
    return
Response.Success(_mapper.Map<IEnumerable<ShortCourseDto>>(courses));
}

```

```

public async Task<Response<ActualCourseDto>>
UpdateTitleAsync(UpdateCourseTitleDto dto)
{
    var actual = await _courseGateway.UpdateAsync(dto.Id, o =>
    {
        o.Title = dto.Title;
    });
    return Response.Success(_mapper.Map<ActualCourseDto>(actual));
}

```

```

public async Task<Response<ActualCourseDto>>
UpdateDescriptionAsync(UpdateCourseDescriptionDto dto)
{
    var actual = await _courseGateway.UpdateAsync(dto.Id, o =>
    {
        o.Description = dto.Description;
    });
    return Response.Success(_mapper.Map<ActualCourseDto>(actual));
}

public async Task<Response<ActualCourseDto>>
UpdateImageByUrlAsync(UpdateCourseImageByUrlDto dto)
{
    if (await _updateImageByUrlValidator.ValidateAsync(dto) is { IsValid: false
} result)
        return Response.ValidationFailed<ActualCourseDto>(result.Errors);

    var actual = await _courseGateway.UpdateAsync(dto.Id, o =>
    {
        o.ImageUrl = dto.ImageUrl;
    });
    return Response.Success(_mapper.Map<ActualCourseDto>(actual));
}

public async Task<Response<ActualCourseDto>>
UpdateChatLinkAsync(UpdateCourseChatLinkDto dto)
{
    var actual = await _courseGateway.UpdateAsync(dto.Id, o =>
    {
        o.ChatLink = dto.ChatLink;
    });
}

```

```

        return Response.Success(_mapper.Map<ActualCourseDto>(actual));
    }

    public async Task<Response<CoursePageDto>> GetCoursePageDataAsync(int
courseId)
    {
        var course = await
_courseGateway.GetByIdWithTeacherAndModulesAsync(courseId);
        if (course == null)
            Response.Failed();

        return Response.Success(_mapper.Map<CoursePageDto>(course));
    }

    public async Task<Response<double>> GetAverageScoreAsync(int courseId)
    {
        var score = await _courseGateway.GetAverageScoreAsync(courseId,
_userContext.UserId);
        return Response.Success(score ?? 0);
    }
}

```

Бізнес логіка «HomeworkService» - для взаємодії з домашніми завданнями на платформі

```

public class HomeworkService : IHomeworkService
{
    private readonly IHomeworkGateway _homeworkGateway;
    private readonly IMapper _mapper;

    public HomeworkService(IHomeworkGateway homeworkGateway, IMapper
mapper)

```

```
{
    _homeworkGateway = homeworkGateway;
    _mapper = mapper;
}

public async Task<Response> SendAsync(SendHomeworkDto dto)
{
    try
    {
        await
_homeworkGateway.AddOrUpdateAsync(_mapper.Map<Homework>(dto));
        return Response.Success();
    }
    catch
    {
        return Response.Failed();
    }
}

public async Task<Response> EvaluateAsync(EvaluateHomeworkDto dto)
{
    try
    {
        await _homeworkGateway.EvaluateAsync(dto.Id, dto.Appraisal,
dto.Comment);
        return Response.Success();
    }
    catch
    {
        return Response.Failed();
    }
}
```

```

    }

    public async Task<Response<IEnumerable<CompletedHomeworkDto>>>
    GetByLessonAsync(int lessonId)
    {
        var homeworks = await
        _homeworkGateway.GetByLessonIdWithStudentAsync(lessonId);
        return
        Response.Success(_mapper.Map<IEnumerable<CompletedHomeworkDto>>(hom
        eworks));
    }

    public async Task<Response> CancelSendingAsync(CancelHomeworkDto dto)
    {
        await _homeworkGateway.CancelSendingAsync(dto.Id);
        return Response.Success();
    }

    public async Task<Response<StudentHomeworkDto>> GetByStudentAsync(int
    studentId, int lessonId)
    {
        var student = await
        _homeworkGateway.GetByStudentAndLessonIdAsync(studentId, lessonId);
        return Response.Success(_mapper.Map<StudentHomeworkDto>(student));
    }
}

```

Бізнес логіка «StudentService» - для керування обліковим записом користувача студент:

```
public class StudentService : IStudentService
```

```

{
    private readonly IStudentGateway _studentGateway;
    private readonly ICourseGateway _courseGateway;
    private readonly IMapper _mapper;
    private readonly IUserContext _userContext;

    public StudentService(IStudentGateway studentGateway, IMapper mapper,
        ICourseGateway courseGateway, IUserContext userContext)
    {
        _studentGateway = studentGateway;
        _mapper = mapper;
        _courseGateway = courseGateway;
        _userContext = userContext;
    }

    public async Task<Response<IEnumerable<StudentDataDto>>> GetAllAsync()
    {
        var students = await _studentGateway.GetAllAsync();
        return
        Response.Success(_mapper.Map<IEnumerable<StudentDataDto>>(students));
    }

    public async Task<Response> JoinCourseAsync(JoinCourseDto dto)
    {
        if (await _courseGateway.HaveStudentAsync(_userContext.UserId,
            dto.CourseId))
            return JoinCourseResponseHelper.StudentAlreadyHaveCourse();

        if (await _courseGateway.HaveFreeSlotsAsync(dto.CourseId) == false)
            return JoinCourseResponseHelper.HaventFreeSlots();
    }
}

```

```

        var isJoined = await _studentGateway.JoinCourse(_userContext.UserId,
dto.CourseId);
        return Response.Result(isJoined);
    }

public async Task<Response<StudentDataDto>> GetMyDataAsync()
{
    var student = await _studentGateway.GetByIdAsync(_userContext.UserId);
    return student == null
        ? GetStudentByIdResponseHelper.Failed()
        : Response.Success(_mapper.Map<StudentDataDto>(student));
}

public async Task<Response<IEnumerable<StudentCourseDto>>>
GetCoursesAsync()
{
    var courses = await
_courseGateway.GetStudentCoursesWithModulesAndLessonsAsync(_userContext
.UserId);
    return
Response.Success(_mapper.Map<IEnumerable<StudentCourseDto>>(courses));
}

public async Task<Response<bool>> AlreadyJoinedCourseAsync(int courseId)
{
    return Response.Success(await
_courseGateway.HaveStudentAsync(_userContext.UserId, courseId));
}

public async Task<Response> LeaveCourseAsync(LeaveCourseDto dto)
{

```

```

        if (await _courseGateway.HaveStudentAsync(_userContext.UserId,
dto.CourseId) == false)
            return LeaveCourseResponseHelper.NotJoinedCourse();

        var isLeaved = await _studentGateway.LeaveCourse(_userContext.UserId,
dto.CourseId);
        return LeaveCourseResponseHelper.Result(isLeaved);
    }

    public async Task<Response<IEnumerable<StudentDataDto>>>
GetAllByCourseAsync(int courseId)
    {
        var students = await _studentGateway.GetByCourseIdAsync(courseId);
        return
Response.Success(_mapper.Map<IEnumerable<StudentDataDto>>(students));
    }
}

```

Бізнес логіка «TeacherService» - для керування обліковим записом користувача
ВЧИТЕЛЬ:

```

public class TeacherService : ITeacherService
{
    private readonly IUserGateway _userGateway;
    private readonly ITeacherGateway _teacherGateway;
    private readonly IMapper _mapper;
    private readonly IValidator<CreateTeacherDto> _createTeacherValidator;

    public TeacherService(IUserGateway userGateway, IMapper mapper,
IValidator<CreateTeacherDto> createTeacherValidator, ITeacherGateway
teacherGateway)
    {

```

```

    _userGateway = userGateway;
    _mapper = mapper;
    _createTeacherValidator = createTeacherValidator;
    _teacherGateway = teacherGateway;
}

```

```

public async Task<Response> CreateAsync(CreateTeacherDto dto)
{
    var result = await _createTeacherValidator.ValidateAsync(dto);
    if (result.IsValid == false)
        return Response.ValidationFailed<bool>(result.Errors);

    var teacher = _mapper.Map<Teacher>(dto);
    var isAdded = await _teacherGateway.AddAsync(teacher);
    return isAdded
        ? Response.Success()
        : Response.Failed();
}

```

```

public async Task<Response<IEnumerable<TeacherShortCourseDto>>>
GetCoursesAsync(int userId)
{
    var courses = await _teacherGateway.GetCoursesAsync(userId);
    return courses == null
        ? GetCoursesResponseHelper.Failed()
        :
Response.Success(_mapper.Map<IEnumerable<TeacherShortCourseDto>>(course
s));
}

```

```

public async Task<Response> RemoveAsync(RemoveTeacherDto dto)

```

```

    {
        var isRemoved = await _teacherGateway.RemoveAsync(dto.Id);
        return isRemoved
            ? Response.Success()
            : Response.Failed();
    }

    public async Task<Response<IEnumerable<TeacherInformationDto>>>
    GetAllTeachersAsync()
    {
        var allTeachers = await _teacherGateway.GetAllTeachers();
        return
        Response.Success(_mapper.Map<IEnumerable<TeacherInformationDto>>(allTeac
        hers));
    }
}

```

Фронтенд частина:

Сервіс «Authentication» - які відповідають за запити авторизації

```

import $api from '@api/api'
import { ApiResponse } from '@api/response.type'
import { LoginType } from './types/requests/login.type'
import { RegistrationType } from './types/requests/registration.type'
import { SuccessAuthenticationType } from './types/responses/success-
authentication.type'

export const AuthenticationEndpoints = {
    login: '/login',
    registration: '/student/registration',

```

```

    refresh: '/refresh',
    logout: '/logout'
  }
export const AuthenticationService = {
  login: async (data: LoginType) =>
    await $api.post<ApiResponse<SuccessAuthenticationType>>(
      AuthenticationEndpoints.login,
      data
    ),
  registration: async (data: RegistrationType) =>
    await $api.post<ApiResponse<SuccessAuthenticationType>>(
      AuthenticationEndpoints.registration,
      data
    ),
  refresh: async () =>
    await $api.post<ApiResponse<SuccessAuthenticationType>>(
      AuthenticationEndpoints.refresh
    ),
  logout: async () =>
    await $api.post<ApiResponse>(AuthenticationEndpoints.logout)
}

```

Сервіс «Student» - відповідає за керування методами відправлення запитів на сервер.

```

export const StudentEndpoints = {
  getAll: '/student/all',
  joinCourse: '/student/join-course',
  leaveCourse: '/student/leave-course',
  getMyData: '/student/get-my-data',
  getCourses: '/student/courses',
  alreadyJoinedCourse: (courseId: number) =>

```

```

    `/student/already-joined-course?courseId=${courseId}`,
    getByCourse: (courseId: number) =>
      `/student/get-by-course?courseId=${courseId}`
  }
export const StudentService = {
  getMyData: async () => await $api.get(StudentEndpoints.getMyData),
  getCourses: async () => await $api.get(StudentEndpoints.getCourses),
  alreadyJoinedCourse: async (courseId: number) =>
    await $api.get(StudentEndpoints.alreadyJoinedCourse(courseId)),
  joinCourse: async (courseId: number) =>
    await $api.post(StudentEndpoints.joinCourse, { courseId }),
  leaveCourse: async (courseId: number) =>
    await $api.post(StudentEndpoints.leaveCourse, { courseId }),
  getByCourse: async (courseId: number) =>
    await $api.get(StudentEndpoints.getByCourse(courseId))
}

```

«Layout» - шаблон для усіх сторінок сайту

```

export const metadata: Metadata = {
  title: "StudyEra",
  description: "Generated by create StudyEra",
  icons: LogoImage.src
};

```

```

export default function RootLayout({
  children,
}: Readonly<{
  children: React.ReactNode;
}>) {
  return (
    <html lang="en" className="h-full">
      <body className={styles.body}>

```

```

    <Providers>
      <MobileNavigation />
      <Header />
      {children}
    </Providers>
  </body>
</html>
);
}
«Providers»
const queryClient = new QueryClient()

const Providers = ({children}: {children: React.ReactNode}) => {
  return (
    <QueryClientProvider client={queryClient}>
      <Provider store={store}>
        <PersistGate loading={false} persistor={persistor}>
          {children}
        </PersistGate>
      </Provider>
    </QueryClientProvider>
  )
}

export default Providers

```

ДОДАТКИ Б

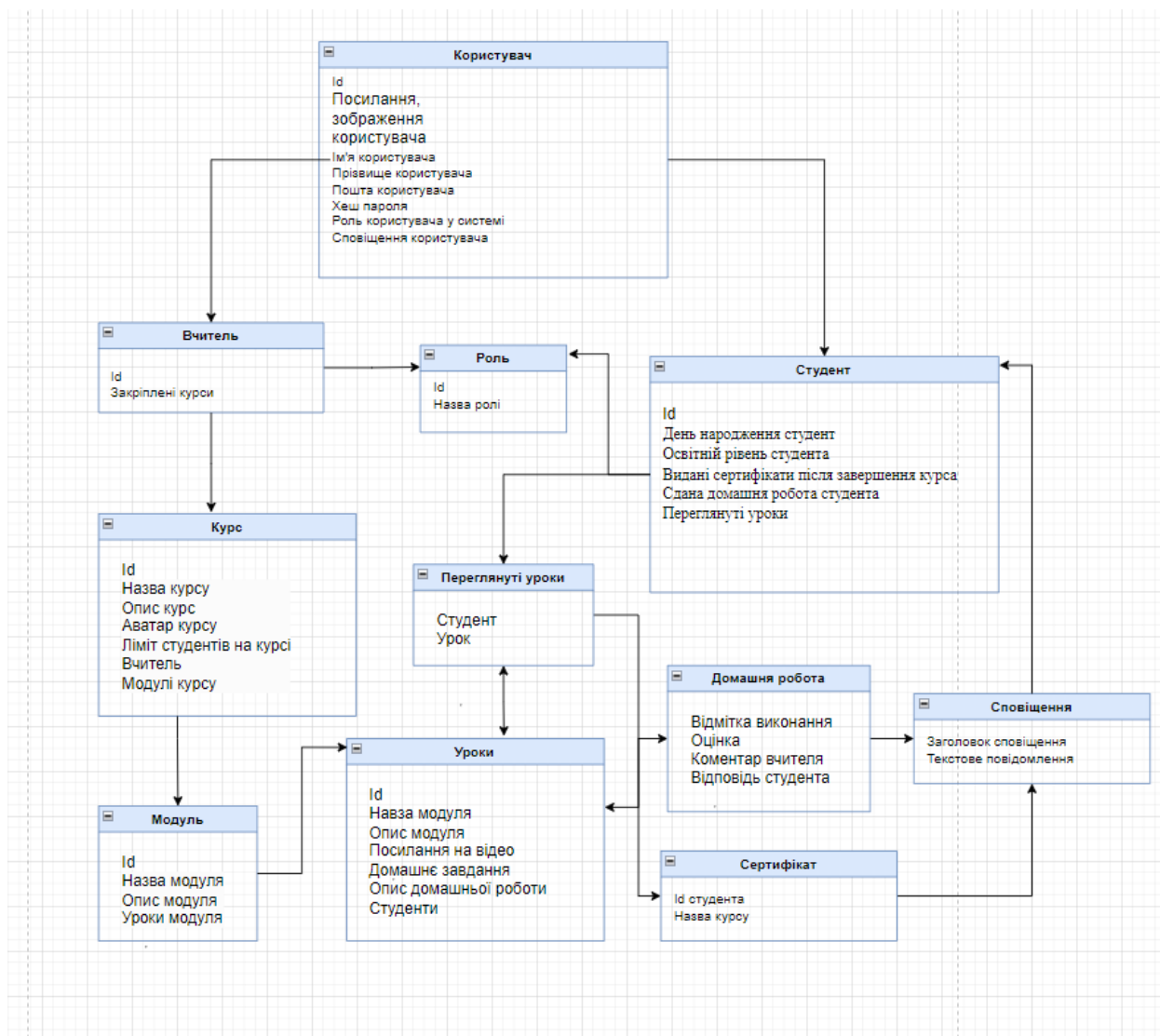


Рис. 3.25 - Інфологічна модель бази даних.



Ім'я користувача:
Інформаційних систем в економіці Шкуратовська Те...

ID перевірки:
1016355032

Дата перевірки:
13.06.2024 07:01:42 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
13.06.2024 11:07:15 EEST

ID користувача:
100005745

Назва документа: Мозговий_Д_ІН-404

Кількість сторінок: 61 Кількість слів: 6941 Кількість символів: 55616 Розмір файлу: 5.48 MB ID файлу: 1016157242

10.4% Схожість

Найбільша схожість: 3.82% з джерелом з Бібліотеки (ID файлу: 1015232018)

3.27% Джерела з Інтернету

407

Сторінка 63

9.87% Джерела з Бібліотеки

489

Сторінка 65

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

2

Назва документа: Мозговий_Д_ІН-404

ID файлу:
1016157242

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ПРОЕКТ

на тему

**ІНФОРМАЦІЙНА СИСТЕМА НАВЧАЛЬНО-ОСВІТНЬОГО ЦЕНТРУ
РОЗВИТКУ**

здобувача Мозгового Дмитра Анатолійовича

Науковий керівник:

к.е.н., доцент

Гордієнко І.В.

**Проект допущений до захисту
перед екзаменаційною комісією з
атестації здобувачів вищої освіти**

завідувач кафедри:

к.е.н., доцент

Тішков Б.О.

Київ 2024

Схожість

Цитати

Посилання

Вилучений
текст

Підміна символів

Коментарі

Джерела на цій сторінці: 1-43, 45, 47-50, 52-57, 59-64, 67, 69-71, 73, 76, 81, 84-85, 87-88, 92, 96, 101-103, 109-110

Сторінка 1 з 68