

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи

на ступінь вищої освіти бакалавр

на тему: «Розробка програмного забезпечення для перевірки виконання
завдань мовою C#»

Виконав: студент 4 курсу, групи ПД–43

спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності/спеціалізації)

_____ Собко І. І.

(прізвище та ініціали)

Керівник _____ Гаманюк І.М.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ –2022

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти -«Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ ____ ” _____ 2022 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТА

Собко Іван Іванович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Створення програмного забезпечення для перевірки

виконання завдань»

Керівник роботи: Гаманюк І.М., ст. викл.

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

Затверджені наказом вищого навчального закладу від «16» лютого 2022 року

№.

2. Строк подання студентом роботи «3» червня 2022 року

3. Вхідні дані до роботи

Методи завантажень файлів; Технічна література програмного забезпечення для перевірки виконання завдань;

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

4.1 Систему реєстрації користувачів з різними ролями

4.2 Вимоги та оцінка якості системи.

4.3 Опис проектування системи.

4.4 Опис використаних технологій.

5. Перелік демонстраційного матеріалу (назва основних слайдів)

5.1 Актуальність проблеми

5.2 Існуюче програмне забезпечення та методи перевірки завдань

5.3 Принцип роботи інформаційної системи

5.4 Розпізнавання та групування даних конкретних користувачів

5.5 Архітектура бази даних

5.6 Логічна діаграма компонентів архітектури програмного забезпечення

6. Дата видачі завдання «11»квітня 2022

КАЛЕНДАРНИЙ ПЛАН

з/п	Назва етапів роботи	бакалаврської	Строк виконання етапів роботи	Примітка
	Підбір літератури	науково-технічної	1.05.22	Виконано

	Вимоги до системи	5.05.22	Ви конано
	Концепція та архітектура програмного забезпечення	10.05.22	Ви конано
	Вступ, висновки, реферат	20.05.22	Ви конано
	Розробка обов'язкових демонстраційних матеріалів	25.05.22	Ви конано
	Попередній захист роботи	3.06.22	
	Здача роботи	20.06.22	

Студент __ _____

(підпис)

(прізвище та ініціали)

Керівник роботи _____

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Мета роботи – покращити спосіб перевірки виконання завдань з використанням розробленої програми.

Об'єкт дослідження – перевірка виконання завдань.

Предмет дослідження – програмне забезпечення для перевірки виконання завдань.

Методи дослідження – методи перевірки виконання завдань.

При створенні проведено розбір створених програмних продуктів, таких як Google Classroom, Moodle, Microsoft Teams для аналізу функціоналу завантаження та оцінювання робіт

Застосунок об'єднує все в спільному середовищі, яке містить коментарі та файл обмінник. Веб-додаток написано на ASP.NET Core 5 (MVC), при використанні бази даних Postgresql.

Вчитель має можливість створити групу і додати учнів та викладати завдання з описом, дані задачі та файли. Також з можливістю перевірити, виставить оцінку та залишити коментарі. Користувачі з роллю учень можуть завантажувати файлові завдання. Адміністратор створює учнів і групу.

Отже, розроблено та описано веб-додаток, завданням якого є надає можливість проектувати, створювати та керувати інформаційно-навчальними ресурсами.

Даний додаток може бути використано у всіх сферах, які потребують насамперед, на забезпечення інтерактивної взаємодії між учасниками навчально процесу.

ЗМІСТ

ВСТУП.....	Error!
ror! Bookmark not defined.	
РОЗДІЛ 1. АНАЛІЗ НАЯВНИХ ЗАСОБІВ ТА ТЕХНОЛОГІЙ ДЛЯ ОРГАНІЗАЦІЇ ПЕРЕВІРКИ ВИКОНАННЯ ЗАВДАНЬ	13
1.1. Основні форми онлайн-комунікації.....	13
1.2. Поширені веб-ресурси для дистанційного навчання.....	15
1.3. Інструментарій дистанційного навчання.....	21
1.4. Переваги та недоліки дистанційного навчання.....	24
1.5. Висновок.....	25
РОЗДІЛ 2. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	27
2.1. Структурна схема комплексної моделі	27
2.2. Архітектура програмного забезпечення.....	28
2.3. Модель системи.....	31
2.4. Висновок.....	34
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	36
3.1. Проект та під проект.....	36
3.2. DataAccess та DataAccess.Migrations.....	39
3.3. Host.....	42
3.4. BussinessLogic.....	50
3.5. AutoMapper.....	Error!
Bookmark not defined.	55
3.6. Висновок.....	56
Висновки.....	58
ПЕРЕЛІК ПОСИЛАНЬ.....	59

ВСТУП

Актуальність дослідження. Сьогодні питання освіти є дуже актуальним. Не у всіх є можливість навчатися по 5-7 годин на день на уроці чи проводити тижні на сесії під час дистанційного навчання, але дистанційне навчання дуже актуальне та корисне. Вона користується великим попитом у всьому світі, але в той же час викликає багато запитань перед навчальними закладами. Як організувати навчальний процес. Коли проводити виховні заходи. За допомогою якої програми виконується навчання.

Комп'ютерна освіта має велике значення в наш час. Загострення конкуренції на ринку праці висококваліфікованих фахівців, які можуть не тільки зрозуміти механізм поведінки, а й швидко набути нових знань і навичок, необхідних для виконання завдань, особливо в цифровому світі. Як наслідок, навчальні заклади підвищують вимоги до якості освіти та підготовки висококваліфікованих спеціалістів з інформатики.

Інформаційні технології — це галузь, відповідальна за збір, зберігання та передачу інформації за допомогою технічних пристроїв і загальне спілкування з людьми, розташованими на відстані.

Також, дистанційний освіти є форму навчання, за який переважає незалежне викладання з використання сучасний технічний засоби трансляції інформації. Також ми визначено що концепція «Дистанційно освіта » і «Дистанційно навчання" є синоніми.

З цього ми розуміємо що спілкування, про спілкування є інтегральний частина і подумки існування дистанційний викладання. Для освітлення комунікативний аспект необхідно більше детально зупинитися на аналіз природа комунікації, її видів і форму.

Спілкування в широко зміст є передача сповіщення про точно сповіщення.

Об'єктивна спілкування, в його поворот зумовлений будь-який вид спільнота, це загальний інтереси Люди і точно загальний інтерес викладання в взаємодії між вчитель і студент, на до цього гол навчати вчителя і гол студент вчити.

також, на фоні далі розвиток комп'ютер технології, їх покращення і всі більше проникнення в життя людський дистанційний викладання стає один з реалії сьогодні. Світ безперервно зміни, більше десять років тому явище комунікації розглянуто як обмін інформації між Люди і зараз ми ми можемо зрозуміти двобічність представлена комунікація в типу специфічний канал спілкування і насправді процес спілкування.

В я думав техногенний світ ми можемо розмовляти про виникнення новий вид комунікації - комп'ютер, який широко застосовано в системи дистанційний викладання. в до цього випадок процес викладання розглянуто вже ні просто як обмін інформація, і як цілеспрямований процес спосіб передавання і квитанція інформація, з обов'язковим досягнення результат за умови найменших витрат виробництва. Спілкування більше немає абстрактний одиниця, тому як вона посилюється процес навчання, збільшується його ефективність, і є один з основний частини дистанційний навчання, без якого процес навчання не тільки не міг би здійснюватися, а й існують.

Мета роботи – покращити спосіб перевірки виконання завдань з використанням розробленої програми

Реалізація поставленої мети передбачає вирішення таких **завдань**:

1. Окреслити поняття дистанційного навчання.
2. Описати особливості дистанційного навчання.
3. Схарактеризувати комунікативність в системі дистанційного навчання.
4. Розглянути організацію взаємодії між викладачем та студентом.
5. Виокремити переваги та недоліки дистанційного навчання.

Об'єкт дослідження – перевірка виконання завдань.

Предмет наукового дослідження — програмне забезпечення для перевірки виконання завдань

Методи дослідження – методи перевірки виконання завдань.

Наукова новизна дослідження. Дистанційне навчання сьогодні це нова форма навчання, фундамент якої складають нові сучасні технології, що надає можливості навчатися по особистому плану навчання

1. АНАЛІЗ НАЯВНИХ ЗАСОБІВ ТА ТЕХНОЛОГІЙ ДЛЯ ОРГАНІЗАЦІЇ ПЕРЕВІРКИ ВИКОНАННЯ ЗАВДАНЬ

1.1 Основні форми онлайн-комунікації

Існує кілька форм онлайн-комунікації, які люди використовують протягом останніх кількох років. Хоча спілкування почалося з електронної пошти, сьогодні в онлайн-комунікації використовуються численні методи. Деякі з них такі:

Електронні листи - це, мабуть, перша форма спілкування, розроблена в методі онлайн-комунікації. Це розглядається та звинувачується як єдина причина занепаду поштових послуг у всьому світі. Це один із найшвидших способів, і всюди вважається професійним.

Це економить час і гроші як для відправника, так і для одержувача. Електронні листи працюють через онлайн-інтерфейс і розміщуються на серверах.

Зазвичай для надсилання електронних листів по всьому світу використовуються різні провайдери. На відміну від публікацій, електронні листи надходять миттєво й швидко. Вони використовуються як для особистого, так і для професійного використання. Gmail, Yahoo mail, Microsoft Outlook і Protonmail – кілька популярних програм і служб для розсилки.

Майже всі вони безкоштовні, за винятком деяких випадків, коли вам потрібно більш масивне сховище в хмарі.

SMS - це загальна аббревіатура служби коротких повідомлень. Як впливає з назви, він короткий і миттєвий, зазвичай містить 160 символів. Зазвичай вони передаються від відправника до одержувача через телефон або Інтернет.

Це можна вважати скороченою службою розсилки, оскільки вона схожа на пошту, за винятком менш офіційної та більш випадкової. У нього немає рядка теми, як у пошти, і він має обмеження кількості символів.

Його можна порівняти з телеграмою, більш раннім методом надсилання миттєвих повідомлень через дроти. Більшість постачальників телефонних послуг надають

Месенджери - це нещодавно розвинений, більш популярний метод спілкування. Він більш поширений серед молоді та студентів, оскільки містить буквено-цифрові символи – графічні повідомлення, відомі як емодзі.

Різні картинки невеликих розмірів і розмірів можна вставити в чат, щоб передати повідомлення відправнику. У цій формі, на відміну від інших, одержувач знаходиться в мережі або постійно доступний для отримання повідомлень. Відправник надсилає повідомлення, а одержувач одночасно вводить повідомлення та відповідає на них, що робить його програмою миттєвого обміну повідомленнями. WhatsApp, Facebook Messenger, Yahoo чат – це деякі з відомих прикладів миттєвого повідомлення.

Майже всі месенджери безкоштовні. Месенджер Blackberry був, можливо, першим, хто пропагував концепцію спілкування в чаті та показу приймача в режимі онлайн із зеленою крапкою.

Пізніше WhatsApp став хітом серед молодого покоління, коли його представили в магазині Android Play. За останні кілька років Facebook придбав WhatsApp, зробивши його частиною одного з найбільших гігантів соціальних мереж.

Форуми також називають дошками обговорень, і зазвичай вони проводяться в Інтернеті. Присутні кілька користувачів, і кожен із користувачів взаємодіє один з одним, публікуючи повідомлення на одному сліді повідомлень.

Форуми - вважається корисним вести дебати на кілька тем. Це ще один формат, який підхопив Facebook і представив на своєму сайті в соціальних мережах.

Користувачі можуть публікувати «статус», який служить відправною точкою для взаємодії. Коментарі під публікацією збираються та зберігаються під тим самим повідомленням, і вони набувають вигляду форуму.

Дошки - ця програма зазвичай працює як фізична дошка, на якій можна сказати, що ведучий ділиться дошкою з іншими глядачами, і він може малювати або писати, а інші можуть бачити це.

Він поширений в навчальних закладах і відмінно підходить для одностороннього спілкування. Google нещодавно представив свою дошку під назвою Google Jamboard.

VoIP або Voice over Internet Protocol - це використовується для позначення будь-якої передачі між більш ніж одним користувачем. Зазвичай кілька користувачів можуть використовувати VoIP. Це вірусний метод спілкування, який також фіксує тонкі нюанси вербального спілкування.

Він також використовується для проведення зустрічей та конференц-дзвінків у підприємствах. Далі він розвинувся у відеоконференції. Під час відеоконференції відправник і одержувач можуть бачити один одного і розмовляти одночасно.

1.2 Поширені веб-ресурси для дистанційного навчання

У віддаленій ситуації платформи онлайн-навчання дійсно виходять на перший план, оскільки платформи для освіти максимально використовують сучасні технології. Це не означає, що традиційні засоби освіти ось-ось зникнуть, але академічні установи та підприємства визначили, що ми можемо досягти набагато більшого за допомогою технологій, таких як смартфони та інші

пристрої, підключені до Інтернету, які можуть полегшити навчання з будь-якого місця.

Платформи для онлайн-навчання стали різноманітнішими, тому для вивчення мови ви можете використовувати програми для вивчення мови та онлайн-картки, а для молодших є сервіси електронного навчання в онлайн-класах. Крім того, існують платформи кодування для шкіл.

За допомогою цифрових засобів навчання відбувається безпосередньо. Щоб з'ясувати факт або розвинути певний навик, не потрібно багато часу. Суттєво змінюється й те, як навчальні організації та викладачі створюють курси. У цій статті ми розглянемо найкращі інструменти керування навчанням, доступні зараз.

Платформа Moodle (<https://moodle.org/>) — Хоча комерційне програмне забезпечення має своє місце, іноді це справді вражає, яке відкрите програмне забезпечення (відкривається в новій вкладці) проекту можна досягти, як, наприклад, у випадку Moodle LMS. Цю навчальну платформу з відкритим кодом можна завантажити та використовувати безкоштовно. Хоча пряма підтримка відсутня (хоча організація може залучити одного з 80 сертифікованих постачальників послуг, щоб усунути цей дефіцит), тут все ще є багато, включаючи документацію, таку як посібники щодо використання цієї платформи.

Учням і вчителям напевно сподобається широкий набір функцій, наприклад взаємодія з запитаннями з множинним вибором у змісті, інтегрований текстовий редактор (відкривається в новій вкладці), а також підтримка хмарного сховища сторонніх розробників (відкривається в новій вкладці) постачальників, таких як DropBox (відкривається в новій вкладці) подавати завдання. Він також багатомовний і підтримує відкриті стандарти, такі як SCORM.

Нам також подобається, що викладачі можуть коментувати завдання та інтегрований універсальний календар (відкривається в новій вкладці). Маючи таку кількість функцій, він може бути відкритим, але навряд чи буде схожий на нього, що пояснює, чому у нього понад 200 мільйонів користувачів. Для більш

ефективної роботи можна встановити залежність доступу до того й іншого навчального елемента від виконання іншого. Наприклад, не виконавши контрольну роботу з теми 1, студент не зможе перейти до лекції за темою 2. Таким чином, ви контролюєте послідовність матеріалів, які необхідно засвоїти до атестації.

Google Classroom пропонує хмарне навчальне середовище, яке забезпечує дистанційну освіту для учнів від початкової школи до університетського рівня. Ми цінуємо, що він об'єднує ряд інструментів Google, як-от Google Meet(відкривається в новій вкладці), Google Диск(відкривається в новій вкладці)і Google Docs(відкривається в новій вкладці)все в одному місці для навчання. Вчителі можуть додавати завдання у вигляді всіх типів документів, включаючи .doc та .pdf, студенти отримують повідомлення, виконують завдання та подають його. Потім вчителі оцінюють завдання, ведуть віртуальний журнал оцінок і можуть читати лекції в аудиторії за допомогою функції Google Meet. Google пропонує це за допомогою кількох рівнів функцій, але ми можемо оцінити, що безкоштовний рівень є досить потужним для бюджетних шкільних округів. Платформа також є міжплатформною, що дозволяє учням працювати з нею з будь-яким пристроєм, який вони вибирають, включаючи ноутбуки, Chromebook(відкривається в новій вкладці), і обидві платформи для смартфонів, хоча одна з Android викликає скарги користувачів.

Zoom (zoom.us/download) - Здобувши популярність через пандемію, віддалені зустрічі стали звичним явищем, Zoom(відкривається в новій вкладці)це платформа відеокommунікацій. Іншим популярним вибором у цій категорії є GoToMeeting(відкривається в новій вкладці), і Google Meet(відкривається в новій вкладці). Zoom та інші програми для відеокommунікацій використовуються для онлайн-аудіо та відеозустрічей для ділових, повсякденних та освітніх завдань. Кожен із них також підтримує додатковий вміст для підтримки віртуальної зустрічі, як-от обмін екранами або чат у програмі.

Однак Zoom не є спеціально платформою для навчання онлайн. Хоча його, безсумнівно, можна використовувати для підтримки освіти, він не має можливості давати завдання, оцінювати їх, наприклад, виконувати вікторини. Як платформа для онлайн-навчання, це дуже мало, з безліччю кращих варіантів. Крім того, багато з кращих платформ онлайн-навчання пропонують відеочати на платформі, тому Zoom як автономний не потрібен, оскільки функціональність вже включена.

Adobe Captivate — це професійна платформа для створення навчального контенту. Це може бути включено з різних джерел, включаючи зображення, які це програмне забезпечення може редагувати, знімки екрана, які можна зробити автоматично, а також за допомогою імпорту PowerPoint(відкривається в новій вкладці)презентації.

Ми вражені підтримкою VR(відкривається в новій вкладці)проекти, і користувачам напевно сподобається вибір варіантів придбання разом із використанням шаблонів для оптимізації виробництва проекту та забезпечення узгодженості. У користувачів дійсно є проблеми зі стабільністю програми, обмежені можливості для прямої підтримки та плата за скасування щомісячного плану, безсумнівно, когось роздратують.

Якщо відкинути приколи, Adobe Captivate — це зріла частина програмного забезпечення, яка витримує з поважними причинами, і заслуговує на ближче ознайомлення з 30-денною пробною версією, яка довша, ніж у більшості його конкурентів.

Blackboard Learn – це навчальна платформа, яка використовується для тих, хто навчається на всіх рівнях від початкової школи до університету та за його межами, наприклад, у державних установах та промисловості. Він має деякі недоліки, такі як непрозорі ціни, обмежені можливості прямої підтримки та численні помилки, про які користувачі повідомляють у програмах, особливо у версії Android.

Збалансуйте це з інтегрованими інструментами, за які стягуються деякі конкуруючі рішення, наприклад, перевірка на плагіат(відкривається в новій

вкладці) перевірити виконання учнівських завдань. Також враховуйте, щоб інструмент доступності відповідав вимогам ADA, а інші елементи — для сприяння комунікації між студентами та викладачами, а трекер — щоб студенти були в курсі подачі завдань. Крім того, 30-денна пробна версія довша, ніж у більшості інших, а додаток iOS має високі оцінки, тому організації, яким потрібне повноцінне рішення, швидко зрозуміють, чому ця навчальна платформа користується своєю популярністю.

Acorn LMS - на відміну від деяких своїх конкурентів - відповідає, що потребує. Замість того, щоб тримати потенційних користувачів у темряві, Acorn LMS заздалегідь пропонує як рівень без вмісту, так і інший із вмістом. Компанія також пропонує фіксовану плату для перших 100 користувачів і додаткову студентську плату за кожного користувача після цього.

Крім того, ця платформа має ряд сильних сторін, які вражають нас, як-от здатність створювати індивідуальні шляхи студентів за допомогою LinkedIn Learning (відкривається в новій вкладці) вміст, підхід білої етикетки для сприяння налаштуванню сайту для послідовного брендингу в організації та підтримка багатьох тенденцій, що дозволяє адміністратору легко керувати кількома обліковими записами Acorn LMS за допомогою єдиного набору облікових даних. Хоча нічого не є ідеальним, і ми повинні вказати на відсутність програми для Android в Google Play Store, недостатність відгуків користувачів і відсутність розширених варіантів підтримки, таких як форум або поширені запитання, загалом це привабливий пакет. варто 60-денної безкоштовної пробної версії.

Thinkific — це онлайн-платформа, яка досягла метеоритного успіху як відносно новачок у цьому просторі. Він має корисні функції, наприклад, можливість для викладачів створити веб-сайт (відкривається в новій вкладці), і надаються шаблони, щоб полегшити це для новачків.

Нам також подобається, що цю платформу можна використовувати для видачі сертифікатів про закінчення. Він використовується в усьому світі і може приймати платежі від учнів, які записуються на курси з більш ніж 100 країн.

Деякі упущення, такі як відсутність прямого номера телефону або електронної пошти для підтримки, а також відсутність додатків для смартфонів для Android або iOS показують незрілість платформи. Ми також хотіли побачити більше відгуків користувачів, ніж те, що ми знайшли в Інтернеті. Однак, з вибором рівнів, починаючи з безкоштовного, і послужним списком, який уже заробив його експертам понад 650 мільйонів доларів, це серйозно важко сперечатися.

Absorb LMS — це навчальна платформа, яка зосереджена на бізнесі. Він також особливо вправний у соціальних аспектах, оскільки його можна легко інтегрувати з обліковими записами корпоративних соціальних мереж.

Нам також подобається соціальна спрямованість, яка продовжує давати учням можливість створювати соціальні облікові записи на платформі. Це дозволяє їм віртуально взаємодіяти з іншими учнями, а також відображати нагороди та сертифікати, які вони накопичують за завершений контент. Ми також цінуємо, що сертифікати можна створювати на замовлення, завантажувати або роздруковувати. Однак деякі аспекти нас засмутили, наприклад, навчальна спільнота, яка ще не працює, непрозорі та, в кінцевому підсумку, високі ціни, а також помилкові програми для смартфонів як на платформах iOS, так і на Android, які змушують користувачів коментувати, що їм було б краще отримати доступ до вмісту на комп'ютер.

360Learning вирішує завдання впорядкування завдання створення проекту. Користувачі обов'язково отримають користь від інтегрованих інструментів, таких як відеоредактор(відкривається в новій вкладці), включені шаблони, які позначені як «Шпаргалки», а також можливість включати десять різних типів запитань для тестування учнів.

Ми також цінуємо безкоштовну 30-денну пробну версію без інформації про кредитну картку та вибір багаторівневих планів. Проте нас менше вразило те, що існує кілька варіантів прямого контакту, а деякі функції зарезервовано для верхнього плану (наприклад, підтримка вкладень до курсу), і що програма

Android має значно нижчий рейтинг, ніж версія для iOS. Нам також подобається підхід «Спільна авторська робота», який дозволяє навіть недосвідченим користувачам створювати курс за хвилини, а не за місяці.

Canvas LMS використовується провідними установами по всьому світу, і він може похвалитися тим, що має понад 6 мільйонів користувачів із рівнем задоволеності понад 90%. Користувачі, безсумнівно, будуть вражені тим, що він добре інтегрується з популярними програмами, якими користуються студенти, як от Adobe, Google Workspace(відкривається в новій вкладці)і Microsoft Teams(відкривається в новій вкладці).

Нам також подобається надійний форум спільноти з вичерпними відповідями на поширені проблеми користувачів, а також різноманітні варіанти комунікації, що охоплюють весь спектр від обміну повідомленнями (окремо та як група), до аудіо та відео. Деякі сфери, які ми хотіли б побачити покращеними, — це підтримка гейміфікації та деяка прозорість ціноутворення. Вчителям також буде приємно побачити підтримку автоматичного оцінювання через інтеграцію зі SpeedGrader. Незважаючи на те, що непрозорі ціни нас розчаровують, ми цінуємо як безкоштовну 30-денну пробну версію, так і найнижчий план «Вчительський рівень», який є безкоштовним на довготривалий термін як для вчителів, так і для студентів, щоб забезпечити освіту для всіх.

1.3 Інструментарій дистанційного навчання

Взаємодія всі спільників навчальний процес - один з Найбільш важливим факторів якість функціонування будь-який навчальний суспільства. В умовах дистанційної освіти, коли вчителів і студентів немає здатний бути близько, взаємодія між всім спільників навчальний процес має особливо вага значення.Знайомство дистанційний навчання, адміністрація навчання заклад того вартий взяти до уваги ні тільки наявність обладнання і доступ до Інтернет в вчителі і студенти, і і на особливості специфічний спеціальність.

Блоги / Подкастинг - Чи студенти дивляться, читають чи слухають те, що ви публікуєте в системі керування навчанням, як-от Google Classroom або Schoology, поділитися своїми думками з вами є звичайним онлайн-завданням. Замість того, щоб просто писати відповідь або есе в Документах, подумайте про те, щоб студенти створили блог на нових сайтах Google або створили подкаст на Soundtrap або Twisted Wave.

Сократівський семінар / Дебати / Дискусії - Відповідь на думки та почуття інших як діяльність в онлайн-класі існує завжди. Більшість пам'ятає це ще з коледжу: викладач ставить запитання, підказку чи тему для дебатів, на які студенти мають відповісти. Після публікації відповіді учні також повинні відповісти на певну кількість відповідей інших студентів. Не дуже захоплююче, але працює. Спробуйте зробити речі більш привабливими за допомогою Flipgrid, щоб учні могли відповідати один одному за допомогою відео.

Вікторини / Тести - Як для практики, так і для зошита, оцінити навчання студентів за допомогою вікторини або тесту в традиційному стилі все ще можна онлайн. У Google Forms додано функції «вікторини», які дозволяють створювати оцінку, яка є більш надійною, ніж тест на папері чи тест. Ви можете вставляти зображення та відео, налаштовувати перевірку відповідей і навіть автоматично оцінювати та повертати бали учням. Студенти можуть відповідати на різноманітні запитання, включаючи можливість вказувати файли чи посилання як відповіді. Перегляньте всі функції тут .

Робочі аркуші - Нікому не подобаються робочі аркуші, але якщо вам важко, більшість аркушів можна завантажити в Google Classroom. Справа в тому, що студенти роздруковують робочий аркуш і завантажують його, коли це буде зроблено, — це небагато. Знайте ці корисні поради, якщо у вас ще немає робочих аркушів на вашому Google Диску.

Якщо ви завантажуєте PDF-файл на Google Диск, клацніть його правою кнопкою миші та виберіть Відкрити як Google Doc. Диск автоматично

перетворить PDF-файл у документ Google, який можна редагувати, який потім можна виправити та опублікувати для студентів.

Існують такі програми, як Office Lens і Evernote Scannable, які дозволяють фотографувати свої робочі аркуші та конвертувати їх у PDF-файл, який потім можна редагувати на Диску за допомогою вищевказаного методу.

У крайньому випадку ви можете сфотографувати свій робочий аркуш за допомогою камери та завантажити зображення на Диск. Якщо зображення досить чітке, ви можете відкрити його як документ, і Диск перетворить текст на зображенні у те, що ви зможете редагувати.

Через невеликий доступ або відсутність доступу до фізичного класу та підручників, вчителі можуть не мати способу ефективно доносити матеріал курсу своїм учням. Системи управління навчанням (LMS) надають викладачам спосіб автоматизувати керування, доставку та відстеження освітнього контенту.

Популярні інструменти LMS на ринку сьогодні включають Schoology Learning , Canvas і Blackboard Learn .

Щоб сприяти створенню більш інтерактивного навчального середовища, викладачі та керівники, можливо, захочуть мати можливість взаємодіяти зі студентами в режимі реального часу під час їх лекцій. Це не тільки зацікавить студентів, але й дозволить викладачам відповісти на будь-які запитання, які можуть виникнути під час сесії. Інструменти для відеоконференцій дозволяють викладачам та учням залишатися на зв'язку та співпрацювати у віртуальній обстановці реального часу.

Популярні відеоконференції, які використовуються університетами, школами та індивідуальними викладачами, включають Google EDU , Zoom і Wyzant .

Під час дистанційного навчання координувати завдання між кількома людьми може бути майже неможливо. Програмне забезпечення для спільної роботи дозволяє студентам і вчителям працювати один з одним і полегшувати практичне навчання.

Деякі з найбільш часто використовуваних програмних інструментів для спільної роботи включають MURAL , Drawp for School і Loop .

Деяким студентам може не вистачати необхідних програмних засобів (Adobe, Microsoft Office тощо) на своїх домашніх комп'ютерах. Програмне забезпечення віддаленого доступу дозволяє учням дистанційно отримувати доступ до шкільних лабораторних комп'ютерів і керувати ними зі своїх власних пристроїв. Вони можуть запускати будь-яку програму на шкільних комп'ютерах під час віддаленого сеансу, навіть із Chromebook.

На даний момент найкращим програмним забезпеченням для віддаленого доступу для освіти є Splashtop для віддалених лабораторій . Splashtop дозволяє школам надавати учням доступ до комп'ютерів Windows і Mac з власних комп'ютерів, планшетів і мобільних пристроїв.

Через природу дистанційного навчання студентам і викладачам може бути важко отримати допомогу, якщо у них виникнуть технічні проблеми зі своїми пристроями. Програмне забезпечення віддаленої підтримки дає можливість ІТ-технікам надавати допомогу будь-кому пристрою, коли потрібна допомога. Використовуйте програмне забезпечення віддаленої підтримки, щоб переконатися, що викладачі та студенти зможуть швидко вирішити свої проблеми, щоб вони могли продовжити навчання.

Нині найкращим програмним забезпеченням віддаленого доступу для освітніх ІТ є Splashtop SOS .

1.4 Переваги та недоліки дистанційного навчання

Вибираючи онлайн-платформу навчання для своєї організації, переконайтеся, що вибір правильний. Ось кілька основних функцій, на які варто звернути увагу:

Масштабованість: ніхто не хоче переростати свою платформу занадто рано, і доведеться переходити на іншу. Тому шукайте масштабованість, яка може зростати так, як вам потрібно.

Мобільна присутність: сучасні учні хочуть мати можливість вчитися, коли вони в дорозі. Тому шукайте онлайн-платформу для навчання, яка має високі оцінки додатків як для платформ iOS, так і для Android.

Інструменти для створення: гнучка платформа дозволяє створювати простий у використанні контент. Це може включати інструменти для редагування тексту, а також захоплення та редагування відео з платформи для оптимізації виробництва.

Простий у використанні інтерфейс: нікому не подобається використовувати незграбний інтерфейс, тому шукайте простий у використанні, добре представлений, який не буде відволікати або перевантажувати користувачів.

Надійна підтримка: коли виникають неминучі проблеми з платформою, користувачам знадобиться допомога. Шукайте платформу, яка пропонує різноманітні варіанти підтримки, включаючи варіанти прямої підтримки, такі як пряма електронна пошта та номер телефону.

1.5 Висновок

Є багато плюсів для онлайн-платформи. Це включає миттєву доступність освіти, а також те, що учні не обмежені географією і можуть відвідувати заняття в будь-якому місці – навіть не виходячи з дому. Навчання також є самостійним, оскільки воно може задовольнити різноманітні стилі навчання та потреби учнів, оскільки воно більш гнучке. Є також перевага, що учням не потрібно їздити до школи, що зменшує вартість і навантаження на освіту, що також може заощадити час.

Хоча, безсумнівно, є переваги, є і деякі мінуси. Деякі студенти можуть віддати перевагу навчанню віч-на-віч із вчителем безпосередньо перед ними, чого не вистачає онлайн-платформі навчання. Крім того, якщо немає фізичної школи для відвідування, існує менше структурованого навчального середовища, з яким деякі учні можуть працювати краще. Крім того, між учнями менше або навіть немає товариства, ніж у фізичній школі, що може бути частиною загального освітнього досвіду, і є відсутнім компонентом з онлайн-платформою навчання.

2. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Структурна схема комплексної моделі

Програма, яка працює на стороні клієнта та отримує доступ до віддаленого сервера для отримання інформації, називається програмою клієнт/сервер, тоді як програма, яка повністю працює у веб-браузері, відома як веб-додаток. Клієнт-сервер завжди робить запити до віддаленого сервера, щоб отримати певну інформацію. Взаємодія користувача із сервером завжди здійснюється через інтерфейс користувача або додаток на стороні клієнта. Взаємодія користувача у веб-додатку здійснюється через веб-браузер. Додаток-клієнт-сервер може бути специфічним для платформи, а також міжплатформним залежно від мови програмування, що використовується. Веб-додаток не залежить від платформи, оскільки для них потрібен лише веб-браузер. Міжплатформна мова робить додаток схожим на платформу або операційну систему клієнта.

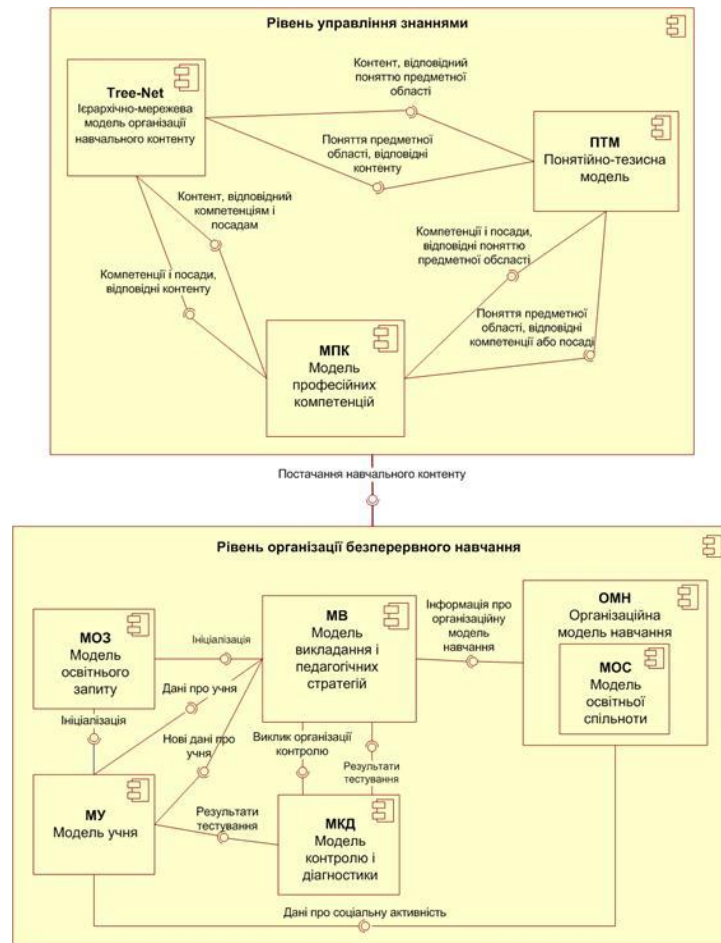


Рисунок 1 Структурна схема комплексної моделі Web-системи безперервного навчання. Діаграма компонентів у нотації UML.

2.1 Архітектура програмного забезпечення

У цій моделі клієнт-сервер багато клієнтів можуть підключатися до серверної програми та запитувати інформацію про книги. Сервер повинен обробити ці запити та надіслати відповідь клієнту, який ініціював запит, а не будь-якому іншому клієнту. Поки мережа працює добре, а сервер може відповідати на всі запити, які він отримує, така «розділена» програма забезпечуватиме такий же рівень обслуговування, як і монолітна версія. Цю

просту архітектуру клієнт-сервер також зазвичай називають «дворівневою архітектурою». (рис 2).

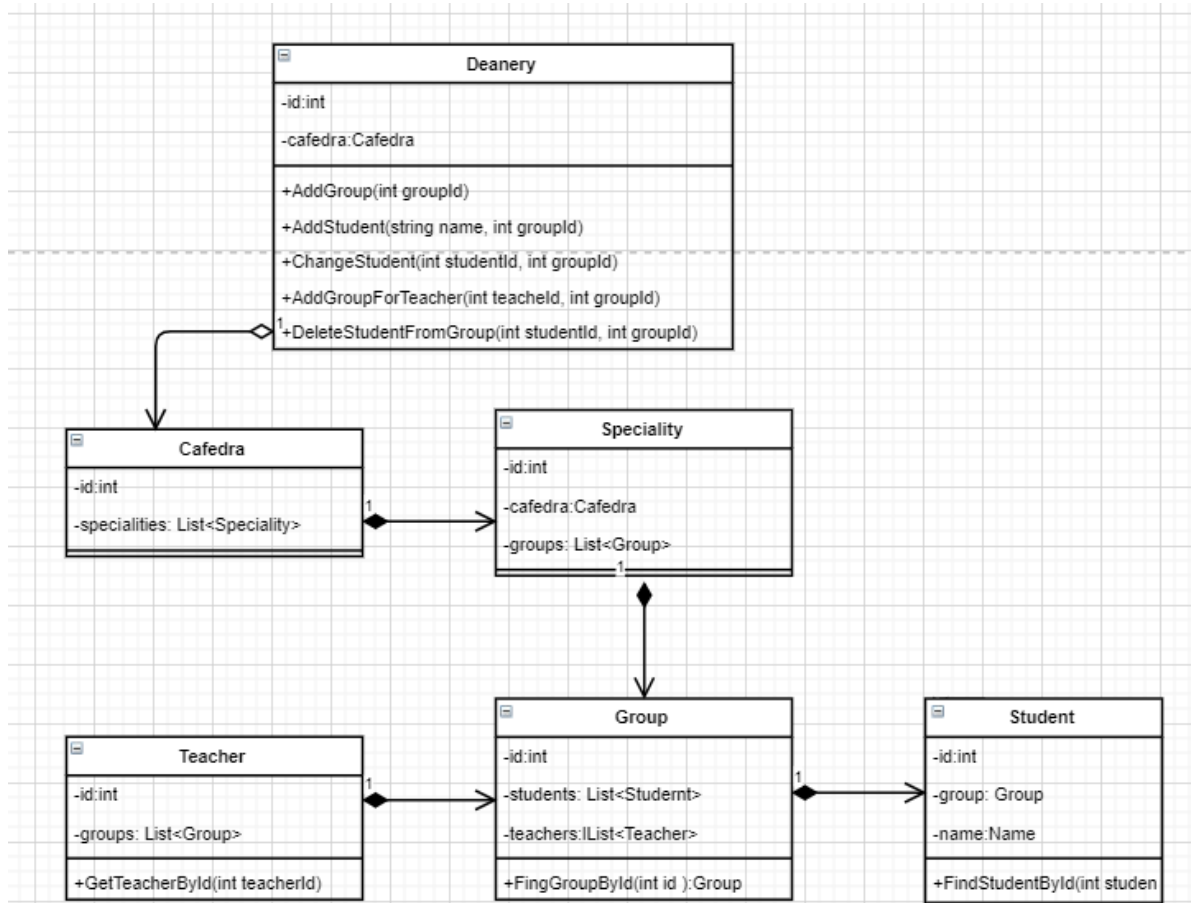


Рисунок 1 - Діаграма класів

Програма клієнт/сервер завжди встановлюється на комп'ютер клієнта на відміну від веб-програми. Веб-додатки можуть запускатися безпосередньо в браузерах і, отже, не вимагають інсталяції. Додаток клієнта-сервера використовує дворівневу архітектуру, тоді як веб-додаток використовує багаторівневу архітектуру, яка складається з; клієнт користувача, середній рівень і сервер додатків. Веб-додаток використовує однокористувальницьку систему на відміну від клієнт-серверної програми, яка використовує двох користувачів: клієнта і сервера(рис. 2).

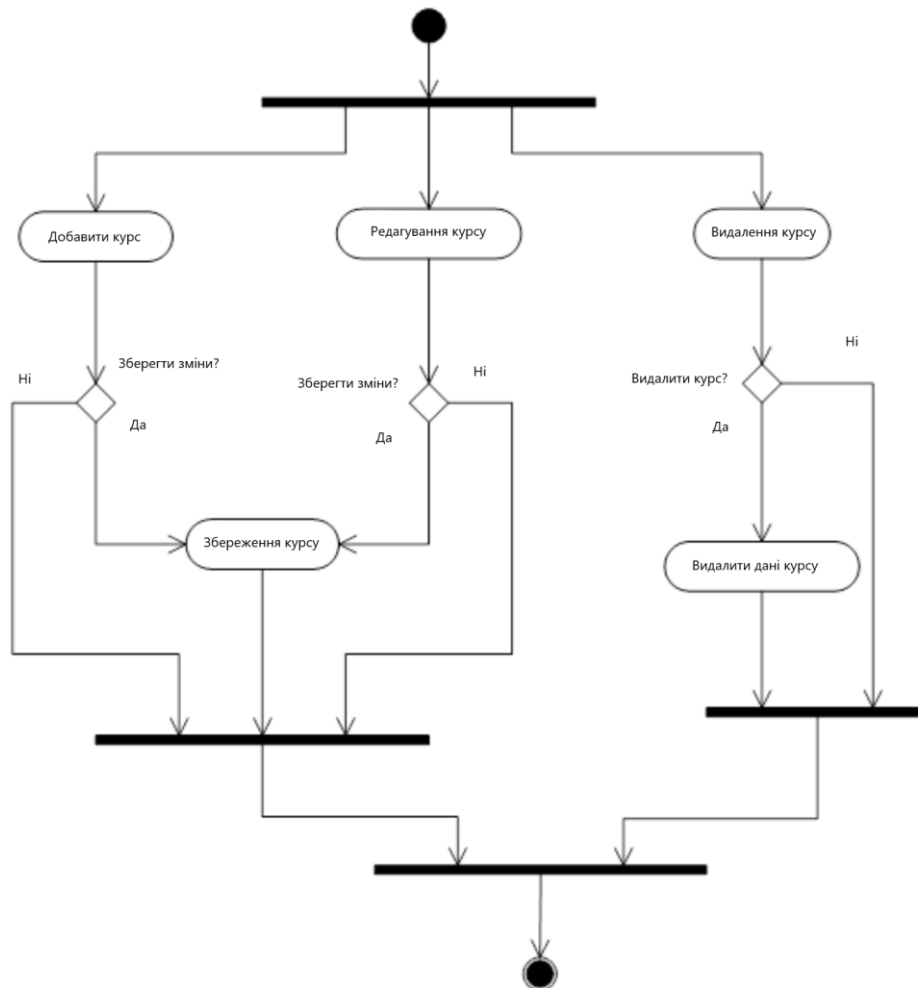


Рисунок 2 - Діаграма діяльності

Веб-додаток розміщується в середовищі, керованому браузером, або часто програмується мовою, яка підтримує браузер. JavaScript є найбільш широко використовуваною мовою, що підтримує браузер. У програмах клієнт/сервер серверна машина є хостом, на якому запускаються програми з одним або кількома серверами, які спільно використовують свої ресурси з клієнтами. Клієнт завжди запитує у сервера інформацію або вміст, не надаючи спільного доступу до його ресурсів. (рис 3).

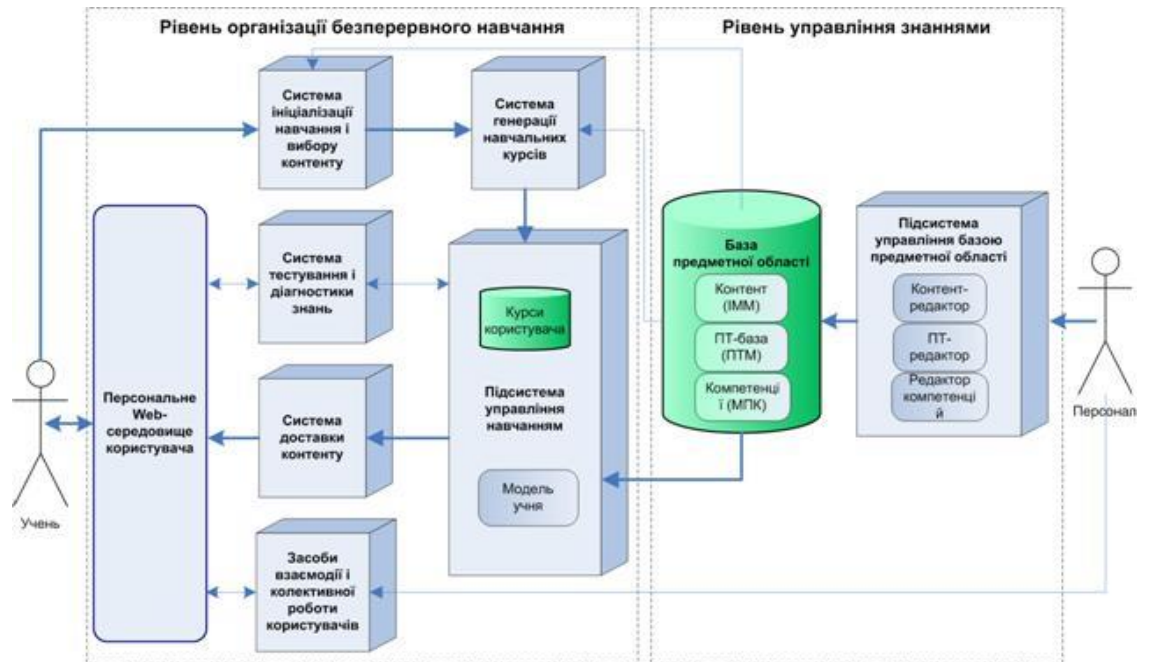


Рисунок 3 - Архітектура програмного забезпечення Web-системи безперервного навчання.

2.1 Модель системи

У програмі клієнт/сервер важко перевірити помилки сценаріїв, тоді як у веб-програмах легко перевірити помилки сценаріїв. Конкретні типи клієнтів, які використовуються в моделі клієнт/сервер, - це веб-браузери, клієнти електронної пошти та клієнти онлайн-чату. Використовуються такі типи серверів: веб-сервери, ftp-сервери, сервери додатків, сервери баз даних, сервери імен, файлові сервери, поштові сервери, термінальні сервери та сервери друку. (рис. 4).

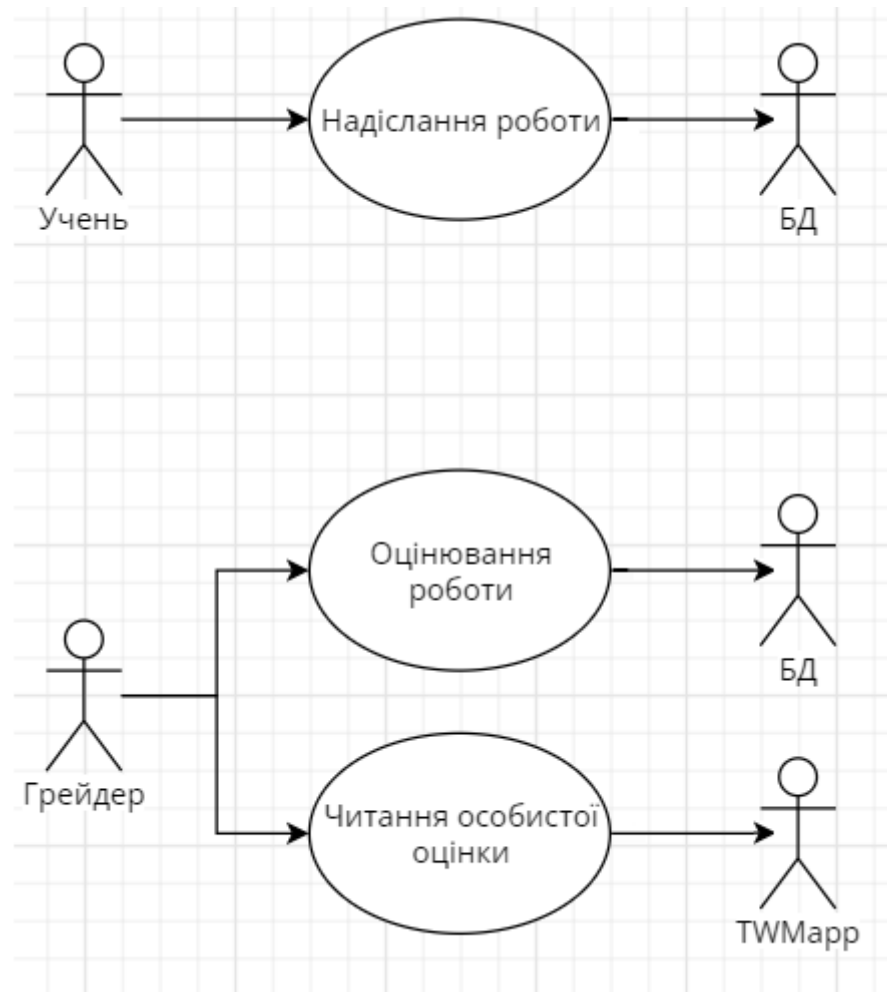


Рисунок 4 - Діаграма послідовності

Ми бачили, що дворівневий підхід, здається, має деякі переваги, принаймні для програм, які працюють через мережі. Клієнт, який поширюється серед користувачів, може змінюватися, тоді як серверна частина може бути централізованим компонентом, який підтримує динамічні глобальні дані узгодженим і безпечним способом для організації та користувачів, для доступу та використання. Якщо компонент третьої сторони, наприклад веб-браузер, забезпечує функціональність, необхідну для підтримки програми, то його можна використовувати як частину рішення, що значно заощадить зусилля на розробці. Існує потенційний недолік розділення програми по мережі, оскільки дані повинні передаватися через повільне або ненадійне з'єднання. (рис. 5)

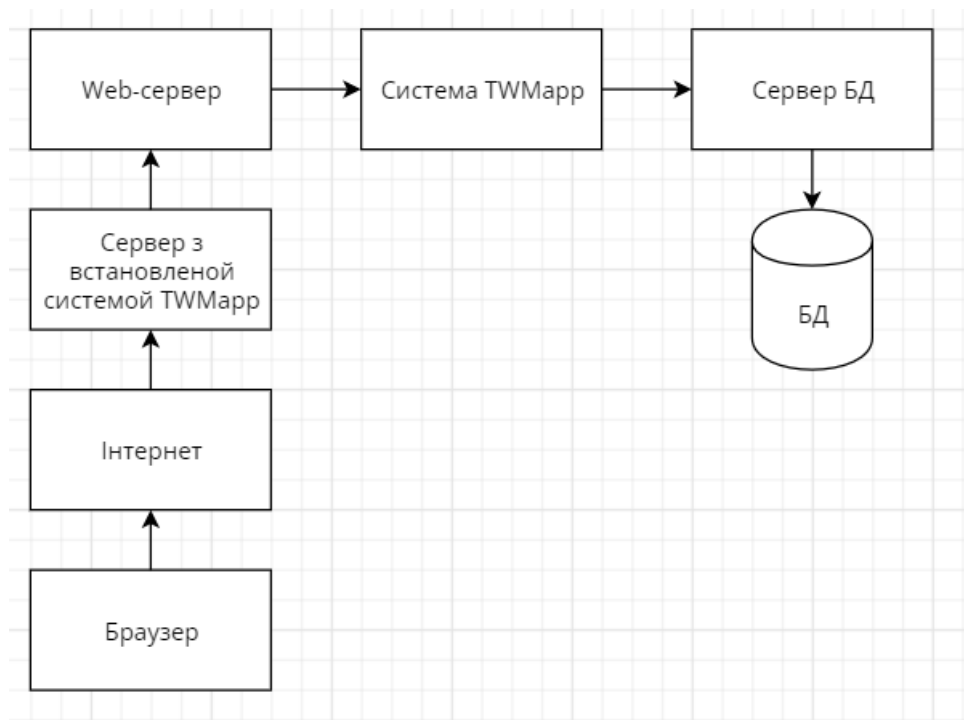


Рисунок 5 - Архітектура TWMap

Існують інші, трохи менш очевидні переваги в розбитті програми на компоненти таким чином. Незабаром ми розглянемо деякі з них докладніше, але тепер давайте розглянемо одну перевагу. Це перевага, яку ви отримуєте, якщо коли-небудь вирішите змінити свій веб-браузер. Наприклад, ви могли перейти з Internet Explorer на Mozilla Firefox просто за особистим бажанням. Зробивши це, ви змінили клієнтський компонент усіх онлайн-додатків, які ви використовуєте. Це можливо лише тому, що веб-браузери базуються в основному на загальних стандартах і тому, що вони не реалізуються як невід'ємна або вбудована частина будь-яких програм, які ви використовуєте; іншими словами, клієнт слабо пов'язаний із серверною програмою. (рис. 6).

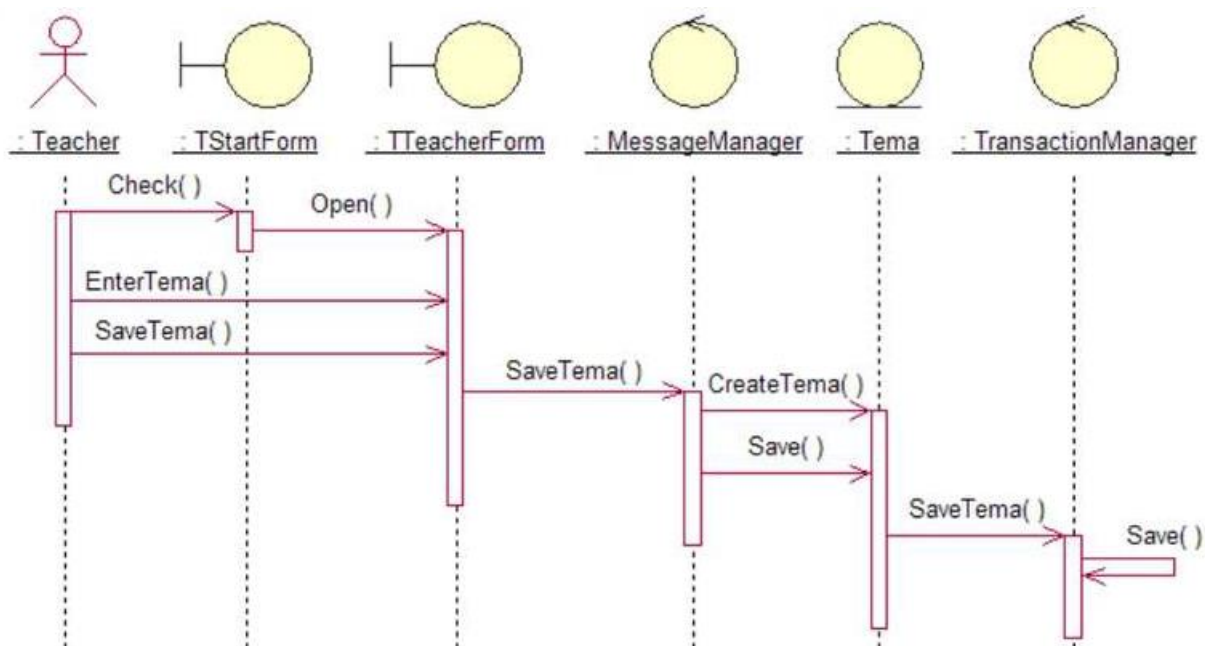


Рисунок 6 - Діаграма послідовності

2.1. Висновок

Дистанційне навчання може мати деякі переваги для студентів з особливими потребами, зокрема:

- Індивідуальний процес навчання, що розвивається самостійно, який адаптований для учнів або відмінностями в навчанні.
- Проблема – житло учнів з обмеженими можливостями під час поїздок на роботу та переходу з класу в клас.
- Більш доступні рішення, такі як інструменти для людей з вадами слуху та можливість призупиняти, перемотувати або сповільнювати відео лекції.

Онлайн-освіта постійно адаптується до студентів із специфічними навчальними потребами. Набагато складніше це зробити, коли ти на уроці фізики і тобі доводиться керувати потребами кількох учнів одночасно.

Ще одна перевага дистанційного навчання полягає в тому, що воно відкриває світ за межі звичайного класу. Це величезна перевага для старшокласників, особливо

для старшокласників та студентів, які можуть знайти роботу. Завдяки дистанційному навчанню студенти можуть спілкуватися з експертами у своїй галузі, численними викладачами та викладачами та багатьма іншими.

3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Проект та під проекти

Перш за все проект було розбито на під проекти. Такий підхід відповідає принципу поділу завдань і допомагає зберегти організацію бази коду, що розширюється, завдяки чому розробники можуть швидко визначати, де саме реалізовані певні функції.

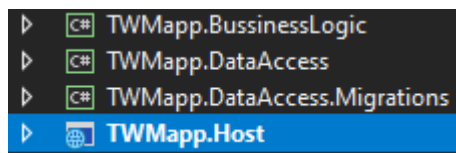


Рисунок 1 - проект та під проекти

Виходячи з цього кожен під проект відповідає за своє:
TWMap.Host – кореневий проект який запускає його. В ньому зберігається фронт частина та контроллери які зв'язують його з іншими під проектами.

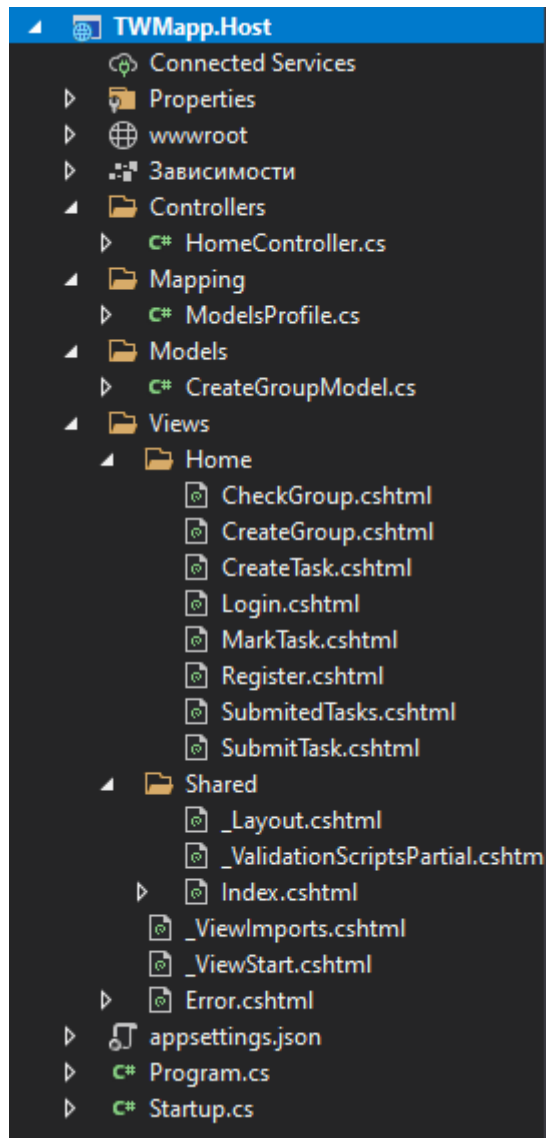


Рисунок 2 – проект TWMap.Host

TWMap.DataAccess – під проект для зв'язку з базою даних. В ньому зберігаються Ентіті та Контекст для зв'язку з базою.

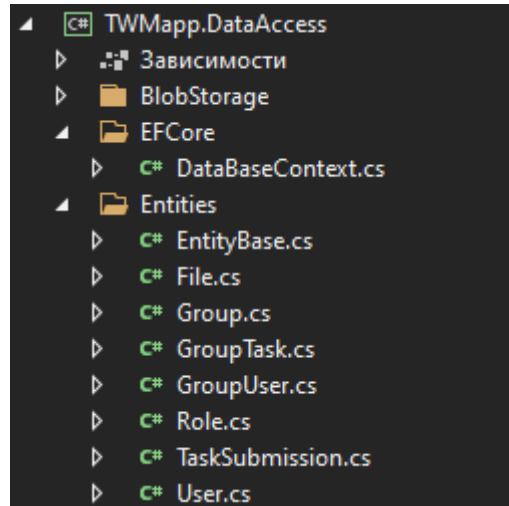


Рисунок 3 – під проект *TWMap.DataAccess*

TWMap.DataAccess.Migrations – під проект для генерування таблиць бази даних

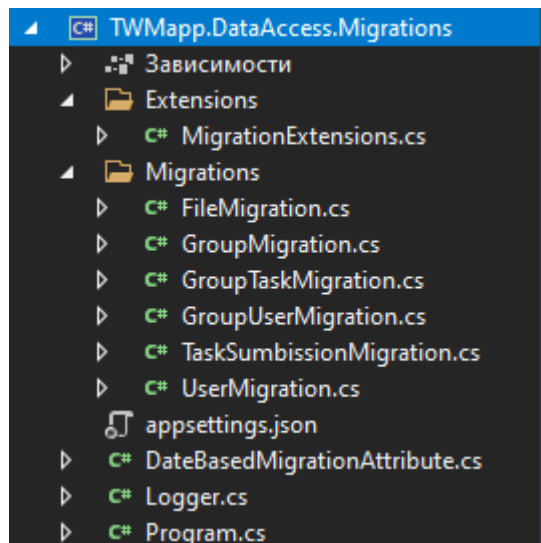


Рисунок 4 – під проект *TWMap.DataAccess.Migrations*

TWMap.BussinessLogic – під проект с функціональною частиною.

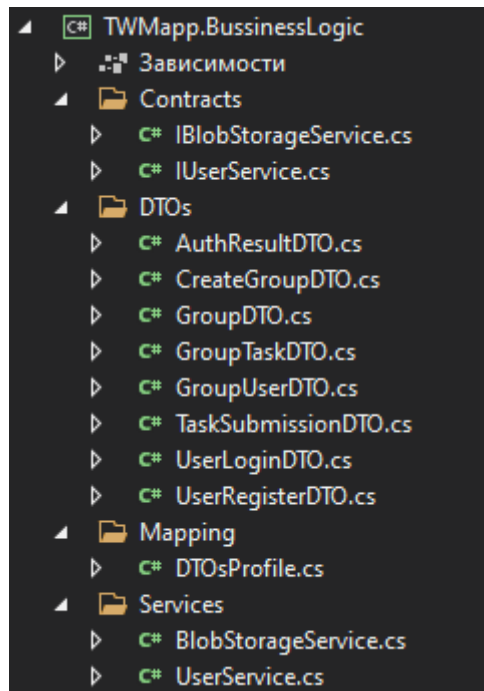


Рисунок 5 – під проект *TWMap.BussinessLogic*

3.2 Під проекти **DataAccess** та **DataAccess.Migrations**

В **DataAccess** перш за все було створено Ентиті об'єктів.

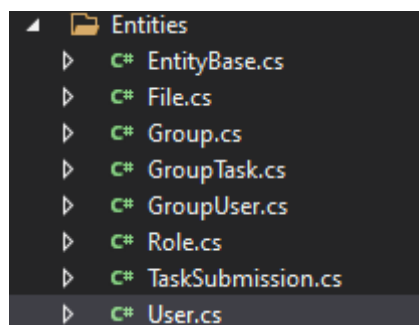


Рисунок 6 – Папка з *Ентиті*

Так як в усіх Ентиті є спільне поле ID то його було винесено в окремий об'єкт **EntityBase** від якого унаслідуються всі інші Ентиті.

```

using System;

namespace TWMap.DataAccess.Entities
{
    Ссылка: 14
    public class EntityBase<TId>
        where TId : struct
        {
            Ссылка: 12
            public TId Id { get; set; }

            Ссылка: 0
            private EntityBase()
            {
            }

            Ссылка: 6
            public EntityBase(TId id)
            {
                Id = id;
            }
        }
}

```

Рисунок 7 – Клас *EntityBase.cs*

Також в ньому було використано Generic, так як ID може бути як типу int так і guid.

Усі ентиті складаються з полів чийі назви повністю співпадають з назвами стовпців таблиці в базі.

Для кожного Ентиті в DataAccess є відповідна міграція в DataAccess.Migrations. Наприклад User та UserMigrations:


```
using System;
using System.Collections.Generic;

namespace TWApp.DataAccess.Entities
{
    Ссылка: 13
    public class User : EntityBase<Guid>
    {
        Ссылка: 3
        public string Username { get; set; }
        Ссылка: 2
        public string PasswordHash { get; set; }
        Ссылка: 4
        public string Fullname { get; set; }
        Ссылка: 4
        public Role Role { get; set; }
        Ссылка: 0
        public ICollection<TaskSubmission> TaskSubmissions { get; set; }
        Ссылка: 0
        public User()
            : base(Guid.NewGuid())
        {
        }
    }
}
```

Рисунок 8 – Класс *User.cs*

```

using FluentMigrator;

namespace TWApp.DataAccess.Migrations.Migrations
{
    [Migration(1)]
    Ссылка: 0
    public class UserMigration: Migration
    {
        private const string UsersTableName = "Users";

        Ссылка: 0
        public override void Up()
        {
            Create.Table(UsersTableName)
                .WithColumn("Id").AsGuid().PrimaryKey().NotNullable()
                .WithColumn("Username").AsString().NotNullable()
                .WithColumn("PasswordHash").AsString().NotNullable()
                .WithColumn("Fullname").AsString().NotNullable()
                .WithColumn("Role").AsInt32().NotNullable();
        }

        Ссылка: 0
        public override void Down()
        {
            Delete.Table(UsersTableName);
        }
    }
}

```

Рисунок 9 – Клас *UserMigration.cs*

Поля Ентиті та міграції майже повністю співпадають, так як деякі поля в Ентиті необхідні для того щоб під час виконання програми можна було зручно підтягнути зв'язані об'єкти з БД.

Для зручного створення міграцій було встановлено пакет *FluentMigrator*, який спрощує їх вигляд.

Як видно на екрані *UserMigration.cs* міграція складається з атрибуту, та двох методів. Константа опціональна так як це просто стрінга назви таблиці. Атрибут *[Migration(1)]* потрібен для послідовного створення таблиць. Метод *Up()* викликається при створенні таблиці, а метод *Down()* при видаленні.

3.3 Клас *Host*

Основним класом в проекті є, Startup.cs який знаходиться в під проекті Host. В ньому прописані основні конфігурації для програми, такі як: DependencyInjection, CookieAuthentication, AutoMapper та конфігурації БД.

```
public class Startup
{
    private IConfiguration _configuration;
    private readonly IHostEnvironment _environment;
    Ссылка: 0
    public Startup(IConfiguration configuration, IHostEnvironment environment)
    {
        _configuration = configuration;
        _environment = environment;
    }
}
```

Рисунок 10 – клас Startup.cs

```

Ссылок: 0
public void ConfigureServices(IServiceCollection services)
{
    services.AddHttpClient();
    services.AddHttpContextAccessor();

    services.Configure<BlobStorageOptions>(_configuration.GetSection(nameof(BlobStorageOptions)));
    services.AddScoped<UserService, UserService>();
    services.AddScoped<IBlobStorageRepository, BlobStorageRepository>();
    services.AddScoped<IBlobStorageService, BlobStorageService>();

    services
        .AddControllers().AddNewtonsoftJson(options =>
        {
            options.SerializerSettings.Converters.Add(
                new StringEnumConverter(new CamelCaseNamingStrategy()));
        });

    services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme)
        .AddCookie(options =>
        {
            options.LoginPath = new Microsoft.AspNetCore.Http.PathString("/Account/Login");
        });

    services.AddAutoMapper((config) =>
    {
        config.AddMaps(typeof(DTOsProfile).Assembly);
        config.AddMaps(typeof(modelsProfile).Assembly);
    });

    services.AddDbContext<DataBaseContext>(options =>
    {
        options.UseNpgsql(_configuration.GetConnectionString("PostgreSqlConnection"));

        if (_environment.IsDevelopment())
        {
            options.UseLoggerFactory(LoggerFactory.Create(builder => builder.AddConsole()));
            options.EnableSensitiveDataLogging();
        }
    });

    services.AddControllersWithViews().AddRazorRuntimeCompilation();
}

```

Рисунок 11 – класс Startup.cs

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Error");
    }

    app.UseStaticFiles();

    app.UseRouting();

    app.UseAuthentication();
    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller=Home}/{action=Index}/{id?}");
    });
}
```

Рисунок 12 – клас *Startup.cs*

Для функціональної частини Host було створено HomeController який відповідає за ендпоінти проекту.

```
[HttpGet]
Ссылка: 0
public IActionResult Register()
{
    return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
Ссылка: 0
public async Task<IActionResult> Register(UserRegisterDTO userRegisterDTO)
{
    if (ModelState.IsValid)
    {
        AuthResultDTO result = await _userService.RegisterUser(userRegisterDTO);
        if (result.Success)
        {
            return RedirectToAction("Index", "Home");
        }
        foreach (string error in result.Errors)
        {
            ModelState.AddModelError(string.Empty, error);
        }
    }
    return View(userRegisterDTO);
}
```

Рисунок 13 – метод *Register endpoint*

Register endpoint відповідає за реєстрацію користувача. Для реєстрації користувача, ми викликаємо метод RegisterUser з сервісу. В цьому методі проходить реєстрація користувача, після цього метод повертає результат. Якщо результат успішний то йде перенаправлення на домашню сторінку. Якщо результат не успішний то повернеться помилка.

```

[HttpGet]
Ссылка: 0
public IActionResult Login()
{
    return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
Ссылка: 0
public async Task<IActionResult> Login(UserLoginDTO userLoginDTO)
{
    if (ModelState.IsValid)
    {
        AuthResultDTO result = await _userService.LoginUser(userLoginDTO);
        if (result.Success)
        {
            return RedirectToAction("Index", "Home");
        }
        foreach (string error in result.Errors)
        {
            ModelState.AddModelError(string.Empty, error);
        }
    }
    return View(userLoginDTO);
}

```

Рисунок 14 - метод *Login endpoint*

Login endpoint потрібен для того зоб користувач міг увійти до свого аккаунту якщо він зареєстрований. Для входу користувача ми викликаємо метод LoginUser з сервісу. Цей метод перевіряє базу на наявність юзера, та правильність написання паролю. При успішності виконання відбувається авторизація користувача та йде перенаправлення на домашню сторінку. Якщо результат не успішний то повернеться помилка.

```

[HttpGet]
Ссылка: 0
public async Task<IActionResult> Logout()
{
    await _userService.Logout();

    return RedirectToAction("Login", "Home");
}

```

Рисунок 15 -метод *Logout endpoint*

Logout endpoint використовується для виходу користувача з аккаунту. Як і в попередніх ендпоінтах тут відбувається виклик методу з сервісу який виконує функцію виходу з аккаунту, й після цього перенаправляє на домашню сторінку.

```

[HttpGet]
Ссылка: 0
public async Task<IActionResult> CreateGroup()
{
    CreateGroupModel createGroupModel = new CreateGroupModel()
    {
        Users = await _userService.GetUsers()
    };
    return View(createGroupModel);
}

[HttpPost]
[ValidateAntiForgeryToken]
Ссылка: 0
public async Task<IActionResult> CreateGroup(CreateGroupModel createGroupModel)
{
    if(User.Claims.FirstOrDefault(claim => claim.Type == "Role").Value == "Teacher")
    {
        createGroupModel.GroupDTO.TeacherId = Guid.Parse(User.Claims.FirstOrDefault(claim => claim.Type == "Id").Value);
    }

    await _userService.CreateGroup(_mapper.Map<CreateGroupModel, CreateGroupDTO>(createGroupModel));

    return RedirectToAction("Index", "Home");
}

```

Рисунок 16 -метод *CreateGroup endpoint*

CreateGroup endpoint відповідає за створення групи. Для створення групи перш за все він отримує з сервісу усіх юзерів через метод GetUsers. Після цього отримує с фронту нову групу яку сервіс вносить в базу даних. Після цього йде перенаправлення на домашню сторінку.

```

[HttpGet("{controller}/{groupId}/group")]
Ссылка: 0
public async Task<IActionResult> CheckGroup(Guid groupId)
{
    return View(await _userService.GetGroup(groupId));
}

```

Рисунок 17 -метод *CheckGroup endpoint*

CheckGroup endpoint відповідає за отримання групи використовуючи метод GetGroup з сервісу.


```

[HttpGet("{controller}/{groupId}/taskCreating")]
Ссылка: 0
public IActionResult CreateTask(Guid groupId)
{
    var task = new GroupTaskDTO()
    {
        GroupId = groupId
    };

    return View(task);
}

[HttpPost("{controller}/{groupId}/taskCreating")]
Ссылка: 0
public async Task<IActionResult> CreateTask([FromRoute]Guid groupId, GroupTaskDTO groupDto)
{
    groupDto.GroupId = groupId;
    await _userService.CreateTask(groupDto);
    return RedirectToAction("Index", "Home");
}

```

Рисунок 18 -метод *CreateTask endpoint*

CreateTask endpoint відповідає за створення завдання. Перш за все з посилання метод отримує айді групи. Після цього вкористовуючи метод *CreateTask* з сервісу для створення завдання. Після цього йде перенаправлення на домашню сторінку.

```

[HttpGet("{controller}/{taskId}/taskSubmission")]
Ссылка: 0
public IActionResult SubmitTask(Guid taskId)
{
    var task = new TaskSubmissionDTO()
    {
        GroupTaskId = taskId
    };

    return View(task);
}

[HttpPost("{controller}/{taskId}/taskSubmission")]
Ссылка: 0
public async Task<IActionResult> SubmitTask([FromRoute] Guid taskId, TaskSubmissionDTO taskSubmissionDTO)
{
    taskSubmissionDTO.GroupTaskId = taskId;
    taskSubmissionDTO.UserId = Guid.Parse(User.Claims.FirstOrDefault(claim => claim.Type == "Id").Value);
    await _userService.SubmitTask(taskSubmissionDTO);
    return RedirectToAction("Index", "Home");
}

```

Рисунок 19 -метод *SubmitTask endpoint*

SubmitTask endpoint використовується для відправлення виконаного завдання. Перш за все з посилання метод отримує айді завдання. Після цього використовуючи метод SubmitTask йде відправка виконаного завдання. Після цього йде перенаправлення на домашню сторінку.

```
[HttpGet("{controller}/{taskId}/submittedTasks")]
Ссылка: 0
public async Task<IActionResult> SubmittedTasks(Guid taskId)
{
    return View(await _userService.GetGroupTask(taskId));
}
```

Рисунок 20 -метод SubmittedTasks endpoint

SubmittedTasks endpoint відповідає за отримання групи використовуючи метод GetGroupTask з сервісу.

```
[HttpGet("{controller}/{taskId}/markTask")]
Ссылка: 0
public async Task<IActionResult> MarkTask(Guid taskId)
{
    return View(await _userService.GetSubmittedTask(taskId));
}

[HttpPost("{controller}/{taskId}/markTask")]
Ссылка: 0
public async Task<IActionResult> MarkTask([FromRoute] Guid taskId, TaskSubmissionDTO taskSubmissionDTO)
{
    taskSubmissionDTO.Id = taskId;
    await _userService.MarkTask(taskSubmissionDTO);
    return RedirectToAction("Index", "Home");
}
```

Рисунок 21 - метод MarkTask endpoint

Так як контроллер не має мати ніякого функціоналу у кожному ендпоінті викликається метод з сервісу який знаходиться у BussinessLogic.

3.4 Під проект BussinessLogic

BussinessLogic відповідає за функціональну частину проекту. Він має прямий доступ до під проекту DataAccess, але не має доступу до Host, це зроблено

для сек'юрності проекту. Весь функціонал знаходиться у сервісі UserService. Для сервісу UserService було створено інтерфейс IUserService від якого він унаслідується. Це було зроблено для того щоб зменшити залежність.

```
public async Task<AuthResultDTO> LoginUser(UserLoginDTO userLoginDTO)
{
    var existingUser = await _dbContext.Users
        .AsNoTracking()
        .SingleOrDefaultAsync(user => user.Username == userLoginDTO.Username);

    if (existingUser == null || !BCrypt.Net.BCrypt.Verify(userLoginDTO.Password, existingUser.PasswordHash))
    {
        return new AuthResultDTO
        {
            Success = false,
            Errors = new string[] { "Invalid login or password" }
        };
    }

    await Authenticate(existingUser.Username, existingUser.Role, existingUser.Id);

    return new AuthResultDTO() { Success = true };
}
```

Рисунок 22 – метод *LoginUser*

Метод LoginUser викликається під час логіну користувача. Для того щоб його залогінити ми для початку перевіряємо чи є користувач з таким логіном у базі, і після цього перевіряємо чи дорівнює хеш введеного паролю с хешем паролю у базі. Якщо перевірки пройдено успішно користувача залогінює, а метод повертає результат з успішною міткою.

```

public async Task<AuthResultDTO> RegisterUser(UserRegisterDTO userRegisterDTO)
{
    var existingUser = await _dbContext.Users
        .AsNoTracking()
        .SingleOrDefaultAsync(user => user.Username == userRegisterDTO.Username);

    if (existingUser != null)
    {
        return new AuthResultDTO
        {
            Success = false,
            Errors = new string[] { "User with this login is already exists" }
        };
    }

    User newUser = _mapper.Map<UserRegisterDTO, User>(userRegisterDTO);

    _dbContext.Add(newUser);
    await _dbContext.SaveChangesAsync();

    return new AuthResultDTO() { Success = true };
}

```

Рисунок 23 – метод RegisterUser

Метод RegisterUser викликається під час реєстрації користувача. Для реєстрації насамперед йде перевірка чи не зайнятий такий логін. Якщо логін не зайнятий то новий користувач вноситься в базу даних і віддається результат про успіх.

```

public async Task CreateGroup(CreateGroupDTO createGroupDTO )
{
    Group group = _mapper.Map<GroupDTO, Group>(createGroupDTO.GroupDTO);

    GroupUser[] groupUsers = createGroupDTO.GroupUsersId.Select(id =>
    {
        return new GroupUser
        {
            UserId = id
        };
    }).ToArray();

    group.GroupUsers.AddRange(groupUsers);
    _dbContext.Group.Add(group);
    await _dbContext.SaveChangesAsync();
}

```

Рисунок 24 – метод CreateGroup

Метод CreateGroup викликається під час створення групи. Для його створення не портібно ніяких перевірок, тому група просто вноситься в базу.

```
public async Task<ICollection<User>> GetUsers()
{
    return await _dbContext.Users.AsNoTracking().ToListAsync();
}
```

Рисунок 25 – метод *GetUsers*

Метод *GetUsers* викликається коли потрібно отримати всіх користувачів.

```
public async Task<ReadOnlyCollection<GroupDTO>> GetGroups(Guid id)
{
    var groups = await _dbContext.Group.AsNoTracking().Include(group => group.GroupUsers)
        .Where(group => group.TeacherId == id || group.GroupUsers.Any(groupUser => groupUser.UserId == id)).ToArrayAsync();
    return _mapper.Map<Group[], GroupDTO[]>(groups);
}
```

Рисунок 25 – метод *GetGroups*

Метод *GetGroups* викликається коли потрібно отримати всі групи користувача. Так як в групи можуть мати як вчителі так і студенти, перевірка по айді йде і в вчителя, і в студентів.

```
public async Task Logout()
{
    await _httpContextAccessor.HttpContext.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme);
}
```

Рисунок 26 – метод *Logout*

Метод *Logout* викликається коли потрібно розлогінити користувача.

```
public async Task<GroupDTO> GetGroup(Guid id)
{
    return _mapper.Map<Group, GroupDTO>(await _dbContext.Group
        .Include(group => group.Teacher)
        .Include(group => group.GroupUsers)
        .Include(group => group.GroupTasks).ThenInclude(groupTask => groupTask.File)
        .FirstOrDefaultAsync(group => group.Id == id));
}
```

Рисунок 27 – метод *GetGroup*

Метод `GetGroup` викликається коли потрібно отримати групу, користувачів групи та завдання групи. Так як група зберігає вчителів та студентів, використовуючи метод `Include` йде підтягування з бази даних.

```
public async Task CreateTask(GroupTaskDTO groupTaskDTO)
{
    var groupTask = _mapper.Map<GroupTaskDTO, GroupTask>(groupTaskDTO);

    var fileId = await _blobStorageService.AddTaskFile(DataAccess.BlobStorage.BlobContainer.GroupTasks, groupTaskDTO.File);
    groupTask.FileId = fileId;
    _dbContext.GroupTask.Add(groupTask);
    await _dbContext.SaveChangesAsync();
}
```

Рисунок 28 – метод `CreateTask`

Метод `CreateTask` викликається коли потрібно створити завдання для групи.

```
public async Task SubmitTask(TaskSubmissionDTO taskSubmissionDTO)
{
    var submittedTask = _mapper.Map<TaskSubmissionDTO, TaskSubmission>(taskSubmissionDTO);

    var fileId = await _blobStorageService.AddTaskFile(DataAccess.BlobStorage.BlobContainer.TaskSubmissions, taskSubmissionDTO.File);
    submittedTask.FileId = fileId;
    submittedTask.SubmissionDate = DateTime.UtcNow;
    _dbContext.TaskSubmissions.Add(submittedTask);
    await _dbContext.SaveChangesAsync();
}
```

Рисунок 29 – метод `SubmitTask`

Метод `SubmitTask` викликається коли користувач відправляє виконане завдання.

```
public async Task<GroupTaskDTO> GetGroupTask(Guid taskId)
{
    return _mapper.Map<GroupTask, GroupTaskDTO>(await _dbContext.GroupTask.AsNoTracking()
        .Include(groupTask => groupTask.Submissions).ThenInclude(task => task.File)
        .Include(groupTask => groupTask.Submissions).ThenInclude(task => task.User)
        .FirstOrDefaultAsync(groupTask => groupTask.Id == taskId));
}
```

Рисунок 30 – метод `GetGroupTask`

Метод `GetGroupTask` викликається коли потрібно отримати завдання та всі здані відповіді до нього. Для підтягування з бази користувачів та завдання використовується `Include`.

```
public async Task<TaskSubmissionDTO> GetSubmittedTask(Guid taskId)
{
    return _mapper.Map<TaskSubmission, TaskSubmissionDTO>(await _dbContext.TaskSubmissions.AsNoTracking()
        .FirstOrDefaultAsync(task => task.Id == taskId));
}
```

Рисунок 31 – метод `GetSubmittedTask`

Метод `GetSubmittedTask` викликається коли потрібно отримати виконане завдання для перевірки.

```
public async Task MarkTask(TaskSubmissionDTO taskSubmissionDTO)
{
    var task = await _dbContext.TaskSubmissions.FirstOrDefaultAsync(task => task.Id == taskSubmissionDTO.Id);
    task.Mark = taskSubmissionDTO.Mark;
    task.TeacherComment = taskSubmissionDTO.TeacherComment;
    await _dbContext.SaveChangesAsync();
}
```

Рисунок 31 – метод `MarkTask`

Метод `MarkTask` викликається коли потрібно оцінити виконане завдання та залишити коментар до нього.

3.5 Клас `AutoMapper`

Майже в усіх методах сервіса викликається `_mapper.Map`, цей метод потрібен для мапінгу об'єктів.

Мапінг це процес переносу даних з полів одного об'єкту в інший. В проекті він використовується для заповнення полів з `DTO(Data Transfer Object)` у Ентиті та навпаки. Це потрібно для зменшення залежності.

Для мапінгу потрібен окремий клас для налаштування правил який називається Профіль. Профіль потрібен щоб автомапер розумів який об'єкт він

може мапити у інший, а також якщо імена полів не співпадають то записувати правила мапінгу для коректної роботи.

```
namespace TWApp.BussinessLogic.Mapping
{
    public class DTOsProfile: Profile
    {
        public DTOsProfile()
        {
            CreateMap<User, UserLoginDTO>();
            CreateMap<UserRegisterDTO, User>()
                .ForMember(user => user.PasswordHash,
                    rule => rule.MapFrom(userRegisterDTO => BCrypt.Net.BCrypt.HashPassword(userRegisterDTO.PasswordHash)));
            CreateMap<GroupDTO, Group>().ReverseMap();
            CreateMap<GroupTaskDTO, GroupTask>()
                .ForMember(groupDTO => groupDTO.File,
                    rule => rule.Ignore())
                .ReverseMap();
            CreateMap<TaskSubmissionDTO, TaskSubmission>()
                .ForMember(taskSubmissionDTO => taskSubmissionDTO.File,
                    rule => rule.Ignore())
                .ReverseMap();
        }
    }
}
```

Рисунок 31 – клас *DTOsProfile*

3.6 Висновок

Люди, які володіють обладнанням, користуються великим попитом на робочому місці. І більшість програм дистанційного навчання використовують певні технології для доставки контенту.

Програми дистанційного навчання розвивають ці технічні навички, які застосовуються до багатьох професій:

- Онлайн дослідження
- Обробка текстів
- Відеоконференція
- Слайд-презентації
- Дискусійні дошки
- Заявки на співпрацю
- Соціальна мережа

- Листування електронною поштою
- Публікуйте інформацію на Wiki / веб-сайті / блозі
- Створення відео

Згідно зі звітом незалежної цільової групи The Work Ahead за 2018 рік, «майже дві третини з 13 мільйонів нових робочих місць, створених у Сполучених Штатах з 2010 року, вимагали середніх або просунутих цифрових навичок». Тож не можна заперечувати, що за онлайн-грамотністю майбутнє, а дистанційне навчання сприяє розвитку цих навичок.

ВИСНОВОКИ

У першому розділі дипломної роботи ми розглянули декілька тлумачень терміна «дистанційне навчання», визначили, що це нова форма навчання, як денна, так і заочна, важливою особливістю дистанційного навчання є використання нових інформаційні технології. з певного місця і часу.

Також розглянули особливості дистанційної освіти, серед яких: постійне спілкування викладача та студента, можливе залучення іноземних викладачів та студентів до співпраці, проведення різноманітних занять, велика аудиторія з мінімальними витратами.

У практичному розділі ми розглядали комунікативність у системі дистанційного навчання. Спілкування є основою дистанційного навчання, оскільки в основі лежить взаємодія учасників навчального процесу, від якості дистанційного навчання залежить рівень реалізації спілкування.

Ми окреслили систему організації взаємодії вчителя та учня, яка базується на зворотному зв'язку з викладачем, адже від цього фактора залежить те, як учень здобуде знання. Іншим важливим фактором є конкретна причина спілкування, спрямована на досягнення мети процесу навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Знайомство з Moodle // MoodleLearn. Як створити сайт із системою дистанційного навчання URL: <http://moodlelearn.ru/course/view.php?id=18>
2. Ефективність використання СДО Moodle для організації дистанційного навчання у школі // Шадрінський державний педагогічний університет URL: http://shgpi.edu.ru/files/faculties/f11/publication/conf_2012/it/menshikova.doc
3. Терміни та визначення дистанційного навчання // rudocs.exdat.com URL: <http://rudocs.exdat.com/docs/index-140798.html>
4. ПЕДАГОГІЧНІ ПІДСТАВИ КОНЦЕПЦІЇ ДИСТАНЦІЙНОГО НАВЧАННЯ // Лабораторія дистанційного навчання URL: <http://distant.ioso.ru/library/publication/con7.htm>
5. ПРІОРИТЕТИ ДИСТАНЦІЙНОГО НАВЧАННЯ І САМОКОНТРОЛЬ ВИКОРИСТАНИХ ЗНАНЬ // Publishing house Education and Science s.r.o. URL: http://www.rusnauka.com/10_NPE_2010/Pedagogica/62080.doc.htm
6. Елементи системи ДН // Студопедія. Орг – 2014-2016 рік. URL: <http://studopedia.org/10-109977.html>
7. Застосування технологій дистанційного навчання під час освоєння програм елективних курсів // ВД «Перше вересня», 2003–2016 URL: <http://festival.1september.ru/articles/581053/>
8. Цифрові технології в освіті URL: <http://festival.1september.ru/articles/581053/>
9. Дистанційне навчання. Проблеми та перспективи // Copyright 2001-2011. Питання Інтернет Освіти. URL: http://vio.uchim.info/Vio_98/cd_site/articles/art_2_6.htm

10. Система WebTutor // WebSoft, 1999-2014 р. URL: http://www.websoft.ru/db/wb/root_id/webtutor/doc.html
- 11.. Система дистанційного навчання Competentum.Magister// Портал "Дистанційне навчання", 2006-2016. URL: <http://dstudy.ru/?page=5b09fc48-7f08-42fd-adce-324d923bf727&item=99a10969-f05d-4c7d-8464-b334fd9f3ab1&type=page>
12. Програми для віддаленого навчання (сьогодні говоримо про Adobe Connect) // Хабрахабр URL: <https://habrahabr.ru/post/98346/>
13. Moodle // dLearn.org. 2008-2016 URL: <http://www.dlearn.org/directory/%D0%A1%D0%94%D0%9E/Moodle> (дата звернення: 07.05.2016). 18. Навчальне середовище Moodle // ALT Linux URL: <https://docs.altlinux.org/ruRU/archive/4.1/html-single/school-server/moodle/index.html>
14. Програма дистанційного навчання з математики // Проект «Інфоурок» 2012–2016 URL: https://infourok.ru/programma_distancionnogo_obucheniya_po_matematike_4_klass.shko_la_rossii-394740.htm
15. Навчання на відстані // Web-сервер БелДУ URL: <http://www.bsu.edu.ru/bsu/gazeta/archive/num522-523/contains/study.htm>
16. Дистанційна освіта – один із напрямів інформатизації вузу // Російсько-американський форум освіти: електронний журнал 2016 URL: <http://www.rus-ameeduforum.com/content/ru/?task=art&article=1000858&iid=10>
17. Основи теорії дистанційного навчання // Російська академія природознавства 2005-2016 URL: <https://monographies.ru/ru/book/section?id=4724>
18. Ярошенко Т. О. Дистанційне навчання в системі вищої освіти:

сучасні тенденції. *Інженерні та освітні технології*. 2019. № 4. С. 8–21.

19. Яценко Г. Ю. Комунікативний аспект в системі дистанційного навчання. URL: http://www.rusnauka.com/SND/Pedagogica/1_jacenko.doc.htm
(дата звернення: 28.11.2020)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра Інженерії програмного забезпечення



Розробка програмного забезпечення для перевірки виконання завдань мовою C#

Виконавець: студент 4 курсу,
групи ПД-43

Собко Іван Іванович

Керівник роботи: ст. викл. Гаманюк І. М.

Київ 2022

1

Аналіз аналогів

	Moodle	Talent LMS	Moco
Складність системи	+	+	+
Недоки інтерфейсу	+	+	-
Функціональність	+	-	+
Вартість	-	+	+



2

Мета, об'єкт та предмет роботи

Мета роботи – покращити спосіб перевірки виконання завдань з використанням розробленої програми.

Об'єкт дослідження –перевірка виконання завдань.

Предмет дослідження – програмне забезпечення для перевірки виконання завдань.

Використані інструменти



Мова програмування C#



Платформа .NET із прикладним інтерфейсом Windows Forms



Середовище розробки Visual Studio



PostgreSQL

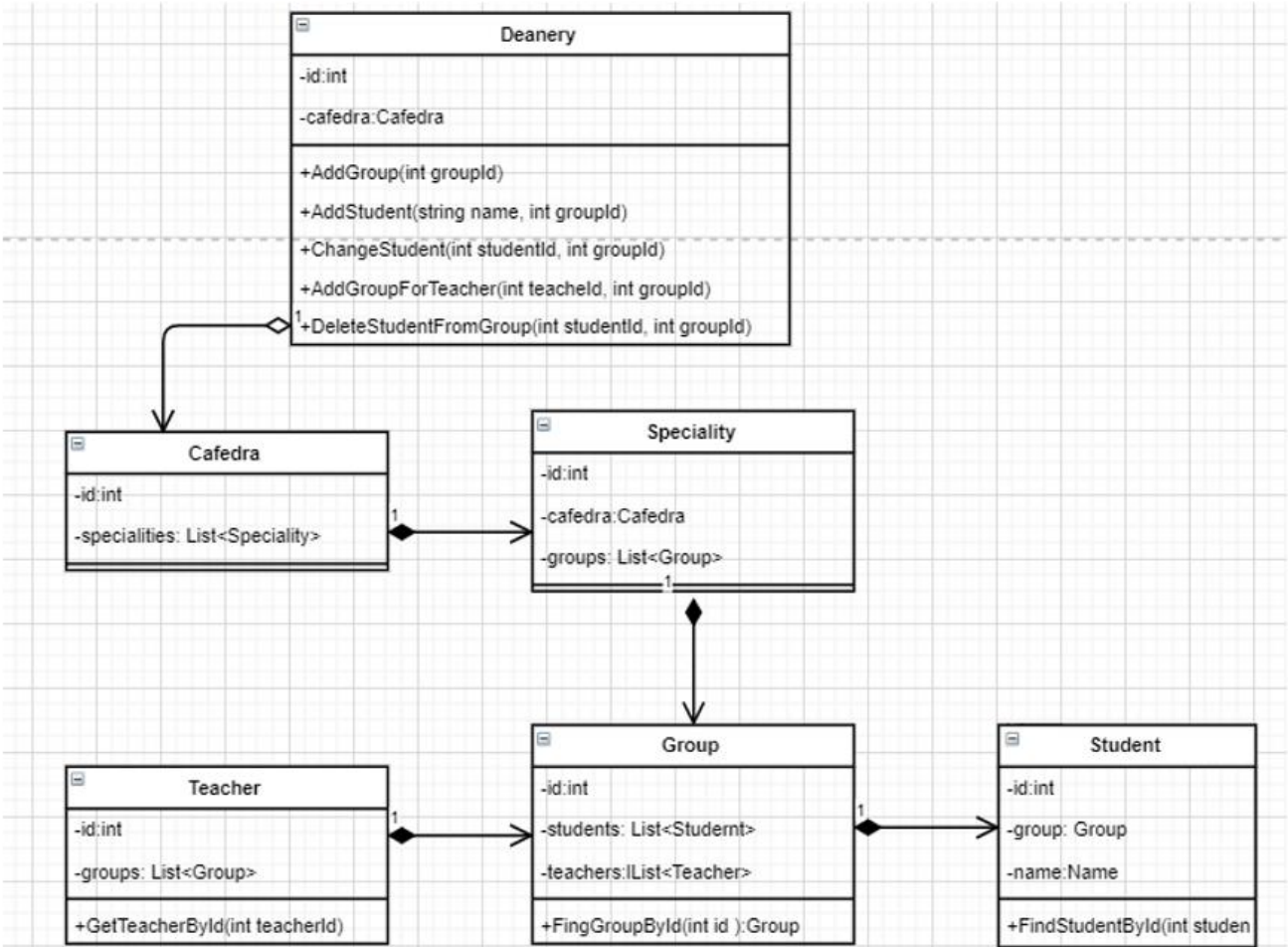
Система керування базами даних PostgreSQL

5

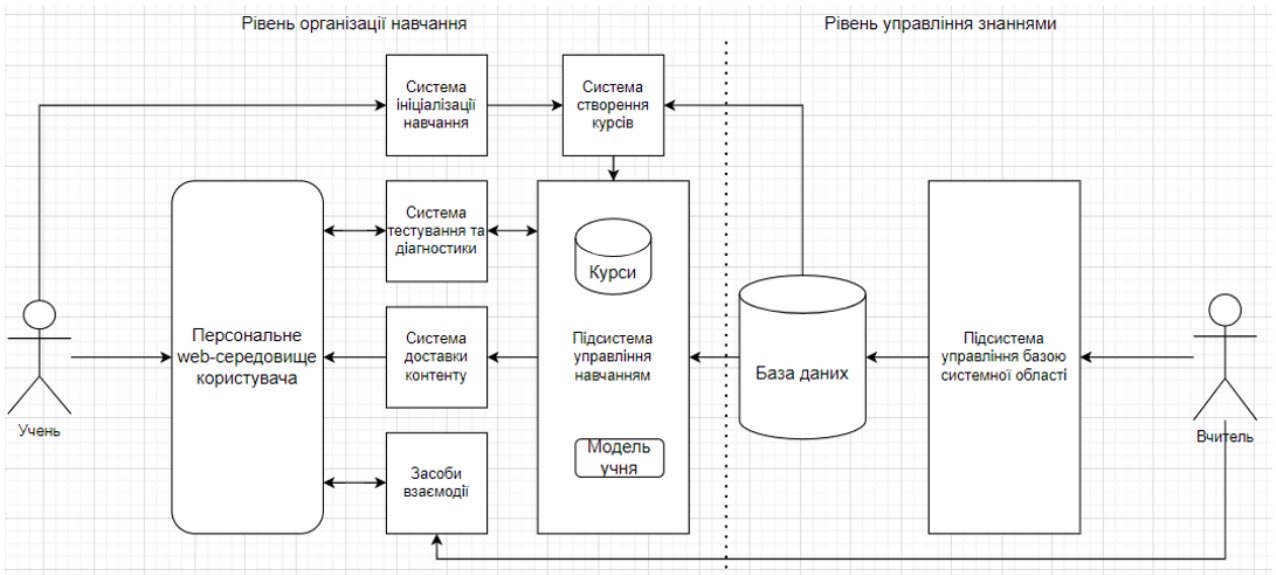
Технічне завдання

1. Розробити реєстрування користувачів з різними ролями.
2. Опрацювати функцію завантаження робіт.
3. Створити метод перевірки виконання завдань.
4. Підготувати реалізацію способів передачі даних вчителів до учнів.

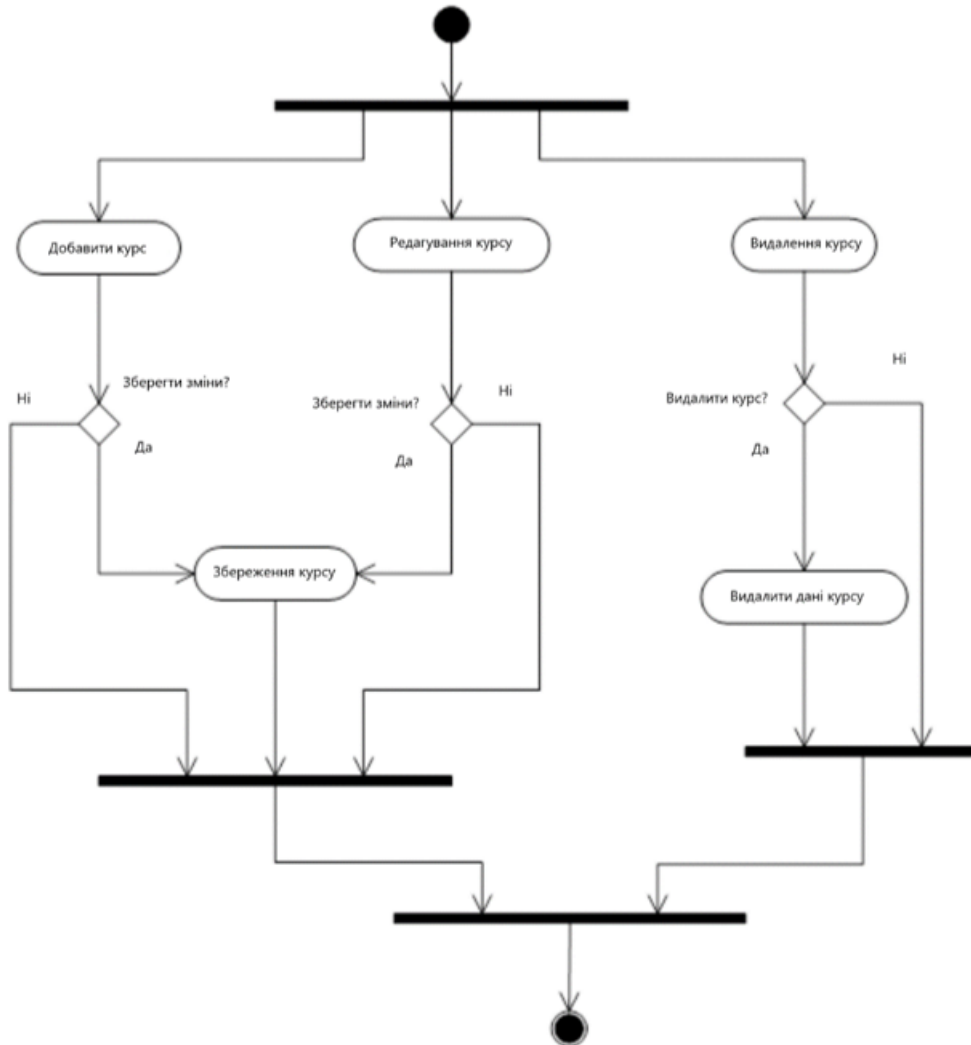
Діаграма класів



Архітектура системи



Діаграма взаємодії з курсом



Практичне застосування

The screenshot shows the login page of the TWMap application. The header includes 'TWMap' and 'Home' on the left, and a 'Sign In' link on the right. The main form contains a 'Login:' field with the text 'Teacher', a 'Password:' field with masked characters '...', and a 'SignIn' button.

Сторінка авторизації

The screenshot shows the 'Create Group' page. The header includes 'TWMap', 'Home', and 'Create Group' on the left, and a 'Logout' link on the right. The form includes a 'Group name:' field with 'Some Group', a 'Students:' dropdown menu with 'qweqwe' and 'Some Student' as options, and a 'Create Group' button.

Сторінка створення класу

9

Практичне застосування

The screenshot shows the 'Some Group' page. The header includes 'TWMap', 'Home', and 'Create Group' on the left, and a 'Logout' link on the right. The main content area displays 'Some Group' and a summary card with 'Some Group', 'Teacher: Some Man', and 'Tasks count: 0'. A 'Create Task' button is located at the bottom of the card.

Сторінка створення завдання

The screenshot shows the 'Create Task' page. The header includes 'TWMap', 'Home', and 'Create Group' on the left, and a 'Logout' link on the right. The form includes a 'Title:' field, a 'Description:' field, a 'File:' field with a file selection button and the text 'Файл не выбран', and a 'Create Task' button.

Сторінка завантаження завдання

10

Практичне застосування

Some Student

Comment: Some submit

Submission date:02.06.2022
00:00:00

Mark:0

[Get submitted task](#)

[Mark task](#)

Оцінювання роботи

TWMapp Home Create Group

TeacherComment:

Mark:

[Mark](#)

Виставлення оцінки

Some Student

Comment: Some submit

Submission date:02.06.2022
00:00:00

Mark:100

[Get submitted task](#)

[Mark task](#)

Робота оцінена

11

Апробація результатів дослідження

Собко Іван Іванович РОЗРОБКА МЕТОДИКИ ЗАСТОСУВАННЯ GPS ТРЕКЕРІВ ДЛЯ АВТОТРАНСПОРТУ / Ігор Михайлович Гаманюк, Іван Іванович Собко // Науково-технічна конференція «Застосування програмного забезпечення в інфокомунікаційних технологіях». Збірник тез. – К.: ДУТ, Київ - 2022.– С. 72

Висновки

1. Проаналізовано аналоги;
2. Визначено архітектуру системи;
3. Опрацьовано діаграму класів;
4. Змодульовано взаємодію з курсом;
5. Реалізовано реєстрування користувачів з різними ролями.
6. Розроблено функцію завантаження робіт.
7. Створено метод перевірки виконання завдань.
8. Підготовлено спосіб передачі даних вчителів до учнів.