

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

КОМПЛЕКСНА МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Web-система для дистанційного вивчення слів англійської мови.


Частина 1. «Підсистема оцінювання якості навчання»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконав: студент 2-го курсу, групи ІКІ-21м
спеціальності 123 – Комп'ютерна інженерія


 _____ Супрун П. Б.

Керівник: к.т.н., проф. каф. ОТ

 _____ Азарова А. О.

« 16 » _____ 12 _____ 2022 р.

Опонент: к.т.н., доцент каф. МБІС

 _____ Карпинець В. В.

« 19 » _____ 12 _____ 2022 р.

Допущено до захисту

Завідувач кафедри ОТ

д.т.н., проф. Азаров О. Д.

 _____
« 20 » _____ 12 _____ 2022 р.

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітній рівень – магістр
Напрямок підготовки – 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри
обчислювальної техніки

Азаров проф., д.т.н. О. Д. Азаров

« 15 » 03 2022 р.

З А В Д А Н Н Я

НА КОМПЛЕКСНУ МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Супруну Павлу Борисовичу

1 Тема роботи: «Web-система для дистанційного вивчення слів англійської мови. Частина 1. «Підсистема оцінювання якості навчання»»
керівник роботи: Азарова Анжеліка Олексіївна к.т.н., проф., затверджено наказом вищого навчального закладу від 15.09.2022 р. № 205-А.

2 Строк подання студентом роботи 19.12 2022

3 Вихідні дані до роботи: методи розробки web-застосунків, навчальні матеріали на тему створення web-застосунків, технічна документація стеку технологій .Net Core та мови програмування C#, системи-аналоги.

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

- вступ;
- аналітичний огляд комп'ютерних систем та існуючих підходів із оцінювання результатів навчання;
- моделювання та проєктування підсистеми оцінювання якості навчання;
- практична реалізація підсистеми оцінювання якості навчання;

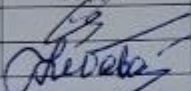
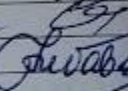
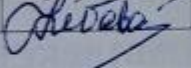
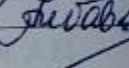
— економічна частина.

5 Структура підсистеми оцінювання якості навчання:

- модуль перевірки результатів навчання;
- модуль оцінювання результатів навчання;
- модуль обліку результатів навчання.

6 Консультанти розділів роботи наведені у таблиці 1.

Таблиця 1 – Консультанти розділів КМКР


Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Снігур А. В. к.т.н., доц. каф. ОТ		
4	Небава М. І. к.с.н., проф. каф. ЕПВМ		


7 Дата видачі завдання 19.09.2022

8 Календарний план розділів роботи наведено у таблиці 2.

Таблиця 2 – Календарний план КМКР

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Приміт
1	Постановка задачі роботи	19.09.2022	Викона
2	Дослідження методів дистанційного вивчення іноземної мови	23.09.2022	Викона
3	Математичне моделювання процесу оцінювання якості навчання	3.10.2022	Викона
4	Аналіз та вибір технологій проектування та розробки web-додатків	17.10.2022	Викона
5	Програмна реалізація системи	31.10.2022	Викона
6	Підготовка матеріалів для опису проектування підсистеми оцінювання якості навчання	7.11.2022	Викона
7	Оформлення пояснювальної записки	21.11.2022	Викона
8	Перевірка якості виконання магістерської роботи	01.12.2022	Викона

Студент  Супрун Павло Борисович

Керівник  к.т.н., проф. каф. ОТ Азарова Анжеліка Олексіївна

АНОТАЦІЯ

УДК 004.62

Супрун П. Б. Веб-система для дистанційного вивчення слів англійської мови. Частина 1. «Підсистема оцінювання якості навчання». Комплексна магістерська кваліфікаційна робота зі спеціальності 123 – комп'ютерна інженерія, освітня програма – комп'ютерна інженерія. Вінниця: ВНТУ, 2022. 131 с.

На укр. мові. Бібліогр.: 30 назв; рис.: 30; табл.: 9.

Комплексна магістерська кваліфікаційна робота присвячена розробленню веб-системи для дистанційного вивчення слів англійської мови та її підсистеми оцінювання якості навчання.

Основний напрямок дослідження даної частини магістерської роботи – проєктування та інтеграція підсистеми оцінювання якості навчання користувача. Під час виконання магістерської кваліфікаційної роботи проаналізовано системи-аналоги, виявлено та враховано їх недоліки у власній розробці, вивчено основні способи використання веб-технологій у різних навчальних процесах, у тому числі оцінювання результатів навчання, на основі яких розроблено математичні моделі для перевірки, оцінювання та зберігання результатів навчання.

Розроблено та реалізовано масштабовану архітектуру системи з використанням патернів проєктування. Систему розроблено в програмному середовищі .NET на основі технології ASP.NET Web API та призначена для використання у веб-браузерах.

Ключові слова: веб-розробка, веб-система, дистанційне навчання, оцінювання якості навчання, .NET.

ABSTRACT

Suprun P.B. Web-system for remote learning of English words. Part 1. «Subsystem for quality evaluation of the education». Comprehensive master's qualification work on specialty 123 – computer engineering, educational program – computer engineering. Vinnytsia: VNTU, 2022. 131 p.

In Ukrainian language. Bibliography: 30 titles; fig.: 30; tab.: 9.

The complex master's qualification work is devoted to the development of a web system for remote learning of English words and its subsystem of evaluating the quality of education.

The main research direction of this part of the master's thesis is the design and integration of the subsystem of evaluating the quality of user's training. During the completion of the master's qualification thesis, similar systems were analyzed, their shortcomings were identified and taken into account, the main ways of using web technologies in various educational processes were studied, including the evaluation of learning results, on the basis of which mathematical models were developed for verification, evaluation and storage of learning results.

A scalable architectural system was developed and implemented using design patterns. The web system is developed in the .NET programming environment based on the ASP.NET Web API technology and is intended for use in web browsers.

Keywords: web development, web system, distance learning, evaluation of learning quality, .NET.

ЗМІСТ

ВСТУП	8
1 ТЕОРЕТИЧНІ ЗАСАДИ ПОБУДОВИ КОМП'ЮТЕРНИХ СИСТЕМ ДИСТАНЦІЙНОГО НАВЧАННЯ.....	
	11
1.1 Роль та значення комп'ютерних систем дистанційного навчання.....	11
1.2 Аналіз підходів до організації процесу контролю й оцінювання результатів навчання	17
1.3 Функції діагностики результатів навчання.....	21
1.4 Аналіз недоліків та переваг існуючих математичних моделей процесу оцінювання знань респондента	25
1.5 Вивчення комп'ютерних систем та програмних засобів оцінювання якості навчання	30
Висновки до розділу 1.....	38
2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПІДСИСТЕМИ ОЦІНЮВАННЯ ЯКОСТІ НАВЧАННЯ.....	
	39
2.1 Розроблення структури підсистеми оцінювання якості навчання	39
2.2 Побудова моделі перевірки результатів навчання.....	45
2.3 Створення моделі оцінювання якості навчання	45
2.4 Моделювання процесу обліку результатів навчання	48
Висновки до розділу 2.....	50
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОЦІНЮВАННЯ ЯКОСТІ НАВЧАННЯ.....	
	52
3.1 Обґрунтування вибору технологій розробки та мов програмування	52
3.2 Проєктування програмних засобів підсистеми оцінювання якості навчання	58
3.3 Тестування програмного продукту.....	81

					08-23.КМКР.032.00.000 ПЗ			
<i>Змн.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Супрун П. Б.</i>			Web-система для дистанційного висчення слів англійської мови. Частина 1. «Підсистема оцінювання якості навчання»	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Азарова А. О.</i>					6	
<i>Реценз.</i>		<i>Карлінець В. В.</i>				ВНТУ, гр. 1КІ-21м		
<i>Н. Контр.</i>		<i>Швець С. І.</i>						
<i>Затверд.</i>		<i>Азаров О. Д.</i>						

Висновки до розділу 3	83
4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	85
4.1 Аналіз результатів, отриманих на базі запропонованого ПЗ.....	85
4.2 Керівництво оператора	89
4.3 Порівняння результатів роботи ПЗ із результатами провідних аналогів.....	93
Висновки до розділу 4.....	97
5. ЕКОНОМІЧНА ЧАСТИНА	98
5.1 Комерційний та технологічний аудит науково-технічної розробки.....	98
5.2 Прогнозування витрат на виконання науково-дослідної роботи	100
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	104
Висновки до розділу 5.....	109
ВИСНОВКИ.....	110
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	112
ДОДАТОК А Технічне завдання	116
ДОДАТОК Б Структура системи	119
ДОДАТОК В Алгоритм роботи системи	120
ДОДАТОК Г Лістинг репозиторіїв	121
ДОДАТОК Д Лістинг сервісів	125
ДОДАТОК Е Лістинг контролерів	131
ДОДАТОК Є Ілюстративний матеріал до захисту	138
ДОДАТОК Ж Протокол перевірки кваліфікаційної роботи.....	144

					08-23.КМДР.032.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

У сучасному світі, де можливостей для саморозвитку стає все більше, спостерігається помітне зростання кількості людей, які бажають вивчати іноземні мови онлайн. Це зручно, доступно та значно дешевше, ніж традиційне навчання, крім того, такий підхід дозволяє раціонально витратити обмежений час.

У такої форми навчання є широкий спектр споживачів: від школярів, абітурієнтів, студентів, науковців до пересічних громадян, які бажають покращити свій рівень конкурентоспроможності на ринку праці за рахунок отримання нових кваліфікаційних ознак, що є особливо важливим в умовах інтеграції України до світової спільноти.

За сучасного рівня розвитку комп'ютерних технологій дистанційне вивчення англійської мови відкриває нові можливості в навчанні, значно розширюючи інформаційні простір та сферу навчання. Використовуючи сучасні форми інформаційного обміну, системної та міжсистемної взаємодії, навчання на відстані надає нові механізми реалізації міжпредметних, міжнаукових та соціокультурних зв'язків. Дистанційна сфера навчання англійської мови має бути необмеженою за обсягом та відкритою для інформаційної взаємодії, доступу та використання всіма суб'єктами освіти. З іншого боку, вона повинна бути визначеною стандартами навчання, що координуються системами навчання та керуються викладачами англійської мови. Тобто інформаційна сфера як середовище навчання англійської мови повинна мати потужне систематизоване ядро (базове середовище навчання), яке породжує цю сферу та всі її процеси і відносини.

Серед провідних науковців, які займалися розвитком методології такої форми навчання необхідно зазначити: Азарова О. Д., Бикова В. Ю., Гриценчука О. О., Залізецького В. В., Коневщинську О. Е., Крупельницького Л. В., Маркову Є. С., Сімонсона М., Черняка О. І. та ін. [1–5].

Варто зауважити, що існуючі математичні моделі та методи дистанційного навчання мають свої обмеження, зокрема, неповне оцінювання рівня знань респондента, недостатня його точність та висока вартість впровадження у практичну діяльність. Крім того, аналіз наявних комп'ютерних систем дистанційного навчання дозволив виявити низку їх недоліків, зокрема, непродуманий інтерфейс системи, слабка взаємодія суб'єктів навчального процесу, платний доступ до повнофункціональних систем.

Отже, з урахуванням активного запиту суспільства на дистанційну форму навчання та недостатню розвиненість відповідного теоретичного та практичного підґрунтя актуальними є подальші дослідження у цьому напрямку, що реалізовані в межах комплексної магістерської роботи.

Метою роботи є удосконалення дистанційної системи вивчення англійської мови шляхом створення власної підсистеми оцінювання якості навчання та її інтеграції до веб-системи дистанційного навчання.

Для досягнення зазначеної мети було поставлено та розв'язано такі **завдання**:

- проаналізувати існуючі підходи до організації контролю та оцінювання результатів навчання;
- вивчити недоліки та переваги наявних комп'ютерних систем оцінювання якості дистанційного навчання;
- розробити структуру підсистеми оцінювання якості навчання;
- на базі аналізу недоліків та переваг існуючих моделей дистанційного навчання побудувати базові моделі власної системи, зокрема: моделі перевірки результатів навчання та оцінювання його якості, модель процесу обліку результатів навчання;
- обґрунтувати вибір інструментів для програмної реалізації розробки;
- інтегрувати підсистему оцінювання якості навчання до веб-системи дистанційного вивчення слів англійської мови;

— здійснити тестування розробленої системи.

Об'єкт дослідження – процес розроблення веб-системи для дистанційного навчання.

Предмет дослідження – розроблення Web-системи для дистанційного вивчення слів англійської мови у межах підсистеми оцінювання якості навчання.

Наукова новизна отриманих результатів магістерської роботи полягає в удосконаленні системи дистанційного навчання англійської мови, яка, на відміну від існуючих, має розширені можливості за рахунок впровадження підсистеми оцінювання результатів навчання, створеної засобами математичного моделювання, та врахування індивідуальних факторів користувача.

Практичне значення отриманих результатів полягає в можливості підвищення рівня володіння англійською мовою засобами створеного веб-застосунку для оцінювання якості навчання, що вбудовується до Web-системи для дистанційного вивчення слів англійської мови.

Апробацію результатів магістерської роботи було здійснено на регіональній конференції ВНТУ (м. Вінниця, 2022).

Публікації. За результатами роботи підсистеми оцінювання якості навчання було опубліковано тези доповідей [6], подано заяву на реєстрацію авторського права на програму.

1 ТЕОРЕТИЧНІ ЗАСАДИ ПОБУДОВИ КОМП'ЮТЕРНИХ СИСТЕМ ДИСТАНЦІЙНОГО НАВЧАННЯ

1.1 Роль та значення комп'ютерних систем дистанційного навчання

Технології дистанційного навчання мають великий дидактичний потенціал в системі мовної підготовки респондентів на всіх етапах навчання. Цифровізація суспільства та соціально-економічні умови – все це спонукає сьогодні навчальні заклади до переорієнтації освітніх ресурсів на впровадження нових форм навчання, зокрема до застосування технологій дистанційного навчання. Активне використання інформаційних та комунікаційних технологій на сьогодні є одним із стратегічних напрямів реформування освітньої системи України на шляху до набуття нею конкурентоспроможності на ринку освіти. Впровадження таких технологій може становити вагомий аргумент для громадян, які обирають вищі навчальні заклади України для здобуття освіти.

В умовах широкого розповсюдження веб-технологій актуальною стає потреба використання їх в освіті як для покращення методів та технології навчання, так і для підвищення його ефективності [1]. Аналіз актуальних методів навчання показує, що до останнього часу використанню новітніх інформаційно-комунікаційних технологій, зокрема в процесі мовної підготовки, приділялося надзвичайно мало уваги. Тому метою дослідження стало вдосконалення наявних технологій дистанційного навчання для вивчення іноземної мови.

Дистанційне навчання у широкому сенсі – це форма організації навчального процесу, що базується на використанні інформаційних технологій і може бути реалізована як в умовах географічної віддаленості суб'єктів навчального процесу, так і безпосередньо в закладі освіти для формування самостійної діяльності респондентів щодо засвоєння програми навчання за спеціальністю.

Під терміном «дистанційне навчання» передбачається організація освітнього процесу, що ґрунтується на принципі самостійного навчання студента; сукупність інформаційного забезпечення та всіх технічних засобів, які забезпечують доступ до інформації, інтерактивну взаємодію між суб'єктами навчального процесу, можливість самостійно і незалежно від факторів часу та простору здійснювати навчання, а також створюють умови для перевірки й оцінювання знань; новий ступінь заочного навчання, ключовою ознакою якого є використання комп'ютера, програмного забезпечення та доступу до комп'ютерної мережі. Дистанційне навчання також визнається як інтегральна, синтетична, гуманістична форма навчання, що використовує широкий спектр традиційних і нових інформаційних технологій та їх технічних засобів, основними ознаками якого є використання інформаційно-комунікативних технологій, активність того, хто навчається, його самостійність щодо організації навчального процесу, визначення часу, темпу та інтенсивності навчальної діяльності, а також безперешкодний доступ до інформації та з можливістю інтерактивного спілкування з викладачем у асинхронному режимі.

Застосування елементів дистанційного навчання фактично є атрибутивною ознакою сучасної іншомовної освіти, оскільки навчально-методичні матеріали для неї зазвичай містять мультимедійні додатки, які роблять навчальний процес більш ефективним, забезпечуючи комплексне формування та розвиток навичок усіх видів мовної діяльності [2].

Для освітніх установ пошук нових і більш досконалих методів передавання знань є одним із провідних завдань системи дистанційного навчання. До переваг навчання із застосуванням технологій системи слід віднести таке:

— свобода доступу – суб'єкти навчального процесу мають можливість доступу до електронних курсів, підручників, завдань із будь-якого місця, де є вихід до глобальної інформаційної мережі;

- можливість навчання респондента за індивідуальним планом відповідно до власних можливостей і потреб;
- можливість консультації із викладачем у процесі виконання завдання, особливо якщо на початковому етапі виникають труднощі;
- об'єктивна методика оцінки знань, набутих у процесі навчання (технології системи дистанційного навчання дозволяють виставляти чіткі критерії, за якими оцінюються знання, отримані в процесі навчання);
- гнучкість навчання (тривалість і послідовність вивчення теми, розділу тощо респондент обирає самостійно, адаптуючи таким чином процес навчання під свої можливості);
- можливості професійно-інтелектуального розвитку як для респондентів, так і для викладачів, що залучені до роботи із ними, адже і перші, і другі розвивають навички та знання відповідно до сучасних технологій);
- своєчасність оновлення навчальних матеріалів, що подані в електронних курсах;
- компетентність освіти, оскільки частіше за все електронні курси створюються цілою командою фахівців;
- цільова економія витрат на навчально-методичне забезпечення, що є актуальною рисою системи дистанційного навчання в умовах сьогодення.

Варто зазначити, що використання інформаційно-комунікативних технологій навчання є доцільним та ефективним лише тоді, коли враховуються три виміри дистанційного навчання: технологічний, педагогічний та економічний, які мають узгоджуватися між собою. І в межах кожного з цих вимірів керівництво вищого навчального закладу має реалізувати заходи для успішного впровадження технологій дистанційного навчання в освітню діяльність закладу [3].

Педагогічні заходи з впровадження системи дистанційного навчання для вивчення мови:

- постійне оновлення та вдосконалення електронних навчально-методичних матеріалів;
- розробка електронних підручників та електронних навчально-методичних комплексів, які б забезпечували опанування не лише розмовної мови, а й профільної, фахової;
- розробка підручників комунікативного типу, які б максимально сприяли опануванню розмовної мови;
- створення ефективної системи оцінювання знань та навичок;
- створення методичного апарату до аудіовізуальних матеріалів, розробка завдань з опанування письмових навичок та методів їх перевірки, що складно зробити, якщо викладач і респондент розділені відстанню;
- уточнення лексичного та граматичного мінімуму, яким має оволодіти респондент у ході навчання, як за окремими модулями, так і за підсумками вивчення дисципліни загалом;
- формування мотивації до опанування мови навчання;
- регламентація взаємодії між викладачами та респондентами, зокрема дотримання правил етикету комунікацій, технологічна дисциплінованість, створення сценарію занять або інших публічних навчальних процесів;

Технологічні заходи з впровадження даної системи:

- забезпечення балансу між синхронними навчальними системами, які забезпечують можливість безпосередньої взаємодії викладача і респондента (у режимі «on-line»), та асинхронними, які передбачають, що викладач і респондент діють незалежно один від одного (у режимі «off-line»);
- підвищення кваліфікації викладачів щодо володіння інформаційно-комунікаційними технологіями, підготовка тьюторів;
- постійний моніторинг та контроль процесу навчання;

— захист інтелектуальної власності, власних технологічних розробок та запобігання їх використанню конкурентами.

До економічних та організаційних заходів з впровадження системи дистанційного навчання належить створення робочих груп кваліфікованих спеціалістів для розробки інформаційних навчальних систем та оплата їхньої праці.

Процес навчання спрямований на розв'язання освітніх завдань, кожне з яких характеризується певною дидактичною завершеністю. Обов'язковим компонентом цього процесу є контроль результатів навчання. Далі у даному розділі пропонується розглянути поняття «діагностика» та «контроль».

Діагностика – це складний процес вимірювання результатів навчання, з'ясування умов і обставин, в яких протікає процес навчання; отримання чіткого уявлення про прогалини в знаннях та вміннях, про ті причини, які сприяють чи перешкоджають досягненню накреслених результатів. Як бачимо, у процес діагностики вкладається більш широка і глибока суть: він розглядає результати навчання у тісному зв'язку зі шляхами і способами їх досягнення. Крім традиційних для контролю перевірки та оцінювання знань, умінь, діагностика включає їх аналіз, виявлення динаміки, тенденцій, прогнозування подальшого розвитку освітнього процесу.

Отже, структура процесу діагностики містить підсистеми контролю, оцінювання результатів контролю, їх аналізу і прогнозування. Розглянемо кожну з підсистем докладніше.

Поняття «контроль» має кілька значень. У дидактиці – це нагляд, спостереження, перевірка і оцінювання результатів навчання респондентів. Головна мета контролю як дидактичного засобу управління навчанням – отримання зворотного зв'язку, забезпечення ефективності процесу навчання шляхом:

— приведення до системи знань, умінь респондентів;

- формування компетентності самостійного застосування здобутих знань на практиці;
- стимулювання навчальної діяльності;
- формування прагнення до самоосвіти.

Об'єктами контролю є набуті компетентності, зокрема:

- знання, навички, уміння;
- способи мислення;
- досвід творчої діяльності респондентів;
- досвід емоційно-ціннісного ставлення до оточуючого світу, ціннісні ставлення, особистий досвід респондентів, їхні дії, переживання, почуття, ставлення до людей, явищ природи, пізнання тощо.

У контексті компетентнісної освіти це виявляється у відповідальності респондентів, прагненні закріплювати позитивні надбання в навчальній діяльності, зростання вимог до програмних результатів навчання [4].

Сьогодні вже стало очевидним, що система дистанційного навчання не є тимчасовим захопленням, і вирішення її проблем потребує комплексного підходу, а саме створення міжвузівських центрів підготовки викладачів, комплексної інформатизації навчальних закладів, формування пунктів з якості оцінки електронних курсів.

Отже, можна підсумувати, що впровадження технологій дистанційного навчання в освітню діяльність навчальних закладів підвищує їхню конкурентоспроможність на ринку освітніх послуг, хоча й потребує здійснення системи заходів з їх реалізації. Подальшого вивчення потребують питання доцільності використання окремих форм навчання на різних етапах мовної підготовки студентів.

1.2 Аналіз підходів до організації процесу контролю й оцінювання результатів навчання

Контроль складається з перевірки, оцінювання та обліку результатів навчання.

Перевірка визначається як виявлення, вимірювання і порівняння результатів навчання на тому чи тому етапі навчання.

Оцінювання – визначення і вираження у балах (оцінка, кількісне оцінювання) чи оцінних судженнях викладача (якісне, вербальне оцінювання) рівня засвоєння знань, сформованості умінь респондента відповідно до програмних результатів навчання. Оцінне судження сприяє розумінню респондентами якості та рівня засвоєних знань, умінь і навичок, включає оцінку способів роботи респондентів, їхнього ставлення до навчання, ступінь їхньої старанності, коротко характеризує відповіді респондента. Обґрунтовуючи бал, викладач аналізує виявлені знання за формою, змістом, обсягом, з погляду правильності та виразності мовлення. Значну користь можуть принести окремим респондентам рекомендації про те, які саме прогалини в знаннях необхідно допрацювати, яку конкретно роботу потрібно для цього виконати. З використанням оцінного судження пов'язане і вміння респондентів розгорнуто проаналізувати свою роботу, виявити залежність своєї оцінки від способу роботи. Якщо об'єктивно встановлений бал підбиває підсумок роботи респондента в минулому, то розумне, добре продумане слово під час його виставлення націлює респондента на майбутнє.

Облік полягає у фіксації і збереженні отриманих результатів навчання у вигляді оцінок і оцінних суджень викладача в журналі, таблиці успішності, сертифікаті тощо. Матеріали обліку результатів навчання є підґрунтям для здійснення аналізу перебігу навчального процесу і процесу прогнозування, який, у свою чергу, складається з:

- прогнозування викладачем діяльності респондентів;
- прогнозування викладачем власної діяльності;
- прогнозування викладачем тенденцій навчального процесу;

Усі зазначені складники взаємозалежні і взаємодіють у процесі діагностики.

Отже, завданнями процесу діагностики є:

- виявлення обсягу, рівня, якості засвоєння навчального матеріалу;
- виявлення труднощів і помилок;
- отримання інформації про самостійну роботу;
- перевірка ефективності обраних форм організації, методів і засобів навчання;
- встановлення готовності респондентів до вивчення нового матеріалу.

Результати контролю навчально-пізнавальної діяльності респондентів виявляються в її оцінці. Поняття «оцінка» означає характеристику цінності, рівень чи значення будь-яких об'єктів або процесів. Оцінити – означає встановити рівень чи якість чогось. Стосовно навчально-пізнавальної діяльності оцінка означає встановлення ступеня виконання респондентами завдань, поставлених перед ними в процесі навчання, рівня їх підготовки і розвитку, якості набутих знань, сформованих навичок і умінь.

Індивідуальний контроль застосовується для ґрунтовного ознайомлення із рівнем та подальшого аналізу результатів навчання. При цьому звертається увага на осмислений характер відповіді респондента, логічність його суджень, доказовість положень, уміння застосовувати засвоєні знання. Ця форма контролю залежить від часу, що відводиться на контроль, характеру й обсягу вивченого матеріалу, рівня підготовки респондентів.

На практиці використовуються три основні види контролю: попередній, поточний та підсумковий (табл. 1.1).

Попередній контроль використовується для виявлення рівня підготовки респондентів безпосередньо перед початком навчання, щоб дидактично цілеспрямовано організувати процес навчання.

Поточний контроль і поточне оцінювання здійснюється під час повсякденної навчальної роботи. Він дозволяє виявляти, як здійснюється процес опанування нових знань, які зустрічаються труднощі. Відповідно до отриманих результатів респондент або впевнено йде далі, або потрібно внести в навчальний план певні корективи.

Підсумковий контроль проводиться, звичайно, після вивчення розділу програми або теми з урахуванням даних поточного контролю і передбачає вияв знань, навичок, умінь, що забезпечує об'єктивність перевірки, реалізацію вимог індивідуального підходу, підвищення у респондентів почуття відповідальності за навчальну роботу [5].

Таблиця 1.1 – Порівняльна характеристика типів контролю результатів навчального процесу

Контроль	Попередній	Поточний	Підсумковий	Індивідуальний
Вхідні дані тестування	Однакові для всіх	Однакові для всіх	Однакові для всіх	Індивідуальні для кожного
Система оцінки	Фіксована	Фіксована	Фіксована	Гнучка
Багаторазовість	Так	Так	Так	Ні
Об'єм вихідних даних	Фіксований	Фіксований	Фіксований	Гнучкий

Кожен з описаний видів контролю має як свої недоліки, так і переваги, тому їх варто застосовувати лише за призначенням.

Попередній контроль застосовується лише один раз у межах вивчення одного матеріалу. Його доцільно застосовувати у предметних областях, де має значення наявний у респондента досвід, як приклад при вивченні мови. Якщо у респондента є певний рівень знань у даній сфері, йому буде не ефективно починати вивчення з нуля. Система, за якою проводиться оцінювання, повинна чітко, за встановленими критеріями визначити рівень знань респондента, тому вона є фіксованою, що в свою чергу не сприяє об'єму вихідних даних після контролю результатів навчального процесу. Даний тип контролю може багаторазово застосовуватися різними респондентами після одноразового створення, тому його впровадження дає значний ефект при відносно малих витратах ресурсів.

Поточний контроль застосовується найчастіше зі всіх типів контролю, зазвичай після закінчення вивчення теми, або модуля. Його доцільно застосовувати в будь-якій предметній області для оцінки якості закріплення невеликого обсягу пройденого матеріалу. Результати даного типу контролю дозволяють оперативно корегувати навчальний план, виносячи проблемні моменти на повторне вивчення одразу, не чекаючи кінця теми, чи модуля. Система, за якою проводиться оцінювання, повинна чітко, за встановленими критеріями відділити засвоєні фрагменти знань від незасвоєних, тому вона є фіксованою, що також не сприяє об'єму вихідних даних після проведення даного типу контролю. Даний тип контролю може багаторазово застосовуватися різними респондентами, тому його впровадження дає значний ефект при малих витратах ресурсів.

Підсумковий контроль застосовується для виявлення рівня знань із певної предметної області. Він включає в себе значно більшу, ніж поточний контроль, кількість та різноманітність завдань. Його доцільно застосовувати в кінці вивчення певної теми, чи курсу. Результати даного типу контролю дозволяють викладачу оцінити ефективність всього курсу, а респонденту – підсумковий рівень своїх знань, як наприклад рівень знання мови після проходження навчання. Система, за якою

проводиться оцінювання, виявляє рівень засвоєння знань за чітко встановленими критеріями, тому вона є фіксованою. Об'єм вихідних даних є фіксованим. Даний тип контролю може бути застосований багаторазово, а його впровадження є необхідністю, оскільки без нього не вдасться оцінити ні рівня знань, ні ефективності курсу навчання.

Індивідуальний контроль є найбільш гнучким, порівняно з попередніми типами контролю. Даний тип контролю найчастіше застосовується замість поточного. Його особлива відмінність походить від назви: для респондента формується індивідуальний список завдань та система оцінювання, що в результаті приводить до значно ширших результатів контролю. Даний тип контролю, в залежності від налаштувань, може давати на виході різноманітні дані, які можуть бути корисними в залежності від цілі проведення контролю, як наприклад визначення рівня засвоєння вибірки знань з непов'язаних тем, виключення певних питань, або подання результатів контролю з врахуванням неповних/неточних відповідей, кількості звернень до довідкової інформації, кількості затраченого часу тощо. Впровадження індивідуального типу контролю вимагає високих затрат ресурсів, але одночасно з цим дозволяє досягати більш детальних результатів.

1.3 Функції діагностики результатів навчання

Для здійснення цілеспрямованого керування процесом навчання необхідний оперативний зв'язок, за що відповідає навчально-перевіряюча функція. Перевірка знань дає інформацію про хід пізнавальної діяльності респондентів, про те, якими є досягнення кожного респондента і всього потоку, як іде процес засвоєння знань, формування навичок і умінь. Навчально-перевіряюча функція передбачає визначення рівня результатів навчання кожного респондента, потоку, а також

можливостей і готовності респондентів до подальшого просування у вивченні програмного матеріалу [6].

Навчальна функція зумовлена перевіркою результатів навчання, що сприяє кращому засвоєнню програмного матеріалу, систематизації знань, їх закріплення. Респондент вчиться критично аналізувати власні відповіді. Крім того, організація контролю сприяє повторенню, вивченню й поглибленню знань, удосконаленню навичок і умінь.

Діагностуюча і коригувальна функції передбачають з'ясування причин труднощів, які виникають у респондента під час навчання; виявлення прогалин у знаннях і уміннях; корегування навчального процесу, спрямоване на усунення цих прогалин, у діяльність як респондента, так і викладача. Перевірка одночасно є засобом виявлення ефективності методів і прийомів навчання, що застосовуються самим викладачем. Викладач запитує себе, якщо бачить погану оцінку, у чому винен сам, які треба зробити висновки для подальшої педагогічної роботи. Таким чином у процесі контролю відбувається справжнє корегування, а не «доучування». Коригування – це зміна навчальних цілей, способів навчальної діяльності, технології контролю відповідно до отриманих результатів (надолуження пробілів у здобутих знаннях, робота над помилками тощо).

Контроль має стимулювати розвиток респондентів, а для цього необхідно враховувати наявний рівень розвитку респондента, що контролює стимулюючо-мотиваційна функція. Якщо вимоги надто високі, це лише гальмуватиме розвиток. Занижені вимоги також не сприяють розвитку респондентів, тому що не активізують їх розумову діяльність. Щоб успішно стимулювати розвиток респондентів у ході контролю, необхідно, щоб останній:

- зрозумів, що від нього вимагають;
- не мав сумніву у необхідності знань;

— був упевнений, що може опанувати навчальну програму, що бажаний результат залежить виключно від прикладених зусиль.

Отже, стимулюючо-мотиваційна функція передбачає стимулювання бажання респондентів поліпшувати свої результати, розвитку відповідальності, сприяння змагальності респондентів та формування позитивних мотивів.

Контроль привчає респондентів до систематичної роботи, дисципліни, сприяє формуванню відповідальності, активності, самостійності, допомагає розібратися у собі, застосовувати прийоми самоконтролю, а також розвиває працелюбність, акуратність, активність, креативність та інші якості особистості. Разом із тим, контроль сприяє розвитку й негативних рис особистості, таких як бажання догоджати, пристосовуватися, обман, лицемірство тощо. Тому контроль потребує розумного педагогічного керівництва, певної майстерності, такту.

Розвивальна функція полягає у розвитку пам'яті, уваги, уяви, формуванні уміння працювати самостійно, розвитку способів мислення. Ця функція нерозривна з навчальною і виховною функціями. У процесі виконання завдань респонденти самостійно роблять висновки, узагальнення, застосовують знання у змінній або новій ситуаціях, навчаються виокремлювати головне, відтворювати інформацію з різними навчальними завданнями тощо.

На основі контролю можна отримати інформацію про стан успішності, успіхи і недоліки кожного респондента, і це дає змогу правильно скорегувати та спланувати подальшу роботу.

Усі названі функції взаємопов'язані та мають комплексний характер.

Підсистема перевірки результатів навчання буде ефективною за умови дотримання педагогами таких педагогічних вимог:

- системність контролю результатів навчання;
- об'єктивність і умотивованість оцінки;
- єдність вимог до оцінювання знань, навичок, умінь;

- диференційований характер контролю;
- розмаїття видів, методів і форм організації контролю;
- всебічність контролю.

Системність контролю результатів навчання. Контроль набуває ефективності лише за умови його регулярного проведення у процесі навчання. Систематична перевірка виховує у респондентів відповідальне ставлення до навчання, дозволяє виявити індивідуальні особливості і застосувати диференційований підхід у навчанні. Вона дає більш достовірну інформацію про досягнення респондентів і прогалини в їх підготовці, дозволяє оперативно керувати процесом навчання. Систематична перевірка знань сприяє виробленню у респондентів установки на тривале запам'ятовування, на заповнення прогалин у знаннях, на повторення і включення раніше здобутих знань у нову систему. Системність контролю реалізується через продуманість і послідовність його видів, доцільність форм, форм організації, методів і засобів, чіткість критеріїв оцінювання.

Об'єктивність і умотивованість оцінки результатів навчання. Кожен респондент має право на справедливе, неупереджене, об'єктивне, незалежне, недискримінаційне та добросовісне оцінювання результатів його навчання незалежно від виду та форми здобуття ним освіти. Отже, успішність кожного респондента повинна бути оціненою з максимальною точністю та об'єктивністю.

Єдність вимог до оцінювання знань, навичок, умінь. Тракткування балу і критеріїв оцінювання навчальної роботи повинні бути однакові для всіх респондентів.

Диференційований характер контролю передбачає врахування специфіки програмного матеріалу, предмета, індивідуальних особливостей респондентів.

Розмаїття видів, методів і форм організації контролю створює умови для реалізації функцій контролю, підвищення інтересу респондентів до його проведення і результатів.

Всебічність контролю. Контроль повинен охоплювати всі розділи програми, аби не перекривати позитивною оцінкою з одного розділу програми незадовільну – з іншого, що зрештою породжує прогалини в знаннях респондентів. Важливо оцінювати всебічно знання, вміння з різних питань окремої теми чи різних розділів програми як у теоретичному плані, перевіряючи рівень засвоєння знань, так і в плані застосування набутих знань на практиці, перевіряючи рівень сформованості простих умінь, навичок і складних, творчих умінь, а також оволодіння навичками самостійної роботи [7].

1.4 Аналіз недоліків та переваг існуючих математичних моделей процесу оцінювання знань респондента

Можна виділити два типи завдань при перевірці знань: дихотомічне та політомічне. Дихотомічне завдання означає, що є лише дві можливі відповіді: правильно/неправильно. Політомічне завдання – більше двох категорій, наприклад правильно/частково правильно/неправильно тощо.

Важливим дослідженням у даній сфері стала робота датського математика Г. Раша, що запропонував модель (1.1), в якій імовірність правильної відповіді i -го учасника на j -те питання визначається різницею латентних параметрів θ_i та β_j – рівня підготовленості респондента та складності завдання [8]:

$$P_{ij} = \frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}} \quad (1.1)$$

На основі розрахунків по формулі моделі Раша, можна отримати характеристичні криві завдань тесту (рис. 1.1).

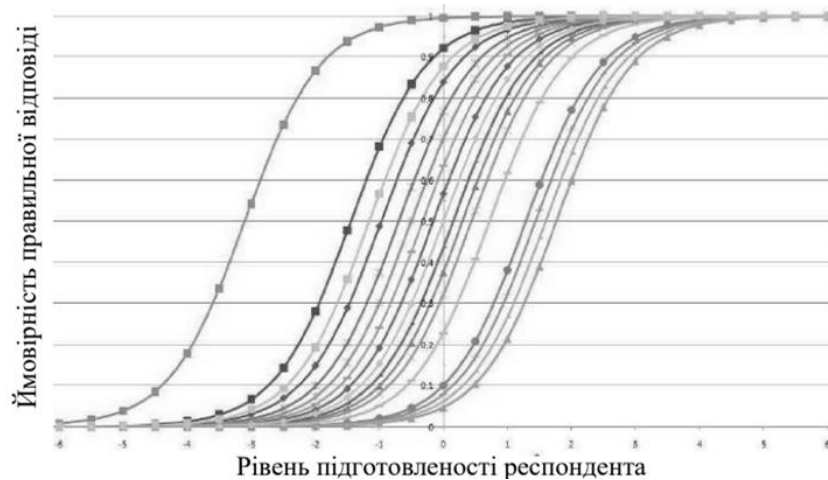


Рисунок 1.1 – Характеристичні криві завдань тесту

Раш довів, що дана математична модель встановлює зв'язок між емпіричними результатами перевірки та значеннями латентних параметрів θ_i та β_j , при цьому рівень підготовленості респондента та рівень складності завдання розміщені на одній шкалі та вимірюються в однакових одиницях – логітах.

Порівняння положень характеристичних кривих між собою дає можливість зробити висновки щодо відповідності складності та якості завдань. Наприклад, якщо криві знаходяться далеко одна від одної, потрібно додати питання з проміжною складністю. Якщо вони накладаються одна на одну – складність однакова, що означає, що одне з питань можна видалити, оскільки це не сприяє загальній інформації про тест та його складність.

Із наведеної вище залежності можна зробити висновок про те, що чим вищий рівень підготовки респондента, тим більша вірогідність дати правильну відповідь на завдання. Якщо рівень підготовленості респондента дорівнює складності завдання, то він з рівною ймовірністю може дати як правильну, так і неправильну відповідь. Дану неоднозначність можна приписати до недоліків даного підходу, оскільки при цьому не враховується ймовірність вгадування відповідей.

Дана модель названа однопараметричною («1 Parametric Logistic Latent Trait Model» – 1PL), оскільки вона описує ймовірність успіху респондента, як функцію одного параметру.

Варто зазначити, що кривизна характеристичних кривих завдання у запропонованій моделі однакова, тобто значення диференціюючої здатності є константою, що дорівнює 0,25. Для усунення цього недоліку, А. Бірнбаум увів ще один параметр – α (диференційна здатність завдання) і завдяки цьому модель (1.2) називають двопараметричною [8]:

$$P_{ij} = \frac{e^{\alpha(\theta_i - \beta_j)}}{1 + e^{\alpha(\theta_i - \beta_j)}} \quad (1.2)$$

Параметр α визначає нахил (кривизну) характеристичних кривих завдання. На основі розрахунків моделі Бірнбаума можна отримати характеристичні криві, на основі яких можна зробити висновок, що чим більше значення параметру, тим крутіше йде крива, тим більша диференціююча здатність завдання (рис. 1.2).

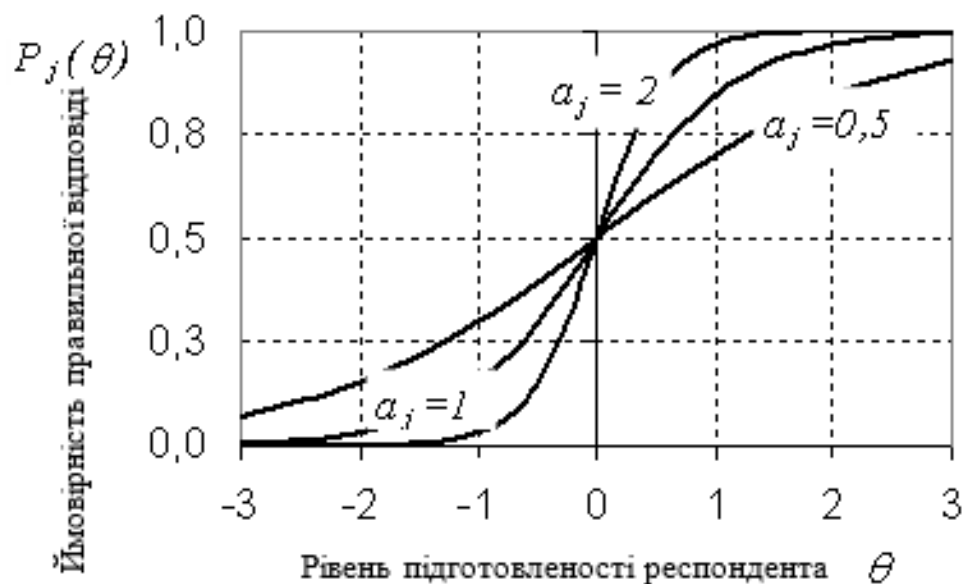


Рисунок 1.2 – Характеристичні криві двопараметричної моделі

Також, він запропонував модель (1.3), яка враховує параметр c_j можливості випадковості правильної відповіді в завданнях закритого типу, що названа трипараметричною. На рис. 1.3 побудовано характеристичні криві для трьох завдань з різними параметрами вгадування $c_j = 0$, $c_j = 0,25$ та $c_j = 0,5$ [9].

Із наведених графіків кривих видно, що наявність параметра вгадування призводить до пропорційного зміщення характеристичних кривих завдань вгору на величину c_j . Тим самим навіть зовсім невідготовлений респондент неспроможний показати нульовий результат. Диференціальна здатність тестового завдання під час введення параметра знижується. З цього випливає нетривіальний висновок: тести із «закритими» завданнями (вимушеним вибором відповіді) гірше диференціюють респондентом за рівнем знань, ніж тести із «відкритими» завданнями [8]:

$$P_{ij} = c_j + (1 - c_j) \frac{e^{\alpha(\theta_i - \beta_j)}}{1 + e^{\alpha(\theta_i - \beta_j)}} \quad (1.3)$$

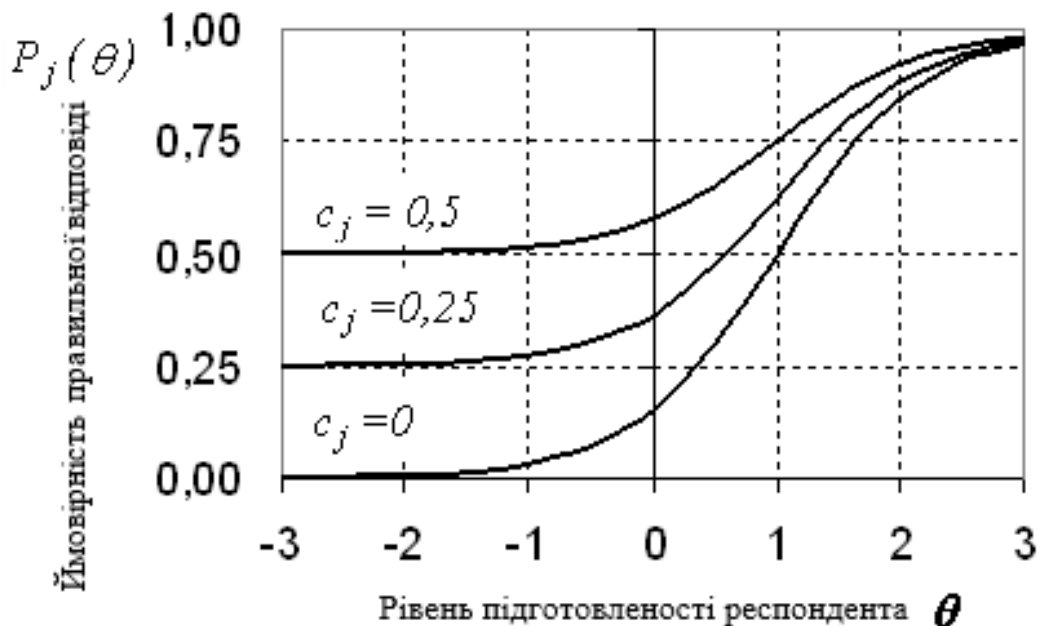


Рисунок 1.3 – Характеристичні криві трипараметричної моделі

У табл. 1.2 наведено характерні риси всіх згаданих моделей.

Таблиця 1.2 – Особливості моделей перевірки знань

Модель	Особливості моделі
Однопараметрична 1PL (Раша)	Дискримінуюча здатність для всіх завдань стала
Двопараметрична 2PL	Враховується дискримінуюча здатність завдань
Трипараметрична 3PL	Враховується дискримінуюча здатність завдань та угадування

Основним недоліком моделі Раша є те, що вона не враховує крутизну характеристичних кривих питань і за замовчуванням цей параметр вважається однаковим для всіх кривих. Зазвичай тест для перевірки рівня знань містить питання з різною диференційною здатністю, і однопараметрична модель 1PL, в цьому випадку, не може описати характеристику завдань, оскільки це константа в цій моделі, що дорівнює 0,25.

Абсолютно інша картина спостерігається для двопараметричної моделі. На рис. 1.2 це добре видно. Завдання зі значенням параметра диференціюючої складності завдання $\alpha = 0,5$ в області значень більше нуля є найважчим з представлених трьох завдань, тобто вірогідність правильної відповіді на це завдання найнижча. В області ж значень менше нуля, це саме завдання тепер уже найлегше, а вірогідність правильної відповіді на нього найбільша. Можна зробити висновок, що для слабких респондентів це найлегше завдання, а для сильних респондентів – найважче.

Аналогічні результати спостерігаються і для трипараметричної моделі. На рис. 1.3 показаний випадок непересічних характеристичних кривих, оскільки для

них вибрані однакові параметри $\beta = 1$ і $\alpha = 1$, тобто всі три завдання мають однакові характеристики складності і однакову диференціюючу здатність. Якщо припустити, що для завдання $c_j = 0$ змінити складність на $\beta = -1$, то це завдання залишиться найлегшим для найменш підготовлених респондентів, але вже в точці $\theta = -1$ його складність перевершить обидва інших завдання та в результаті буде найлегшим.

Таким чином, було з'ясовано, що двопараметрична модель Бірнбаума є більш оптимальним підходом до аналізу результатів перевірки, ніж однопараметрична модель Раша, оскільки в ній виключено головний недолік останньої, а використання трипараметричної моделі має обмеження у вигляді використання питань лише закритого типу, що є не доречним при ефективному моніторингу знань.

1.5 Вивчення комп'ютерних систем та програмних засобів оцінювання якості навчання

Дистанційне навчання зробило набагато більше, ніж просто надало доступ до знань великій кількості людей. Його вплив на професійну підготовку є революційним і складним, більшість його основних принципів перейшли з класичної навчальної моделі.

Ключові фактори, які впливають на рішення представників сфери бізнесу розглядати технології дистанційного навчання [7]:

- вартість – дистанційне навчання допомагає зменшити витрати на поїздки та дозволяє не втрачати продуктивність пов'язану з особистим навчанням;
- масштабованість – мережа та можливості Інтернету дозволяють масштабувати вміст навчальної програми до великої кількості респондентів, з численними презентаціями та навчальними матеріалами, щоб скоротити

відмінності між респондентами закладів очної освіти та дистанційної форми навчання;

— своєчасність – технології дистанційного навчання можуть бути використані для того, щоб дозволити респондентам отримати доступ до необхідних джерел, для негайного виконання знання, і це неодмінно призводить до покращення продуктивності;

— відповідність – методи та зміст доступу можуть бути налаштовані та адаптовані під потреби та контекст кожного окремого респондентів.

Для технічної підтримки системи дистанційного навчання потрібно мати досить велику кількість розроблених програмних засобів. До них належать:

- засіб розробки навчального контенту;
- система управління навчанням;
- система дистанційного навчання;
- система обміну інформацією;
- система управління контентом, навчальним порталом;

Система управління навчанням реалізує такі функції:

- облік слухачів, персоналізація, розмежування прав доступу;
- управління процесом навчання, облік результатів навчання і перевірки;
- інтеграція з механізмами синхронного і асинхронного спілкування;
- підготовка аналітичної звітності;
- інтеграція з зовнішніми інформаційними системами.

За допомогою засобів розробки навчального контенту створюються навчальні матеріали та тести, які потім поміщаються в базу даних системи управління навчанням. Через неї слухачі отримують доступ до навчальних матеріалів. Система обміну інформацією дозволяє респондентам, викладачам, експертам й іншим учасникам навчального процесу обмінюватися інформацією між собою як в режимі

реального часу так і асинхронно. Веб-інтерфейс системи управління навчанням, як правило, будується на основі засобів управління контентом.

Навчальний контент може бути статичним (HTML сторінки, тексти) і інтерактивним, з елементами анімації і голосовим супроводом.

Для створення статичного контенту можна використовувати стандартні редактори, такі як Microsoft Word. Інтерактивний контент створюється за допомогою спеціальних програмних продуктів.

Модуль обміну інформацією системи дистанційного навчання дозволяє реалізувати наступні функції (в залежності від обраного програмного забезпечення):

- асинхронне спілкування (форуми, електронна пошта);
- синхронне спілкування (голосовий і комп'ютерний чат, відеоконференції, спільне використання програмних продуктів, віртуальна аудиторія).

Memrise – це онлайн-сервіс та мобільний додаток для вивчення іноземних слів. У системі Memrise використовуються різні типи інтерактивних завдань для оцінювання рівня знань, де користувачу пропонують скласти зі слів речення (рис. 1.4), знайти зайве слово, вибрати правильний переклад тощо [21]. Система пред'являє вам певну кількість нових слів (кількість нових слів на день ви можете відрегулювати самостійно в налаштуваннях). Нова лексика пропонується вам кілька разів, а потім розпочинаються вправи. У порівнянні з іншими додатками, цей інструмент не застосовує ігрові підходи до ходу вивчення іноземних мов. Також в системі передбачена загальна статистика пройдених курсів, витраченого часу та як швидко користувач давав відповідь в залежності від часу тестування (рис. 1.5), рівень знань призначається орієнтовно, на основі пройдених курсів. Головний плюс від роботи з цим додатком – це, звичайно, його безкоштовність. До того ж, розробники вмістили у Memrise велику кількість стандартних курсів. Користувачу

необхідно лише підібрати найбільш корисний і повноцінний курс і відразу ж розпочати процес навчання. Акцент у рамках Memrise зроблено на лексику.

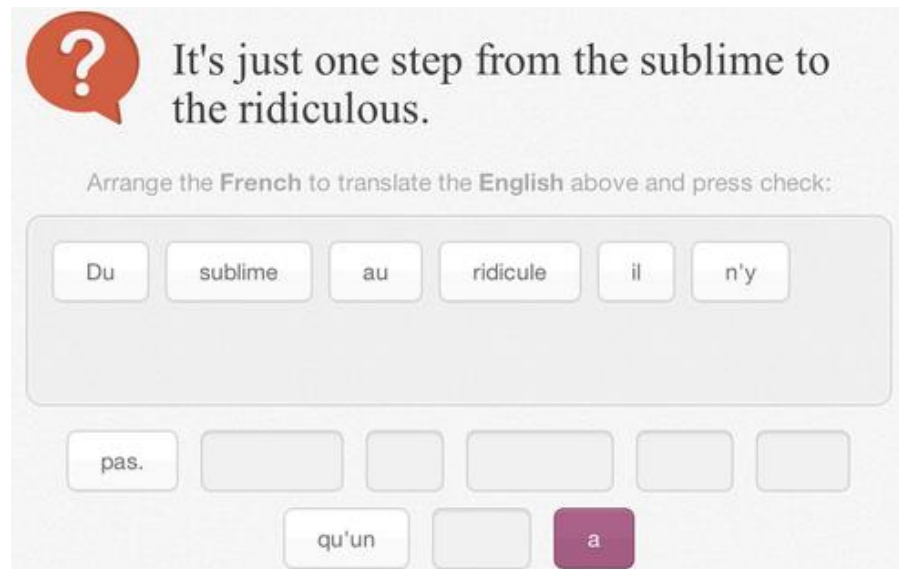


Рисунок 1.4 – Перевірка знань у додатку Memrise

Your last 10000 tests shown by the time of the day

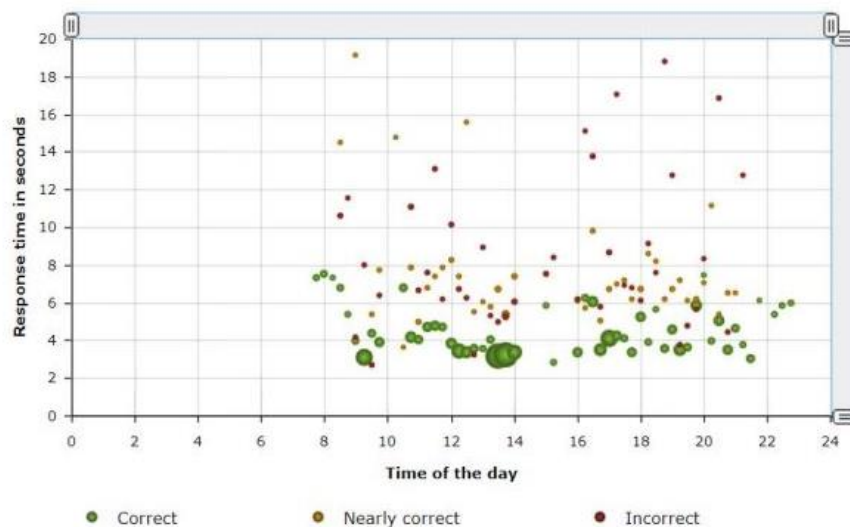


Рисунок 1.5 – Аналіз результатів навчання у додатку Memrise

Система Lingualeo є однією з найбільш комплексних в сегменті вивчення мов онлайн [12]. Матеріали включають аудіо, відео, тексти, що підбираються індивідуально для рівня знань користувача (рис. 1.6), що встановлюється шляхом проходження тестування (попереднього, поточного або підсумкового). Навчання побудоване на щоденному виконанні інтерактивних завдань з письма, аудіювання або встановлення відповідностей. Аналіз якості навчання містить багато графіків статистики, таких як кількість вивчених слів, кількість днів безперервного навчання, орієнтовний прогрес розвитку навичок письма, аудіювання, читання та граматики (рис. 1.7).

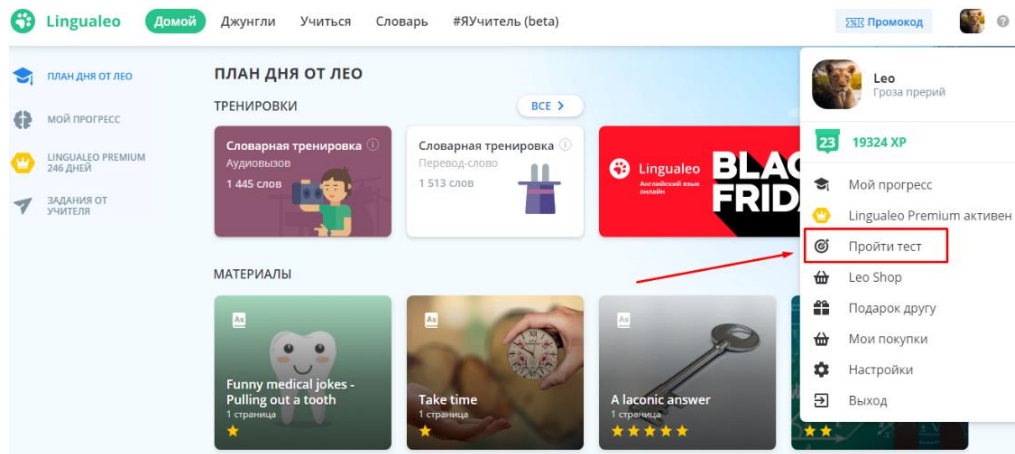


Рисунок 1.6 – Навчання у додатку Lingualeo

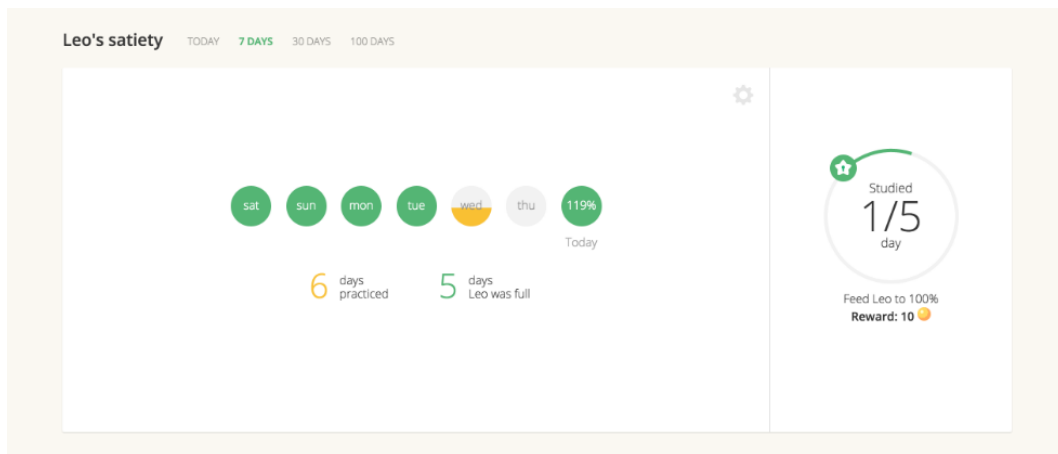


Рисунок 1.7 – Аналіз результатів навчання у додатку Lingualeo

Додаток Duolingo пропонує виконувати прості завдання в ігровій формі для вивчення мови [13]. В системі вже є можливість вивчати більше 30 найпопулярніших мов світу. Навчальні завдання складаються в переважній більшості зі складань речень зі слів, або встановлення відповідності. Також у новій версії з'явився розділ аудіювання діалогів ситуацій із життя, що дозволяє краще сприймати мову, а також паралельно вивчати граматику на слух. Повторюючи за диктором, можна покращити своє мовлення. Однак, одноманітність вправ приводить до того, що додаток підходить для вивчення слів та загальних правил граматики, але не дає глибоких знань для володіння мовою на високому рівні. Вкладка статистики носить побічний характер та не надає якогось аналізу (рис. 1.8).

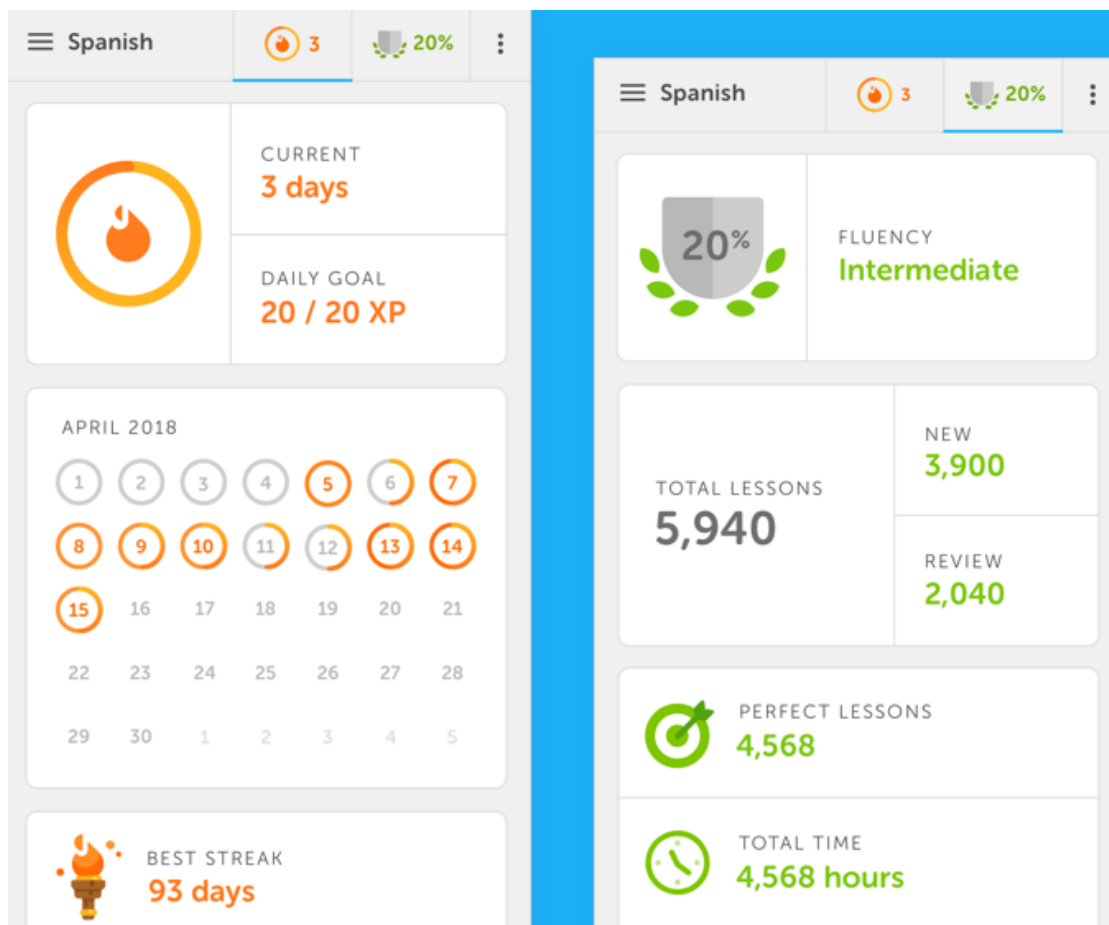


Рисунок 1.8 – Статистика результатів навчання у додатку Duolingo

Для кращого розуміння характеристик систем, що порівнюється, було проведено їхнє порівняння (табл. 1.3).

Таблиця 1.3 – Порівняльна характеристика систем-аналогів

Система	Memrise	Lingualeo	Duolingo	Система, що розробляється
Вступне тестування	-	+	+	+
Поточне тестування	+	+	+	+
Підсумкове тестування	-	+	+	+
Індивідуальне тестування	-	-	-	+
Контроль	+	+	+	+
Розширене оцінювання	-	+	+	+
Передбачена наявність викладача	-	-	-	+
Аналіз результатів навчання	-	-	-	+
Відсутність платного контенту	+	-	-	+

Усі вищеназвані системи розроблялися спеціально для того, щоб полегшити процес вивчення іноземної мови. У загальному випадку респондент обирає курс та вивчає його по спеціальному алгоритму, слідуючи за вказівками системи. Всі наявні системи, як правило, вже мають підготовлені та перевірені експертами стандартні модулі різних напрямків та рівнів складності для вивчення респондентами. Проте варто зазначити, що навіть у гігантів ринку є свої недоліки, які й будуть покращені у системі, що розробляється.

Із переваг Memrise можна виділити його повну безоплатність та автономність. Наявний необхідний мінімум вправ для перевірки знань, але він не є достатнім для конкурування з іншими системами у списку. До недоліків системи можна віднести малу різноманітність видів контролю рівня знань та відсутність аналізу результатів навчання.

Lingualo є одною з найбільш комплексних систем на ринку, але розглядатися буде саме як система для вивчення слів. До переваг системи можна віднести наявність всіх типів контролю рівня знань, окрім індивідуального. До недоліків можна віднести той факт, що система не передбачає роботу з викладачем, а краща частина контенту доступна лише на платній основі.

Duolingo є найбільш поширеною системою для вивчення іноземних мов, в основному за рахунок своєї простоти та відсутності реклами і платного контенту на своєму старті. Зараз система надає можливості для всебічного контролю рівня знань та оцінювання його результатів. Робота з викладачем не передбачена, а доступ до певної частини функціоналу доступний лише на платній основі.

Враховуючи вищесказане, перевагами системи, що розробляється, є наявність всіх типів контролю рівня знань, різноманітність вправ для перевірки знань, аналіз результатів контролю та представлення розширеної статистики респондента у вигляді інфографіків, можливість роботи з викладачем, а також відсутність реклами та платних обмежень Premium режиму.

Висновки до розділу 1

У даному розділі було наведено теоретичні засади побудови комп'ютерних систем дистанційного навчання. Встановлено їхню роль та значення для сучасної системи освіти. Розглянуто процес становлення форми дистанційного навчання як окремої та конкурентноспроможної на ринку освіти.

Проаналізовано підходи до організації процесів перевірки, оцінювання та в цілому діагностики результатів навчання. Розглянуто функції діагностики, на основі яких запропоновано підсистему оцінювання якості навчання.

Побудовано та проаналізовано переваги та недоліки існуючих математичних моделей процесу оцінювання знань респондента. Обґрунтовано вибір двопараметричної моделі Бірнбаума як основи для розробки власної математичної моделі оцінювання результатів навчання та її переваги над одно- та трипараметричними моделями.

Проведено порівняння існуючих на ринку комп'ютерних систем та доведено доцільність розробки власної системи. Вимогами до розроблюваної системи є наявність трьох основних та додаткового індивідуального типу перевірки знань, різноманітність форм завдань, аналіз результатів перевірки та представлення розширеної статистики респондента у вигляді інфографіків, можливість роботи з викладачем, а також відсутність реклами та платних обмежень.

2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПІДСИСТЕМИ ОЦІНЮВАННЯ ЯКОСТІ НАВЧАННЯ

2.1 Розроблення структури підсистеми оцінювання якості навчання

Оцінювання результатів навчальної діяльності не може обмежуватися лише двома протилежностями «успішно» чи «неуспішно». Навіть відсотка правильних відповідей недостатньо, щоб оцінити те, як респондент справився із завданням, які питання викликали труднощі, скільки часу знадобилося для вирішення завдання, скільки знадобилося спроб тощо [14]. Такі розширені результати нададуть можливість подальшого прогнозування, планування та корекції навчального процесу. Детальний аналіз дозволить сконцентрувати увагу на найбільш проблемних місцях та не витратити час на повторне вивчення добре засвоєних фрагментів матеріалу.

У підсистемі оцінювання якості навчання, що розробляється, пропонується враховувати наступні вимоги:

- виявляти обсяг, рівень, якість засвоєння навчального матеріалу;
- виявляти труднощі та помилки;
- перевіряти ефективність обраних форм організації, методів і засобів навчання;
- виявляти готовність респондентів до вивчення нового матеріалу.

Названі вище твердження можна об'єднати поняттям «діагностика» та дати визначення комплексного процесу вимірювання результатів навчання, встановлення умов і обставин протікання процесу навчання, отримання інформації про прогалини в уміннях та знаннях та про причини, які спричиняють чи перешкоджають досягненню накреслених результатів [15]. Як бачимо, у процес діагностики вкладається широкий спектр функцій, а її головна суть – розглядати

результати навчання у зв'язку зі шляхами і способами їх досягнення. Діагностика також включає виявлення динаміки, тенденцій, прогнозування подальшого розвитку освітнього процесу, задля досягнення чого потрібно мати можливість обліку попередніх результатів навчання.

Отже, структура процесу діагностики містить підсистеми контролю, оцінювання результатів контролю, їх обліку, аналізу та прогнозування. У даній роботі розробляються перші три складових, структурна схема яких зображена на рис. 2.1 [10]. Кожну з них детальніше розглянуто у наступних розділах.



Рисунок 2.1 – Структура підсистеми оцінювання якості навчання

Заявлені структурні модулі підсистеми оцінювання якості навчання, що розробляються в даній МКР, будуть взаємодіяти за двома сценаріями, як зображено на рис. 2.2:

- введення нових результатів та проходження повного циклу контроль-оцінювання-збереження;
- виведення попередніх результатів контролю та оцінювання.

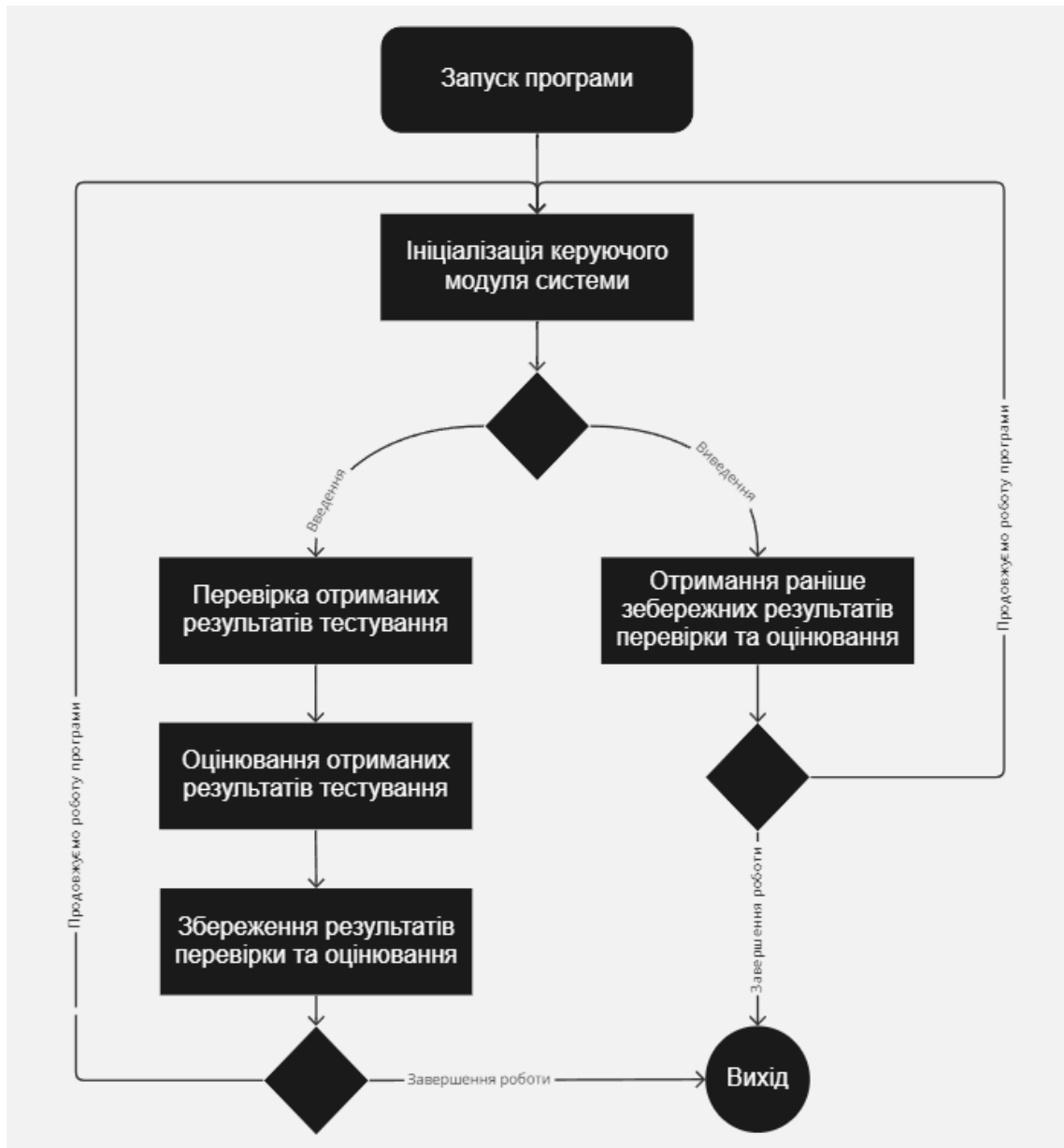


Рисунок 2.2 – UML-діаграма діяльності системи оцінювання якості навчання

На рис. 2.3 зображено UML-діаграму класів системи оцінювання якості навчання. Вона складається з таких класів:

- клас контролера системи оцінювання;
- клас сервісу контролю навчання;
- клас сервісу оцінювання навчання;



Рисунок 2.3 – UML-діаграма класів системи оцінювання якості навчання

- клас сервісу обліку результатів навчання;
- клас моделі контролю навчання;
- клас моделі оцінювання навчання;
- клас моделі обліку результатів навчання.

EvaluationController – клас, що керує взаємодією між користувачем системи та її сервісами, виконує маршрутизацію HTTP запитів. Він отримує в користування сервіси `_verificationService`, `_assessmentService` та `_accountingService` при ініціалізації. Метод `LearningResultModel GetResults(int userId)` повертає модель обліку результатів навчання для конкретного користувача. Метод `IResult Verify(LearningVerificationModel model)` приймає по HTTP модель контролю

навчання та виконує процес контролю, та запускає процес оцінювання якості навчання. Метод `void Assess(LearningAssessmentModel model)` виконує процес оцінювання та запускає процес обліку результатів навчання. Метод `void Account(LearningAssessmentModel model)` виконує процес обліку результатів навчання.

`LearningVerificationService` – клас, що здійснює функції першого контролю результатів навчання, отриманих від користувача. Взаємодію з контекстом даних виконує через репозиторій, доступ до якого отримує через `Unit Of Work`. Метод `LearningAssessmentModel Verify(LearningVerificationModel verificationModel)` відповідає за формування та наповнення даними моделі оцінювання якості навчання. Метод `Dictionary<int, double> CheckAnswersByTopic (LearningVerificationModel model)` перевіряє отримані відповіді на правильність. Метод `double CalculateTime(LearningVerificationModel model)` рахує середній затрачений час на одну відповідь.

`LearningAssessmentService` – клас, що здійснює функції оцінювання якості навчання. Взаємодію з контекстом даних виконує через репозиторій, доступ до якого отримує через `Unit Of Work`. Метод `LearningAssessmentModel Assess(LearningAssessmentModel model)` відповідає за наповнення моделі даними оцінювання якості навчання. Метод `Dictionary<int, double> CalculateTopicsPercentage (LearningAssessmentModel model)` вираховує відсоток даних правильних відповідей для кожної теми. Метод `IEnumerable<int> CalculateTopicRating(LearningAssessmentModel model)` виконує перерахунок загального рейтингу вивчених тем користувача за відсотком даних правильних відповідей.

`LearningAccountingService` – клас, що здійснює функції обліку результатів навчання. Взаємодію з контекстом даних виконує через репозиторій, доступ до якого отримує через `Unit Of Work`. Метод `LearningResultModel GetUserResults(int`

userId) повертає модель з результатами користувача по його ідентифікатору. Метод `void Account(LearningAssessmentModel model)` здійснює запис даних моделі оцінювання навчання у базу даних.

`LearningVerificationModel` – модель даних для контролю результатів навчання. `Int UserId` – ідентифікатор користувача. `Dictionary<int, (int, string)> GivenAnswers` – структура даних, що містить дані користувачем відповіді у форматі ключ-значення. `DateTime TimeSpent` – час, затрачений на всі відповіді.

`LearningAssessmentModel` – модель даних для оцінювання результатів навчання. `Int UserId` – ідентифікатор користувача. `Double CorrectAnswerPercentage` – відсоток правильних відповідей. `Dictionary<int, double> TopicPercentage` – структура даних, що містить дані про відсоток правильних відповідей для кожної теми у форматі ключ-значення. `IEnumerable<int> TopicRating` – структура даних, що містить ідентифікатори тем, відсортовані в залежності від відсотка правильних відповідей. `DateTime TimeSpent` – час, затрачений на всі відповіді. `Double AvgSecondsPerAnswer` – середній час, затрачений на одну відповідь.

`LearningResultModel` – модель даних для обліку результатів навчання. `Int UserId` – ідентифікатор користувача. `Double CorrectAnswerPercentage` – відсоток правильних відповідей. `Dictionary<int, double>? TopicPercentage` – структура даних, що містить дані про відсоток правильних відповідей для кожної теми у форматі ключ-значення. `IEnumerable<int> TopicRating` – структура даних, що містить ідентифікатори тем, відсортовані в залежності від відсотка правильних відповідей. `DateTime? TimeSpent` – час, затрачений на всі відповіді. `DateTime? TotalTimeSpent` – час, затрачений на всі відповіді за весь час. `Double AvgSecondsPerAnswer` – середній час, затрачений на одну відповідь.

2.2 Побудова моделі перевірки результатів навчання

Перевірки результатів навчання – обов'язкова частина процесу навчання, що вимірює результати навчальної роботи, забезпечує зворотний зв'язок та є засобом коригування навчального процесу.

Підсистема використовує тестову форму такого процесу та три види перевірки результатів:

- попередній, що здійснюється безпосередньо після реєстрації нового користувача;
- поточний, що здійснюється після кожного тесту;
- підсумковий, що охоплює весь пройдений матеріал [16].

Для компонента перевірки результатів навчання в магістерській роботі пропонується модель (2.1), яка оцінює відповідь суб'єкта тестування за двобальною шкалою (правильно або неправильно):

$$R = \frac{\sum_{i=1}^k R_i}{n}, \quad (2.1)$$

де R_i – правильна відповідь суб'єкта тестування на i -е завдання;

k – кількість правильних відповідей із n запропонованих ($k \leq n$).

До переваг даної моделі варто віднести простоту її реалізації.

2.3 Створення моделі оцінювання якості навчання

У педагогіці існує таке поняття, як рівень засвоєння матеріалу, що відображає дві складові дидактичного процесу: пізнавальну діяльність студента та

управлінську діяльність викладача цим процесом. Пошук об'єктивних показників засвоєння дозволить кількісно охарактеризувати обидві складові такого процесу. Засвоєння як об'єкт виміру характеризується певною предметною дійсністю, що розкривається шляхом опису діяльності, яку опановує респондент через зміст навчального предмету. У даному випадку ми розглядаємо засвоєння як результат дидактичного процесу, а управлінцем слугує сама система, а не викладач. У той же час засвоєння є самим змістом навчального процесу, як певна поетапна діяльність респондента, спрямована на оволодіння знаннями та навичками [17].

Цей двоякий характер поняття засвоєння (процес і результат) є тією межею, на якій зустрічаються психологія та педагогіка у пошуках аналізу управління процесом навчання. Якщо психологічна сторона характеризує переважно сам процес, то дидактична – його результат.

Та обставина, що процес засвоєння та його результат взаємообумовлені, вимагає інтегративного психолого-педагогічного підходу до розробки критеріїв оцінювання якості навчання. Намагаючись оцінити факт засвоєння знань, ми маємо створити респонденту відповідні умови, за яких він міг би виявити вміння ними користуватися [18].

Тому додатково для покращення процесу контролю був розроблений алгоритм, базований на класифікації завдань за їх дидактичними характеристиками: значимість (z), складність (d), специфікація (c). Число балів, отриманих респондентом за виконання n завдань вираховується по формулі:

$$y = \sum_{i=1}^n w_i R_i, \quad (2.3)$$

де R_i – оцінка за виконання i -го завдання;

n – число завдань;

w_i – вектор вагових коефіцієнтів завдань, що залежить від їхніх дидактичних характеристик.

По завершенню перевірки встановлюється середній бал A , отриманий респондентом за виконання n завдань:

$$A = \frac{y}{k_n}, \quad (2.4)$$

де y – загальне число балів, отриманих респондентом;

k_n – кількість спроб виконання n завдань, $k_n \geq n$.

На основі попередньої формули, пропонується ввести значення уточненого середнього балу A' :

$$A' = A \times r + a_1 \frac{k_n}{n} + a_2 \frac{k_c}{n} + a_3 \frac{k_b}{n}, \quad (2.5)$$

де r – коефіцієнт респондента, який призначається після проходження попередньої перевірки знань;

k_n – кількість спроб виконання n завдань;

k_c – кількість звернень до довідкової інформації;

k_b – кількість завдань, виконаних з перевищенням відведеного часу ($k_b \leq n$);

a_1, a_2, a_3 – вагові коефіцієнти, задаються статично в системі (0...1).

Після проведення обчислень значення результатів за допомогою вектора граничних значень переводяться у звичайну стобальну шкалу. Параметри контролю, а саме значення вагових коефіцієнтів w_i , коефіцієнтів a_i та значення елементів вектора граничних значень встановлюються на початковому етапі

користування системою, або за результатами попередньої перевірки рівня знань респондента.

2.4 Моделювання процесу обліку результатів навчання

Модель для обліку результатів навчання – це фактично модель даних. Модель даних – це концептуальне подання для вираження та передачі бізнес-вимог. Вона наочно показує характер даних, бізнес-правила, що управляють даними, і те, як дані будуть організовані в базі даних [19].

Моделювання даних – одне з найважливіших завдань при розробці програмного застосунку. Воно закладає основу для зберігання, вилучення та подання даних. У даній роботі використовується реляційна модель даних.

Реляційна модель – сукупність даних, що складається з набору двовимірних таблиць. В теорії множин таблиці відповідає термін відношення (relation), фізичним уявленням якого є таблиця, звідси і назва моделі – реляційна. У порівнянні з ієрархічною та мережевою моделлю даних, реляційна модель відрізняється вищим рівнем абстракції даних. Реляційна модель є зручною і звичною формою представлення даних, так у час ця модель є фактичним стандартом, який орієнтуються майже всі сучасні комерційні СУБД. На реляційній моделі даних будуються реляційні бази даних.

При табличній організації даних відсутня ієрархія елементів. Рядки та стовпці можуть бути прочитані в будь-якому порядку, тому даній структурі властива гнучкість вибору будь-якої підмножини елементів у рядках та стовпцях. Будь-яка таблиця в реляційній базі складається з рядків, які називають записами, та стовпців, які називають полями. На перетині рядків та стовпців знаходяться конкретні значення даних. До кожного поля визначається безліч його значень [8].

Переваги використання реляційної моделі:

- інформація зберігається в доступному для людини форматі (таблиця);
- модель даних ґрунтується на строгому математичному виразі, що дозволяє лаконічно описувати необхідні операції над даними;
- незалежність даних в таблиці від змін в програмі, що ними користується;
- для роботи з моделлю не обов'язково повністю знати як організована база даних.

Недоліки використання реляційної моделі:

- відносно повільний доступ до даних;
- складність створення бази даних, що використовує реляційну моделі;
- складнощі, пов'язані з реалізацією складних відносин між таблицями.

Щоб розробити модель даних, спочатку необхідно зібрати бізнес-вимоги, які буде мати система:

- можливість збереження особистої інформації користувачів;
- можливість збереження слів, їх перекладу та належності до певного модуля;
- слова, об'єднані в модулі, повинні бути доступними користувачу, який їх додав;
- слова та модулі повинні мати параметр, що буде відображати статус засвоєння матеріалу.

Наступним кроком буде створення концептуальної моделі даних. На даному етапі потрібно ідентифікувати сутності, що уособлюють атомарні дані та побудувати попередню модель даних. Зважаючи на вимоги, можна виділити 5 окремих сутностей: Користувач, Модуль, Слово, Здоров'я, Стан здоров'я. Зв'язки між сутностями вказані на рис. 2.4.

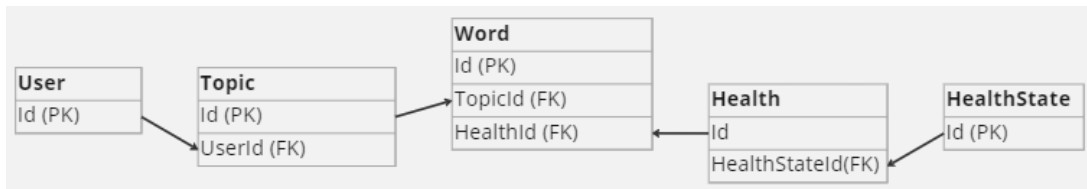


Рисунок 2.4 – Концептуальна модель даних

Далі необхідно завершити роботу над атрибутами для сутностей, наповнивши модель. У випадку даної системи, додаються атрибути, як зображено на рис. 2.5.

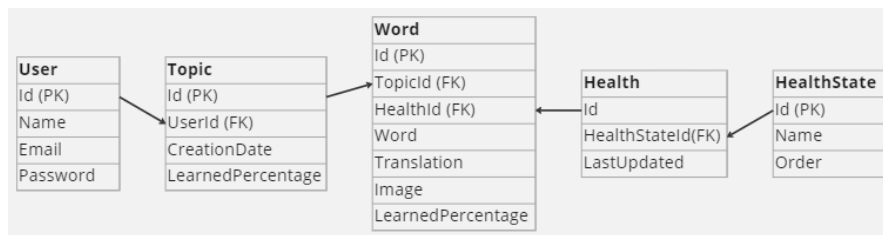


Рисунок 2.5 – Логічна модель даних

Після формалізованого опису моделі необхідно реалізувати їх за допомогою відповідного програмного коду. Детальніше про процес створення бази даних йдеться у розділі 3.

Висновки до розділу 2

У даному розділі магістерської кваліфікаційної роботи здійснено моделювання та проектування підсистеми оцінювання якості навчання.

Запропоновано загальний підхід до проведення дослідження, насамперед, було розроблено та детально описано внутрішню структуру підсистеми оцінювання якості навчання.

На основі існуючих математичних моделей, було побудовано модель перевірки результатів навчання на основі лінійної моделі, вихідний параметр якої є вхідним параметром моделі процесу оцінювання якості навчання. Модель процесу оцінювання якості навчання, у свою чергу, розраховує уточнені результати перевірки знань респондента, враховуючи вагові коефіцієнти як виконаних завдань, так і метрик респондента.

Запропоновано концептуальну та логічну моделі даних для процесу обліку результатів навчання та організації роботи з ними.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОЦІНЮВАННЯ ЯКОСТІ НАВЧАННЯ

3.1 Обґрунтування вибору технологій розробки та мов програмування

У веб-розробці використовується багато різних технологій та готових продуктів: це мови програмування та розмітки у «чистому» вигляді, бази даних та інші сховища інформації, програмні продукти для спрощення розробки (фреймворки та системи керування сайтами), а також серверне програмне забезпечення. Як видно на рис. 3.1, архітектура сучасного сайту комплексна та складається з багатьох компонентів.

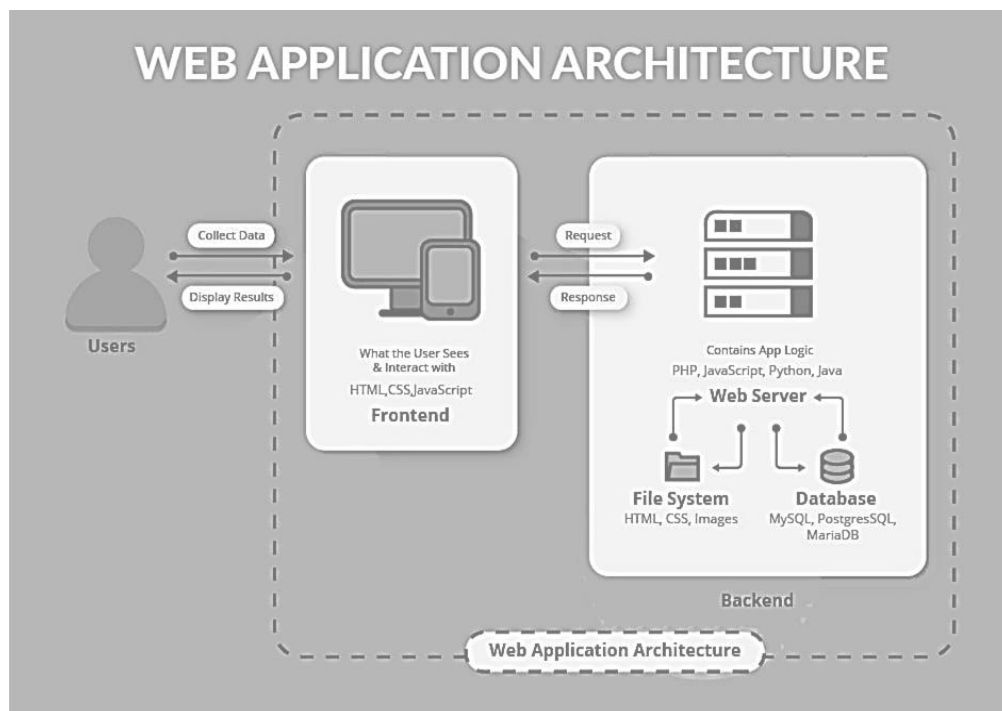


Рисунок 3.1 – Архітектура сучасного веб-додатку

Інтерфейс. Це те, що бачить користувач, коли заходить на сайт. Реалізується за допомогою HTML, CSS та JS.

Програмна частина. Це алгоритми, призначені для обробки запитів користувача. По суті, програмна частина створює або отримує з бази даних інформацію, яку запитує користувач, а також приймає та обробляє дані від користувачів. Програмна частина пишеться серверними мовами програмування і, зазвичай, з урахуванням платформи розробки.

База даних. Це місце для зберігання інформації, яка використовується на сайті. У базі даних зберігається зміст усіх сторінок сайту та їх взаємозв'язку, особисті файли користувачів системи та багато іншого. За запитом програмної частини база даних знаходить та віддає необхідні записи (наприклад, інформацію, що відображається на сторінці) або записує нові дані (наприклад, зміна паролю для входу).

Хостинг. Це сервер (комп'ютер), на якому працює програмна частина сайту і зберігається база даних. Хостинг буває різним, від його виду залежить швидкість та надійність роботи сайту [20].

Власне вимоги до компонентів досить передбачувані:

- інтерфейс повинен бути приємним і зручним для сприйняття, коректно відображатись у різних браузерах і на різних пристроях;
- програмна частина повинна відповідати на запити швидко та без помилок;
- база даних повинна надійно зберігати інформацію та швидко обробляти запити;
- сервер повинен працювати без перебоїв, не гальмувати та витримувати багато одночасних звернень.

Відштовхуючись від завдання та вимог, було обрано відповідні технології розробки та мови програмування. Для розробки інтерфейсу – фреймворк Angular на Typescript, для програмної частини – .Net Core Web API на C#, база даних – SQL

Server. Середовища розробки відповідно Visual Studio Code, Visual Studio 2022 та Microsoft SQL Management Studio 17.

Angular – фреймворк від компанії Google для створення клієнтських програм. Насамперед він орієнтований на розробку SPA-рішень (Single Page Application), тобто односторінкових додатків. У цьому плані Angular є спадкоємцем іншого фреймворку AngularJS. У той же час, Angular це не нова версія AngularJS, а принципово новий фреймворк.

Angular надає таку функціональність як двостороннє зв'язування, що дозволяє динамічно змінювати дані в одному місці інтерфейсу при зміні даних моделі в іншому, може реалізовувати шаблони та маршрутизацію. Загалом схема роботи веб-додатку на основі Angular зображена на рис. 3.2.

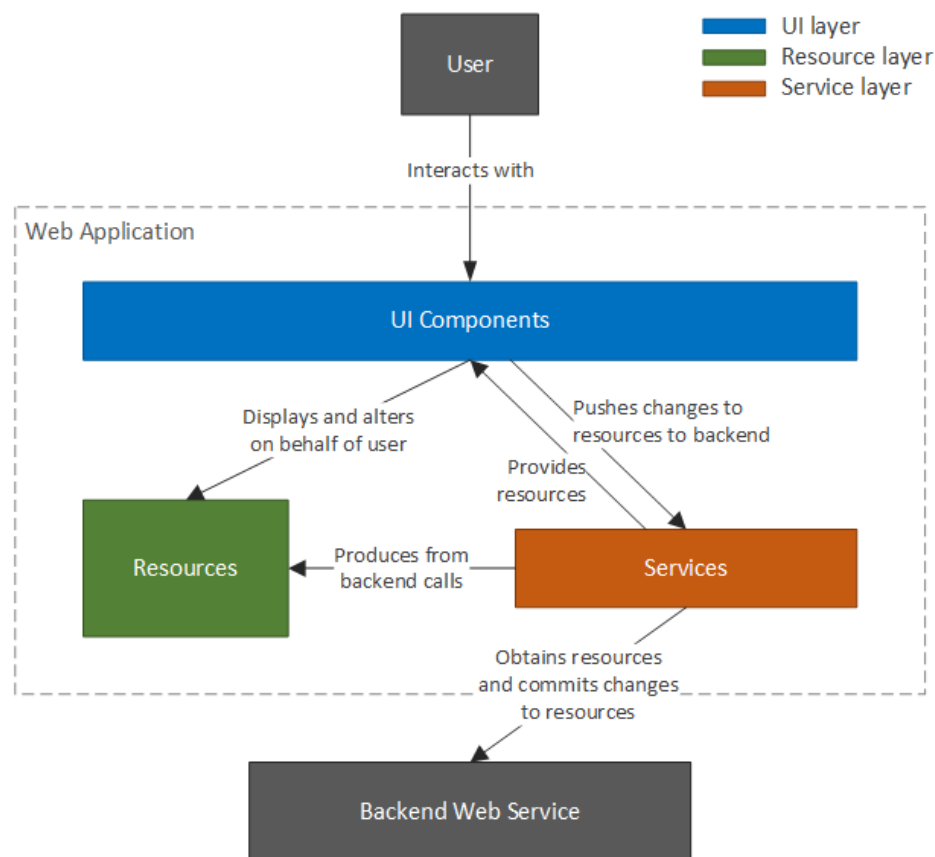


Рисунок 3.2 – Схема веб-додатку на Angular

Однією з ключових особливостей Angular є те, що він використовує мову програмування TypeScript.

Слід зазначити, що TypeScript – це строго типізована мова, чим є легшою для сприйняття програмістів Java, C# та інших строго типізованих мов. На виході компілятор створює той самий JavaScript, який потім виконується браузером. Однак сувора типізація зменшує кількість потенційних помилок, які можуть виникнути при розробці на JavaScript.

TypeScript реалізує багато концепцій, які властиві об'єктно-орієнтованим мовам, як, наприклад, успадкування, поліморфізм, інкапсуляція та модифікатори доступу тощо.

Потенціал TypeScript дозволяє швидше та простіше писати великі складні комплексні програми, відповідно їх легше підтримувати, розвивати, масштабувати та тестувати, ніж на стандартному JavaScript.

Крім того, він є кросплатформним, а це означає, що для розробки ми можемо використовувати як Windows, так і MacOS або Linux.

У той же час TypeScript є типізованим надмножиною JavaScript, а це означає, що будь-яка програма JS є програмою на TypeScript. У TS можна використовувати всі ті конструкції, які застосовуються в JS – ті ж оператори, умовні, циклічні конструкції. Більше того, код на TS компілюється в JavaScript. Зрештою, TS – це лише інструмент, покликаний полегшити розробку додатків.

Але ми не обмежені мовою TypeScript. За бажанням можемо писати програми на Angular за допомогою таких мов як Dart або JavaScript. Однак TypeScript є основною мовою для Angular [11].

ASP.NET Core – технологія для створення веб-додатків на платформі .NET, яку розвиває компанія Microsoft. Як мови програмування розробки додатків на ASP.NET Core використовуються C# і F#.

Мова програмування С# на сьогоднішній момент одна з найпотужніших мов, що швидко розвиваються і затребуваних в ІТ-галузі. Зараз на ньому пишуться різні програми: від невеликих десктопних програм до великих веб-порталів і веб-сервісів, що обслуговують щодня мільйони користувачів.

С# є об'єктно-орієнтованою і в цьому плані багато перейняла у Java та С++. Наприклад, С# підтримує поліморфізм, успадкування, навантаження операторів, статичну типізацію. Об'єктно-орієнтований підхід дозволяє вирішити завдання з побудови великих, але в той же час гнучких, масштабованих і додатків, що розширюються. С# продовжує активно розвиватися і з кожною новою версією з'являється все більше функціональностей [22-25].

Поточну архітектуру платформи ASP.NET Core зображено на рис. 3.3.

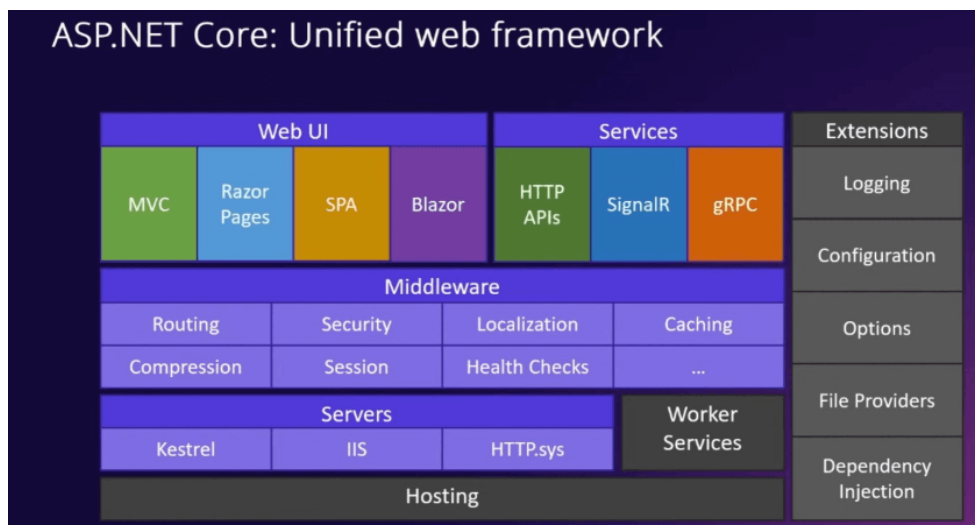


Рисунок 3.3 – Архітектура ASP.NET Core

На верхньому рівні розташовуються різні моделі взаємодії з користувачем. Це технології побудови інтерфейсу користувача та обробки введення користувача, як MVC, Razor Pages, SPA (Single Page Application – односторінкові програми з використанням Angular, React, Vue) і Balzor. Крім того, це сервіси у вигляді вбудованих HTTP API, бібліотеки SignalR або сервісів GRPC.

Всі ці технології базуються та/або взаємодіють із чистим ASP.NET Core, який представлений насамперед різними вбудованими компонентами middleware – компонентами, які застосовуються для обробки запиту. Крім того, технології вищого рівня також взаємодіють з різними розширеннями, які не є безпосередньою частиною ASP.NET Core як розширення для логування, конфігурації і т.д.

І на нижньому рівні додаток ASP.NET Core працює в рамках деякого веб-сервера, наприклад Kestrel, IIS, бібліотеки HTTP.sys.

ASP.NET Core Web API представляє реалізацію патерну REST, у якому кожному за типу http-запиту (GET, POST, PUT, DELETE) призначений окремий ресурс. Такі ресурси визначаються у вигляді методів контролера Web API. Ця модель особливо підходить для односторінкових програм [13].

SQL Server є однією з найпопулярніших систем управління базами даних (СУБД) у світі. Ця СУБД підходить для різних проєктів: від невеликих додатків до великих високонавантажених проєктів.

SQL Server було створено компанією Microsoft. SQL Server довгий час був виключно системою керування базами даних для Windows, проте починаючи з версії 16 ця система доступна і на Linux.

Ключові особливості SQL Server:

- продуктивність, SQL Server працює дуже швидко;
- надійність та безпека, SQL Server надає шифрування даних;
- простота, з цією СУБД легко працювати та вести адміністрування.

Центральним аспектом у MS SQL Server, як і у будь-якій СУБД, є база даних. База даних представляє сховище даних, організованих певним способом. Нерідко фізично база даних представляє файл на жорсткому диску, хоча така відповідність не обов'язкова. Для зберігання та адміністрування баз даних застосовуються системи управління базами даних або СУБД. І саме MS SQL Server є однією з таких СУБД.

Для організації баз даних MS SQL Server використовує реляційну модель. Реляційна модель передбачає зберігання даних у вигляді таблиць, кожна з яких складається з рядків та стовпців. Кожен рядок зберігає окремий об'єкт, а у стовпці розміщуються атрибути цього об'єкта.

Для ідентифікації кожного рядка у межах таблиці застосовується первинний ключ (primary key). Як первинний ключ може виступати один або кілька стовпців. Використовуючи первинний ключ, ми можемо посилатися на певний рядок у таблиці. Відповідно два рядки не можуть мати один і той же первинний ключ. Через ключі одна таблиця може бути пов'язана з іншою, тобто між двома таблицями можуть бути організовані зв'язки. А сама таблиця може бути представлена у вигляді відношення («relation»).

Для взаємодії з базою даних використовується мова SQL (Structured Query Language). Клієнт (наприклад, зовнішня програма) надсилає запит мовою SQL за допомогою спеціального API. СУБД належним чином інтерпретує та виконує запит, а потім надсилає клієнту результат виконання.

Виділяються два різновиди мови SQL: PL-SQL та T-SQL. PL-SQL використовується у таких СУБД як Oracle та MySQL. T-SQL (Transact-SQL) застосовується у SQL Server [26].

3.2 Проектування програмних засобів підсистеми оцінювання якості навчання

Існує безліч різних видів та типів архітектур, які успішно застосовуються. Однією з найбільш використовуваних є класична тривінева система, яка має на увазі поділ програми на три рівні, як зображено на рис. 3.4.

Це клієнт-серверна архітектура, в якій окремо виділені функції представлення, обробки та зберігання даних. Такий архітектурний патерн пом'якшує зростаючу складність додатків, спрощує і робить більш гнучким їхнє масштабування.

Така архітектура складається з різних рівнів, кожен із яких відповідає окремій функції. Таким чином, вносити зміни до кожного окремого рівня простіше, ніж займатися всією архітектурою. Можна створювати гнучкі та повторно використовувані додатки. Це значно полегшує весь процес управління системою.

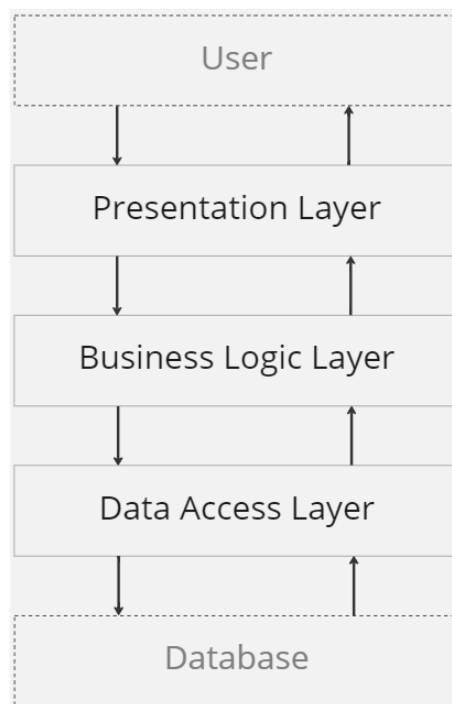


Рисунок 3.4 – Трирівнева архітектура системи

Рівень представлення – це перший і найвищий рівень, який є у додатку. Цей рівень є UI, тобто подання контенту кінцевому користувачеві через графічний інтерфейс. До цього рівня можна отримати доступ через будь-який тип клієнтського пристрою, наприклад настільний комп'ютер, ноутбук, мобільний телефон тощо.

Рівень бізнес-логіки – сервісний шар чи рівень управління. Також на цьому рівні працює бізнес-логіка програми. Бізнес-логіка – це набір правил, необхідних для роботи програми відповідно до встановлених принципів. Компоненти цього рівня зазвичай працюють на одному або кількох серверах програм.

Рівень доступу до даних – це ядро архітектури. В основному цей шар пов'язаний зі зберіганням та видаленням даних програми. Дані програми зазвичай зберігаються на сервері бази даних, файловому сервері або будь-якому іншому пристрої або носії, які підтримують логіку доступу до даних. Це робиться на рівні даних, надаючи API для рівня програми. Надання цього API забезпечує повну прозорість операцій з даними, які виконуються на цьому рівні, не торкаючись рівня програми. Наприклад, оновлення систем на цьому рівні не впливають на рівень програм цієї архітектури [27].

Взявши до уваги вищесказане, система буде поділена на 3 окремих рівні, як зображено на рис. 3.5, кожен з яких буде представлений окремим проєктом або бібліотекою та мати свої функції.

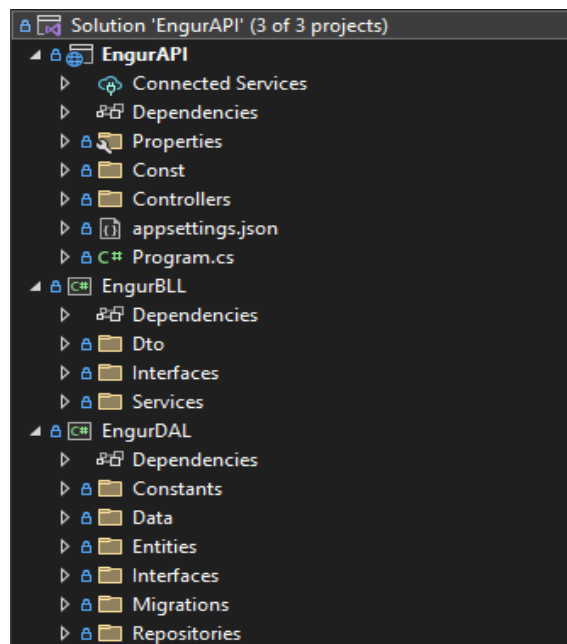


Рисунок 3.5 – Структура файлів системи

Рівень доступу до даних інкапсулюється dll-бібліотекою EngurDAL. На цьому рівні зберігаються моделі, що описують сутності даних. Також тут розміщуються специфічні класи та інтерфейси, необхідні для роботи з технологіями доступу до бази даних EntityFramework. Для інкапсуляції логіки роботи з джерелом даних використовується патерн «Репозиторій», що дозволяє абстрагуватися від конкретних підключень до джерел даних, з якими працює програма, і є проміжною ланкою між класами, що безпосередньо взаємодіють з даними, та іншою програмою. Технічне завдання системи передбачає оперування багатьма сутностями і моделями, для управління якими створюється безліч класів-репозиторіїв. Використання патерну Unit of Work дозволить спростити роботу з різними репозиторіями і дасть впевненість, що всі репозиторії використовуватимуть один і той самий контекст даних.

Рівень бізнес-логіки являє собою ще одну dll-бібліотеку EngurBLL, що містить набір компонентів, які відповідають за обробку отриманих від рівня представлення даних, реалізує всю необхідну логіку програми, всі обчислення, взаємодіє з базою даних та передає рівню уявлення результат обробки. Для кожної сутності створено окремий сервіс, відповідний інтерфейс та DTO для коректної передачі даних між різними рівнями.

Рівень представлення реалізований як Web API додаток EngurAPI. Це рівень, з яким безпосередньо відбувається взаємодія користувача за допомогою клієнтського застосунку. Головним атрибутом даного рівня є контролери. Контролери містять методи, що обробляють вхідні HTTP-запити та надсилають відповідь клієнту. Деякі методи захищені від неавторизованого використання та можуть бути викликані лише за умови автентифікації користувача відповідної ролі. Крім того, на даному рівні відбувається основна конфігурація додатку: підключення до бази даних, реєстрація сервісів, контролерів, маршрутизації.

Проектування бази даних здійснюється безпосередньо в середовищі програмування .NET за допомогою технології Entity Framework Core. Вона дозволяє взаємодіяти з СУБД за допомогою сутностей (entity), тобто класів та об'єктів NET, а не таблиць бази даних. Це найвідоміший і найфункціональніший ORM-інструмент у C#.

ORM має кілька підходів.

Перший – Code First. Він має на увазі, що спочатку пишеться код на C #, а потім з цього коду створюється база даних. Для цього підходу дуже важливо визначити класи моделі або entity, яка буде зберігатися в базі даних, описати її в класах C# у вигляді моделі, і написати клас контексту, який і працюватиме з базою даних, що використовується. Підхід Code First найчастіше використовується програмістами C# та буде використано в даній роботі.

Другий підхід – Database First підходить для тих, хто добре знає SQL, але в цьому випадку необов'язково добре знати C#. Насамперед створюється база даних, потім генерується EDMX-модель бази даних. У цьому XML у файлі .edmx міститься інформація про структуру бази, модель даних та мапінг їх один на одного. У Visual Studio є графічний дизайнер, за допомогою якого можна працювати з .edmx

Model First – третій підхід ORM. Його часто використовують архітектори, тому що при цьому підході можна не знати ні SQL, ні синтаксис C#. У цьому випадку спочатку створюється графічна модель EDMX, у цей час у фоновому режимі створюються класи C# моделі, а потім генерується база даних на основі діаграми EDMX [26].

Отже, в попередньому розділі було створено папку для зберігання моделей – Entities. Усі таблиці бази даних визначаються в Entity Framework як класи моделей чи сутностей за принципом 1 таблиця, наприклад users, – 1 клас у NET, наприклад, User. Такі пари називають умовностями, і вони визначені у класі контексту даних як набори DbSet і такий підхід працює за замовчуванням.

Створимо в даній папці новий клас з назвою Word, який буде представляти модель слова в системі:

```
public class Word
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Translation { get; set; }
    public int TopicId { get; set; }
}
```

Модель слова має чотири властивості: унікальний ідентифікатор, саме слово, його переклад та ідентифікатор батьківського модуля.

Також додамо в папку Entities новий клас, що буде представляти модуль – фактично набір пов’язаних слів:

```
public class Topic
{
    public int Id { get; set; }
    public string Name { get; set; }
    public IEnumerable<Word>? Words { get; set; }
}
```

Модель модуля має три властивості: унікальний ідентифікатор, назва модуля та деяка множина слів.

Для подальшого впровадження Entity Framework необхідно за допомогою менеджера пакетів NuGet Visual Studio [31] встановити для проєкту пакети

Microsoft.EntityFrameworkCore, Microsoft.EntityFrameworkCore.SqlServer та Microsoft.EntityFrameworkCore.Tools (рис 3.6).

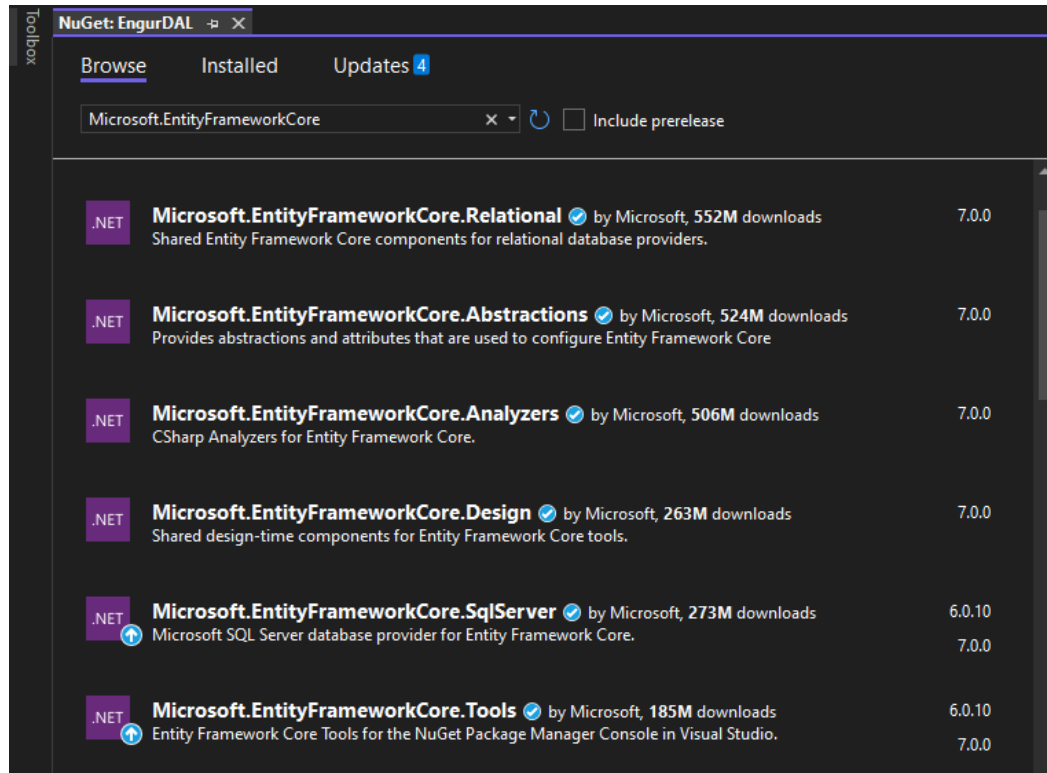


Рисунок 3.6 – Менеджер пакетів NuGet

Для взаємодії з базою даних використовується контекст даних (лістинг 3.3). Щоб створити контекст, ми повинні успадкувати новий клас від класу `DbContext`. Властивості на кшталт `public DbSet Words { get; set; }` допомагають отримувати з БД набір даних певного типу (наприклад, набір `Word` об'єктів). Фактично, кожна властивість `DbSet` буде співвідноситися з окремою таблицею в базі даних. `Entity Framework` враховує, що бази даних може бути ще не створена. Тому в його конструкторі визначено виклик `Database.EnsureCreated()`, який за відсутності бази даних автоматично створює її. Якщо база даних вже є, нічого не відбувається:

```
public class EngurDbContext : DbContext
```

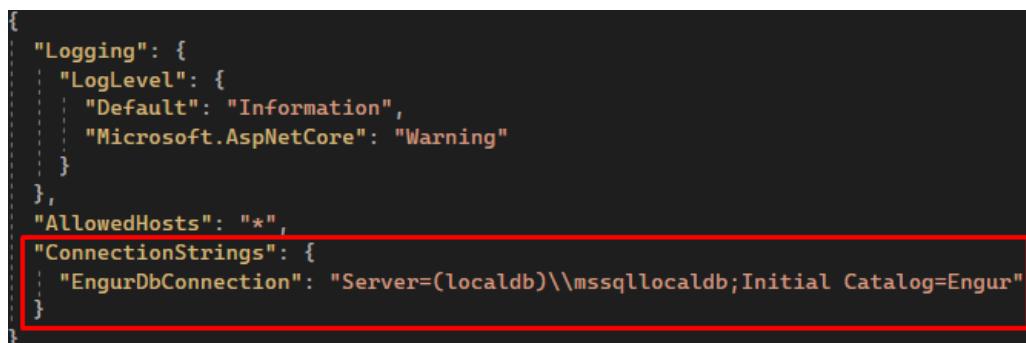


```

{
    public EngurDbContext(DbContextOptions<EngurDbContext> options) :
base(options) { }
    public DbSet<Word> Words => Set<Word>();
    public DbSet<Topic> Topics => Set<Topic>();
}

```

Щоб підключатися до бази даних, потрібно встановити параметри підключення. Для цього змінимо файл `appsettings.json`. За замовчуванням він містить лише налаштування логуювання. Змінимо його, додавши визначення рядка підключення (рис. 3.7).



```

{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "EngurDbConnection": "Server=(localdb)\\mssqllocaldb;Initial Catalog=Engur"
  }
}

```

Рисунок 3.7 – Файл `appsettings.json`

У даному випадку ми будемо використовувати LocalDB, який представляє полегшену версію SQL Server Express, призначену спеціально для розробки програм. Про це говорить параметр `Server=(localdb)\\mssqllocaldb`. Ну, а сама база даних буде називатися `Engur`.

Фінальним кроком налаштування проекту є реєстрація контексту проекту `DbContext` у файлі `Program.cs`. Параметр рядку підключення до бази даних при цьому підставляється той, який вказаний у файлі `appsettings.json`:

```
builder.Services.AddDbContext<EngurDbContext>(opt => opt.UseSqlServer(
builder.Configuration.GetConnectionString("EngurDbConnection")));
```

Створену базу даних можна переглянути через MS SQL Server Management Studio (рис. 3.8).

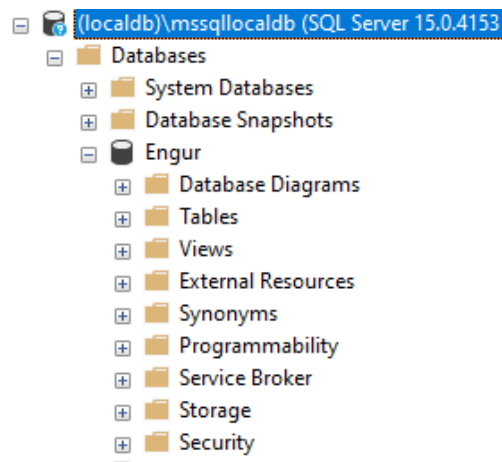


Рисунок 3.8 – База даних Engur

У процесі розроблення цілком імовірна ситуація, що клас моделі Entity Framework змінився, і доводиться видалити базу даних, щоб зберігалася відповідність моделі. Але при видаленні бази даних видаляються всі дані з неї.

Щоб зберегти дані при зміні моделі, в Entity Framework Core існує функція міграції. Вона дозволяє послідовно застосовувати зміни до бази даних, щоб синхронізувати її з моделлю даних.

У міграції існують операції, які дозволяють видалити, додавати стовпці та таблиці, зовнішні ключі, змінювати налаштування стовпців, додавати, видалити та змінювати дані тощо. При створенні міграції автоматично створюється клас, де

виконуються операції, які необхідні застосування міграції Up() та її повернення метод Down().

На даному етапі налаштування бази даних закінчено, а програма має безумовний доступ до бази. У випадку зміни вимог та необхідності додавання нових моделей, буде застосовано механізм міграцій.

Згідно обраної архітектури, систему було поділено на 3 логічних рівні: Data Access Layer, Business Logic Layer та Presentation Layer. Традиційно розробка починається з базового рівня Data Access Layer (рис. 3.9).

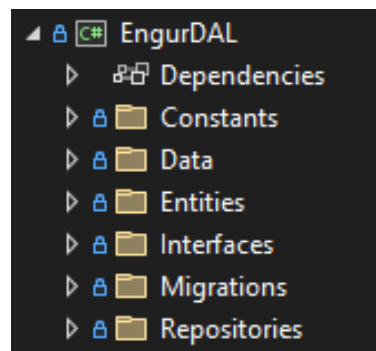


Рисунок 3.9 – Структура рівня DAL

Папка Constants призначена для зберігання даних, що будуть залишатися сталими впродовж роботи програми, до яких можна в будь-який момент отримати доступ з будь-якого класу рівня.

Папка Data призначена для зберігання контексту даних DbContext, про який детально йшлося в попередньому розділі та класу UnitOfWork.

Папка Entities призначена для зберігання критичних для даного рівня моделей даних entities, розробка яких детально описана в попередньому розділі.

Папка Interfaces призначена для зберігання контрактів класів.

Папка Migrations зберігає міграції бази даних, пов'язані зі зміною моделей даних та внесення змін у вже існуючу базу даних.

Папка `Repositories` зберігає класи репозиторіїв даних, які детально описані дані.

Для інкапсуляції логіки роботи з БД використано патерн `Repository`. Додатковий патерн `Unit of Work` допомагає спростити роботу з різними репозиторіями так, що вони будуть використовувати один і той же `DbContext`.

Також використання патерну `Repository` і `Unit of Work` дозволяє створити правильну структуру для розгортання додатків і впровадження DI практик, які в тому числі допомагають при тестуванні проєкту [30] (рис. 3.10).

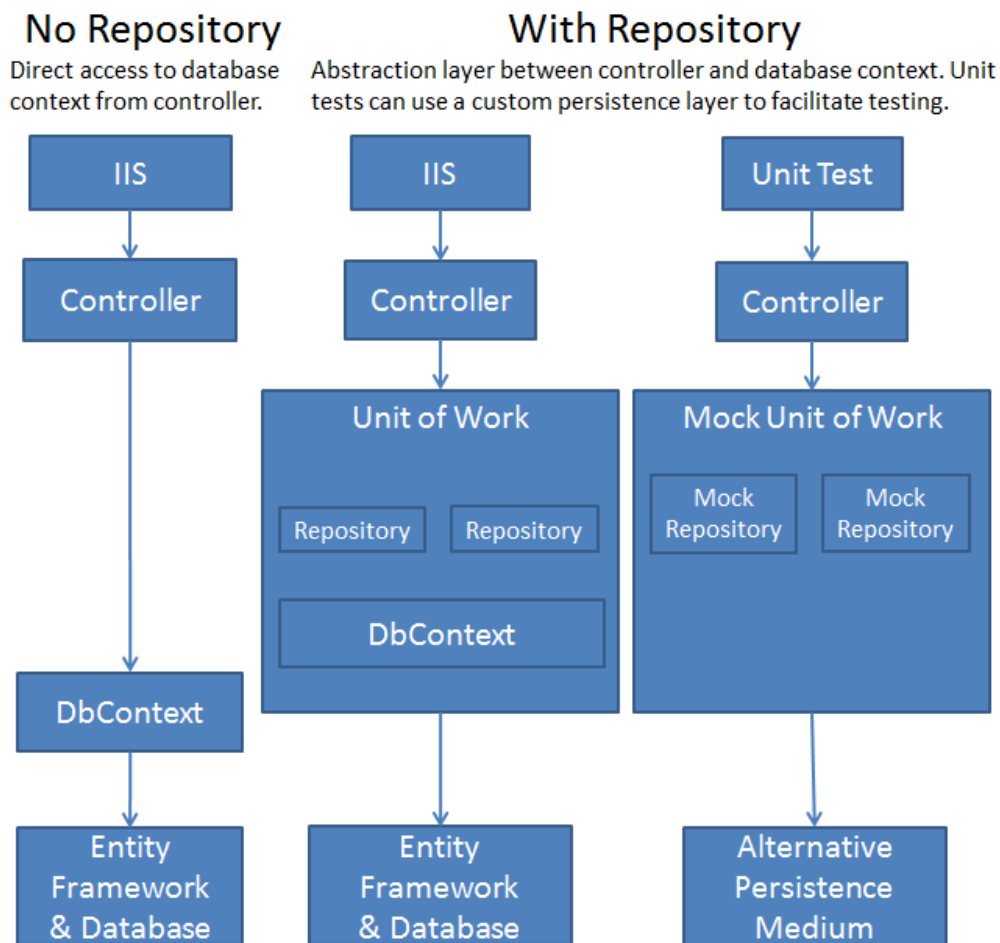


Рисунок 3.10 – Використання патерну `Repository` та `Unit of Work`

На прикладі сутності слова Word буде показано принципи, за якими в системі реалізуються дані патерни. Розробку слід почати з інтерфейсу:

```
public interface IWordRepository : IDisposable
{
    Task<Word?> GetByIdAsync(int id);
    Task<IEnumerable<Word>> GetAllAsync();
    Task CreateAsync(Word entity);
    Task DeleteByIdAsync(int id);
    void Update(Word entity);
}
```

Описаний інтерфейс містить набір CRUD методів, що знадобляться для роботи з базою даних, таких як створення, видалення, оновлення, отримання. Наступним логічним кроком буде реалізація даного інтерфейсу в класі Репозиторію:

```
public class WordRepository : IWordRepository
{
    private readonly EngurDbContext _context;
    public WordRepository(EngurDbContext context)
    {
        _context = context;
    }
    public async Task CreateAsync(Word word)
    {
        if (word == null)
```

```
{
    throw new ArgumentNullException(nameof(word));
}
await _context.Words.AddAsync(word);
}
public async Task DeleteByIdAsync(int id)
{
    var word = await _context.Words.FirstOrDefault(w => w.Id == id);
    if (word != null)
    {
        _context.Words.Remove(word);
    }
}
public async Task<IEnumerable<Word>> GetAllAsync()
{
    return await _context.Words.ToListAsync();
}
public async Task<Word?> GetByIdAsync(int id)
{
    return await _context.Words.FirstOrDefault(t => t.Id == id);
}
public void Update(Word word)
{
    _context.Words.Update(word);
}
public void Dispose()
{
```

```

        _context.Dispose();
    }
}

```

Описаний клас через конструктор отримує контекст даних, містить реалізації асинхронних CRUD методів, описаних в інтерфейсі, а також сервісний метод Dispose, що потрібен для вивільнення некерованих ресурсів збірником сміття C# [29].

На цьому моменті розроблений клас вже можна повноцінно використовувати в сервісах та контролерах, але враховуючи майбутнє масштабування системи з додаванням нових моделей даних, доцільно буде об'єднати всі репозиторії в одному місці. Саме для цього впроваджується патерн Unit Of Work:

```

public interface IUnitOfWork : IDisposable
{
    IWordRepository WordRepository { get; }
    ITopicRepository TopicRepository { get; }
    Task SaveChangesAsync();
}

```

В інтерфейсі у вигляді властивостей перераховуються всі наявні в системі репозиторії, а також сигнатура асинхронного методу SaveChangesAsync(), що буде використовуватися для виконання транзакції. Фінальним кроком реалізації буде клас Unit Of Work:

```

public class UnitOfWork : IUnitOfWork, IDisposable
{

```

```
private EngurDbContext _context;
private IWordRepository _wordRepository;
private ITopicRepository _topicRepository;
public UnitOfWork(EngurDbContext context)
{
    _context = context;
}
public IWordRepository WordRepository
{
    get
    {
        return _wordRepository ??= new WordRepository(_context);
    }
}
public ITopicRepository TopicRepository
{
    get
    {
        return _topicRepository ??= new TopicRepository(_context);
    }
}
public async Task SaveChangesAsync()
{
    await _context.SaveChangesAsync();
}
public void Dispose()
{
```



```

        _context.Dispose();
    }
}

```

Даний клас містить приватні поля, що зберігають посилання на єдиний контекст бази даних та репозиторії моделей даних, доступ до яких здійснюється через публічні властивості. Також тут реалізовано сервісні методи `SaveChangesAsync()` та `Dispose()`, призначення яких було описано раніше.

Рівень `Business Logic Layer` є проміжним сервісним рівнем між даними та їхнім представленням. На даному рівні відбувається конвертація та передача даних з нижнього рівня на верхній та, за потреби, виконуються операції над даними. Рівень має просту та самоописувальну структуру (рис. 3.11).

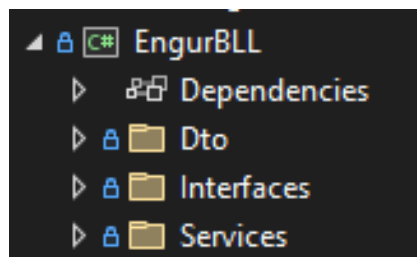


Рисунок 3.11 – Структура рівня BLL

Папка `Dto` призначена для зберігання `Data Transfer Object` моделей передачі даних між рівнями.

Папка `Interfaces` призначена для зберігання контрактів сервісів рівня.

Папка `Services` призначена для зберігання класів сервісів, у яких і виконуються проміжні операції над даними.

Модель зберігання даних часто не відповідає потребам моделі відображення даних, і навпаки. Рішенням тут є патерн `Data Transfer Object`, який може зберігати всю необхідну для виклику інформацію:

```
public class WordDto
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Translation { get; set; }
    public int TopicId { get; set; }
}
```

На прикладі сервісу роботи з сутністю Topic буде показано принципи, за якими в системі реалізуються сервіси. Розробку слід почати з інтерфейсу:

```
public interface ITopicService
{
    Task<TopicDto?> GetByIdWithWordsAsync(int id);
    Task<IEnumerable<TopicDto>> GetAllAsync();
    Task CreateAsync(TopicDto word);
    Task DeleteByIdAsync(int id);
    Task UpdateAsync(TopicDto word);
}
```

Даний інтерфейс фактично повторює CRUD методи, описані в репозиторії, а також на їхній базі реалізує нові комплексні методи, як наприклад `GetByIdWithWordsAsync`, що за одним ідентифікатором буде повертати не тільки сам модуль, але й пов'язані з ним слова. Принцип реалізації у сервісу подібний до репозиторію, але додатково виконується конвертація даних з рівня зберігання даних до рівня представлення даних:

```
public class TopicService : ITopicService
{
    private readonly IUnitOfWork _unitOfWork;

    public TopicService(IUnitOfWork uow)
    {
        _unitOfWork = uow;
    }

    public async Task CreateAsync(TopicDto topic)
    {
        await
        _unitOfWork.TopicRepository.CreateAsync(TopicDto.MapFromDto(topic));
        await _unitOfWork.SaveChangesAsync();
    }

    public async Task DeleteByIdAsync(int id)
    {
        await _unitOfWork.TopicRepository.DeleteByIdAsync(id);

        await _unitOfWork.SaveChangesAsync();
    }

    public async Task<IEnumerable<TopicDto>> GetAllAsync()
    {
        var topics = await _unitOfWork.TopicRepository.GetAllAsync();
        return TopicDto.MapToDtos(topics);
    }

    public async Task<TopicDto?> GetByIdWithWordsAsync(int id)
    {
```

```

var topic = await
_unitOfWork.TopicRepository.GetByIdWithWordsAsync(id);
if(topic == null)
{
    throw new ArgumentException(nameof(topic));
}
return TopicDto.MapToDto(topic);
}
public async Task UpdateAsync(TopicDto topic)
{
    _unitOfWork.TopicRepository.Update(TopicDto.MapFromDto(topic));
    await _unitOfWork.SaveChangesAsync();
}
}

```

Рівень представлення являє собою API проект, основою якого є контролери (рис. 3.12):

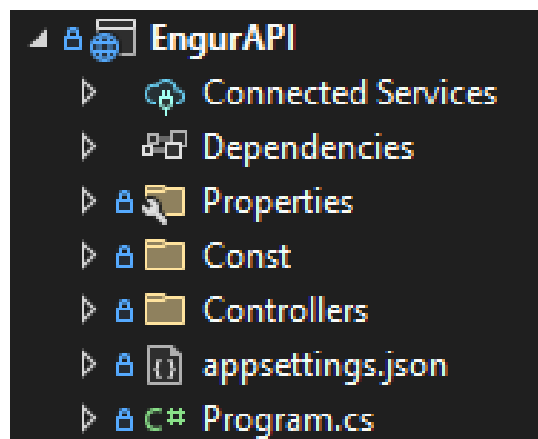


Рисунок 3.12 – Структура рівня PL

Папка `Properties` призначена для зберігання файлів конфігурацій, як наприклад `launchSettings.json`, в якому містяться властивості запуску додатку (`authentication`, `launchUrl`, `sslPort`).

Папка `Const` призначена для зберігання даних, що не будуть змінюватися впродовж всієї роботи програми, до яких можна в будь-який момент отримати доступ з будь-якого класу рівня.

Папка `Controllers` призначена для зберігання контролерів – об'єктів, що обробляють HTTP-запити.

Також важливо відмітити клас `Program.cs`. Раніше він вже згадувався, коли відбувалася реєстрація контексту бази даних. Загалом цей клас важливий тим, що в ньому відбувається реєстрація та побудова всіх сервісів, що задіяні в системі, а також сам метод запуску додатку.

На прикладі контролеру для сутності `Word` буде показано, як відбувається обробка HTTP запитів:

```
[Route("api/[controller]")]
[ApiController]
public class TopicController : ControllerBase
{
    private readonly ITopicService _topicService;
    public TopicController(ITopicService topicService)
    {
        _topicService = topicService;
    }
    [HttpGet]
    public async Task<IEnumerable<TopicDto>> GetAllTopics()
    {
```

```
        return await _topicService.GetAllAsync();
    }
    [HttpGet("{id}")]
    public async Task<IResult> GetTopicByIdWithWordsAsync(int id)
    {
        var topic = await _topicService.GetByIdWithWordsAsync(id);
        return topic == null ? Results.NotFound() : Results.Ok(topic);
    }
    [HttpPost]
    public async Task<IResult> CreateTopic(TopicDto topicDto)
    {
        await _topicService.CreateAsync(topicDto);
        return Results.Ok();
    }
    [HttpPut]
    public async Task<IResult> UpdateTopic(TopicDto topicDto)
    {
        await _topicService.UpdateAsync(topicDto);
        return Results.Ok();
    }
    [HttpDelete("{id}")]
    public async Task<IResult> DeleteTopic(int id)
    {
        await _topicService.DeleteByIdAsync(id);
        return Results.Ok();
    }
}
```

Клас наслідується від базового API контролера ControllerBase та розширяє його функціональність методами, необхідними для обробки запитів сутностей Word. Першим рядком вказується загальний маршрут, по якому буде доступний даний контролер. Кожен метод повинен мати унікальний маршрут та може відповідати на певний тип запитів: Get, Put, Post, Delete. Описуючи додавання абсолютно нової сутності, контролер буде приймати від клієнта модель формату Data Transfer Object та передавати її в сервіс рівня BLL, який, в свою чергу, виконає конвертацію об'єкта у модель зберігання даних та передасть на рівень DAL для запису в базу даних. З іншого боку, описуючи метод отримання сутностей з бази, контролер викличе відповідний метод сервісу рівня BLL, який, в свою чергу, викличе метод рівня DAL, що поверне колекцію сутностей, тип даних якої конвертується у сервісі рівня BLL та повернеться у контролер для виведення клієнту. Крім фактичного повернення даних на запит клієнта, можливо також повертати лише статус коди, наприклад «200» – операція успішна, або «500» – помилка сервера. Тип повернених даних при проєктуванні обирається індивідуально, в залежності від конкретних вимог та очікувань методу клієнта.

Отже, підсистема поділена на три логічних рівні, представлених окремими проєктами: рівні Data Access Layer (DAL) та Business Logic Layer (BLL) являють собою DLL-бібліотеки, а рівень Presentation Layer (PL або API) являє собою проєкт Web API, оскільки він є основним виконуваним проєктом і інтерфейсом для взаємодії із системою (рис. 3.13).

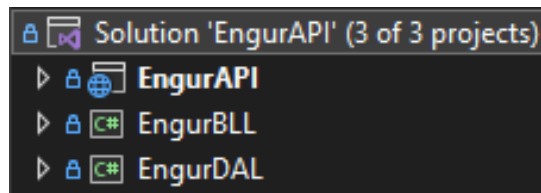


Рисунок 3.13 – Структура проєктів системи

На рівні доступу до даних даними може бути будь-яка база даних (Oracle, SQL Server, Access, PostgreSQL, MySQL тощо) або будь-який файл. Цей рівень є єдиним зв'язком між вашою програмою та цими даними.

На рівні бізнес логіки відбудвається трансформація даних із одного формату в інший та часто певні операції обчислення.

Рівень презентації являє собою інтерфейс, призначений для комунікації клієнта з ядром системи.

Рівні фактично поки що ніяк не взаємодіють, оскільки не знають про існування один одного. Для того, щоб додати проєкт в область видимості іншого, необхідно в контекстному меню проєкту перейти в Add, обрати Project Reference (рис. 3.14).

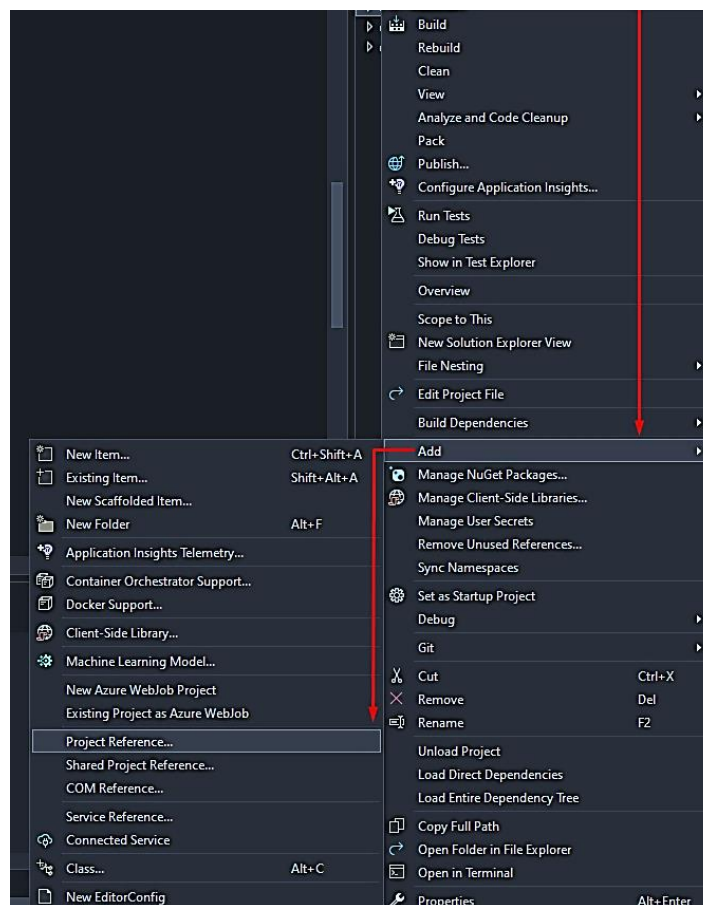


Рисунок 3.14 – Контекстне меню проєкту

Принцип побудови даної архітектури такий, що нижній рівень не здогадується про існування вищого та не залежить від його реалізації, що дозволяє перевикористовувати код, замість того, щоб писати його знову. Слідуючи даному принципу, рівню DAL не потрібно додавати посилань на інші проєкти. Для рівня BLL достатньо буде посилання на проєкт нижнього, по відношенню до нього, рівня, тобто на DAL. Нарешті, для проєкту PL необхідно додати посилання на всі рівні, оскільки в даному проєкті виконується реєстрація сервісів всіх рівнів, таких як контекст даних для рівня DAL, чи сервісів рівня BLL (рис. 3.15).

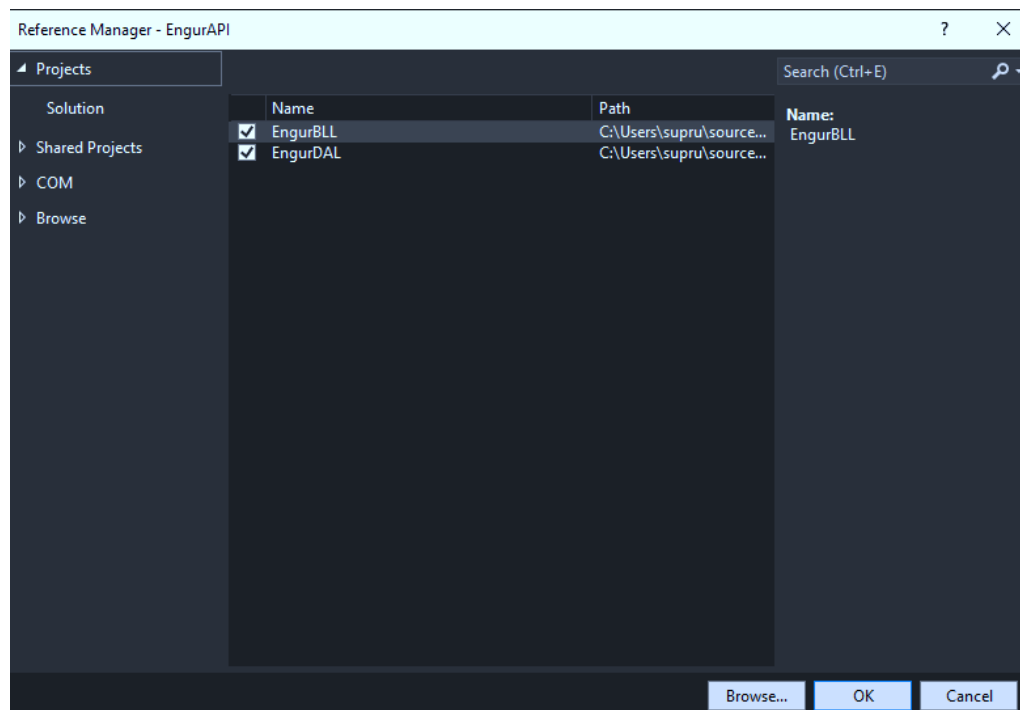


Рисунок 3.15 – Додавання посилань на інші проєкти

3.3 Тестування програмного продукту

Тестування – це комплексна перевірка функцій системи різними методами і способами на правильність роботи. Аналіз необхідний як новій, так і працюючій

системі, для отримання гарантії працездатності ресурсу і підвищення його ефективності.

Для тестування системи було застосовано наступні види тестування: функціональне та нефункціональне в ручному та автоматичному режимах.

Функціональне тестування – це тестування, головною метою якого є перевірка реалізації функціональних критеріїв системи, тобто її здатність в заданих умовах вирішувати відведені їй задачі. Зазвичай, до вимог даного тестування включають наступні критерії:

- захищеність;
- відповідність вимогам;
- здатність до взаємодії;
- точність.

До переваг функціонального тестування можна віднести значне покриття різноманітними функціональними тестами та імітацію тестованої системи з боку користувача. До недоліків відноситься надлишковість тестування та ймовірність допустити логічні помилки в коді.

Нефункціональне тестування – це, в першу чергу, перевірка того, як саме працює система. Перевіряється відповідність таким вимогам:

- зручність (оцінка зручності для користувачів);
- продуктивність (швидкодія системи при різних навантаженнях);
- безпека (захист даних користувача);
- надійність (поведінка системи під час непередбачуваних обставин, здатність обробляти нестандартні дії користувача).

У тестуванні системи брало участь 3 незалежних користувачів, які допомагали з виявленням проблем для налаштування коректної роботи функцій програми.

Систему було протестовано за такими критеріями:

- функціональність (перевірено працездатність API контролерів);
- коректне відображення запитаного контенту;
- коректний запис нових даних в систему;
- відповідність роботи підсистеми оцінювання якості навчання;
- кросбраузерність та масштабованість (перевірено працездатність у найпопулярніших веб-браузерах, таких як Google Chrome, Firefox та Opera, а також при різних налаштуваннях роздільної здатності [28]);
- зручність використання (перевірено, чи зручно звичайним користувачам користуватись веб-системою.

За допомогою автоматичних тестувань було виявлено та виправлено більшість багів системи, решту було знайдено при ручному тестуванні.

Висновки до розділу 3

У даному розділі комплексної магістерської кваліфікаційної роботи описано програмну реалізацію системи для дистанційного вивчення слів англійської мови в межах підсистеми оцінювання якості навчання. Насамперед, було обґрунтовано технології та мову програмування для розроблення дистанційної системи оцінювання якості навчання. Для клієнтської частини було вирішено розробляти Angular-застосунок із візуальним інтерфейсом користувача. Для серверної частини ключовою технологією виступатиме .NET та стек його основних фреймворків. Як сервер бази даних обрано Microsoft SQL Server.

У другій частині розділу описано проєктування програмних засобів підсистеми: обґрунтовано вибір архітектури програмного комплексу, спроектовано базу даних, розроблено основну частину додатку та виконано злагодження всього комплексу.

У результаті роботи було розроблено ASP.NET Web API застосунок та базу даних. Основний застосунок базується на трирівневій архітектурі та надає інтерфейс керування програмою клієнтській частині, а також виконує основні маніпуляції з даними на їхньому шляху від клієнта до бази даних і навпаки. Розроблено нормалізовану базу даних, що керується безпосередньо кодом додатку, але також має можливість ручного керування за допомогою Microsoft SQL Management Studio.

4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Аналіз результатів, отриманих на базі запропонованого ПЗ

Розглянемо послідовність роботи підсистеми із самого початку. Нехай користувач обрав завдання для виконання (рис. 4.1). Із цього моменту починається робота підсистеми оцінювання якості навчання.

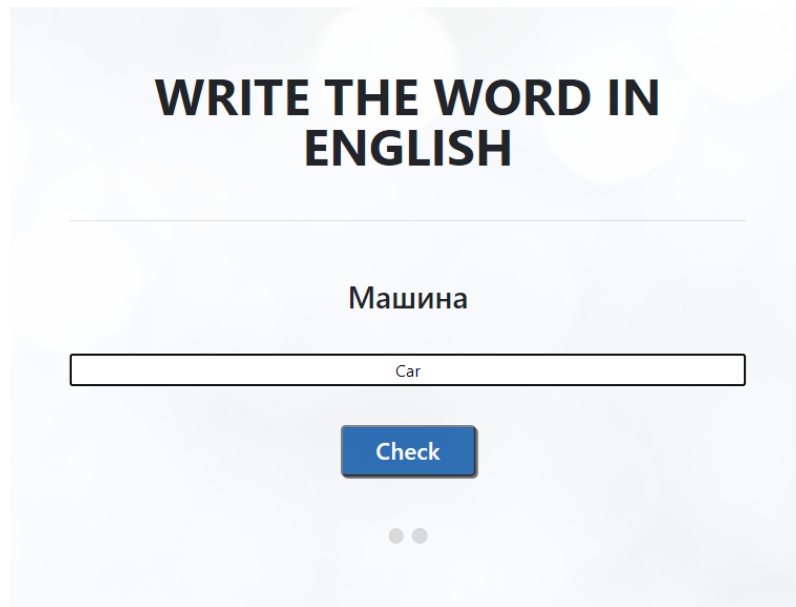


Рисунок 4.1 – Тестове завдання для перевірки знань

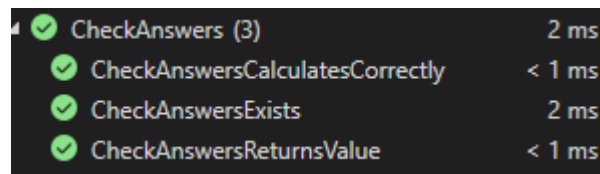
Як тільки користувач завершить проходження даного тестування, дані про його проходження передадуться від клієнта на сервер, де почнеться їх оброблення за запрограмованими моделями, що було описно в другому розділі даної роботи. Перша перевірка – кількість правильних відповідей (4.1).

$$R = \frac{\sum_{i=1}^k R_i}{n} . \quad (4.1)$$

Розглянемо програмну ілюстрацію такого процесу:

```
double CheckAnswers(double questions, List<int> answers)
{
    double res = 0;
    for(int i = 0; i < questions; i++)
    {
        if (answers[i] == 1)
        {
            res += 1 / questions;
        }
    }
    return res;
}
```

Для того, щоб перевірити коректність роботи моделі та відповідної її програмної реалізації, було запропоновано кілька юніт-тестів (рис. 4.2).



✓ CheckAnswers (3)	2 ms
✓ CheckAnswersCalculatesCorrectly	< 1 ms
✓ CheckAnswersExists	2 ms
✓ CheckAnswersReturnsValue	< 1 ms

Рисунок 4.2 – Результат виконання юніт тестів

Клієнт системи побачить свій результат одразу після виконання тестового завдання (рис. 4.3).

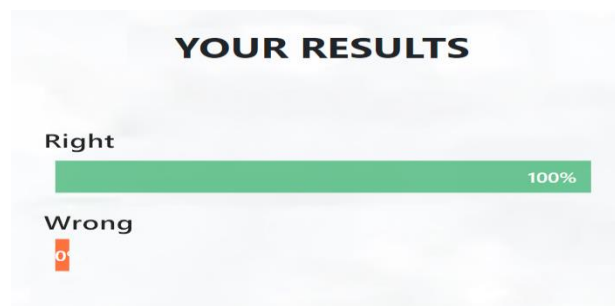


Рисунок 4.3 – Результат виконання тестового завдання

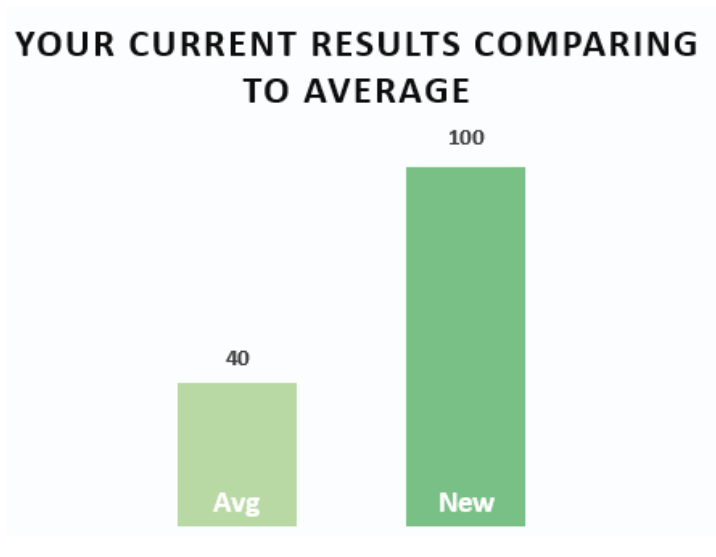


Рисунок 4.4 – Результат виконання перевірки виконання завдання

Отриманий результат перевірки одночасно додається до бази даних і виводиться клієнту. Хоча цей результат є проміжним, для підсистеми він має велике значення, оскільки саме на результатах перевірки тестового завдання базуються інші моделі (4.2):

- модель загального результату;
- модель середньої оцінки;
- модель уточненої середньої оцінки .

$$y = \sum_{i=1}^n w_i R_i ,$$

$$A = \frac{y}{k_n},$$

$$A' = A \times r + a_1 \frac{k_n}{n} + a_2 \frac{k_c}{n} + a_3 \frac{k_b}{n} . \quad (4.2)$$

Розглянемо код, який реалізує роботу даних моделей у підсистемі оцінювання якості навчання:

```

double EvaluateAsnwrs(List<TestResult> userResults, User user, TestProfile
testProfile)
{
    if (userResults.Count == 0)
    {
        return 0;
    }
    double res = 0;
    for (int i = 0; i < userResults; i++)
    {
        res += userResults.Score * userResults.TestType.Coeffitient;
    }
    res = res * user.Coeffitient / userResults.Count +
        +testProfile.TriesCountCoeffitient * user.TriesCount / userResults.Count +
        +testProfile.PeepCountCoeffitient * user.PeepCount / userResults.Count +
        +testProfile.OffTimeCoeffitient * user.OffTimeCount / userResults.Count;
    return res;
}

```

Для того, щоб модель функціонувала коректно, додатково було написано кілька юніт-тестів (рис. 4.4).

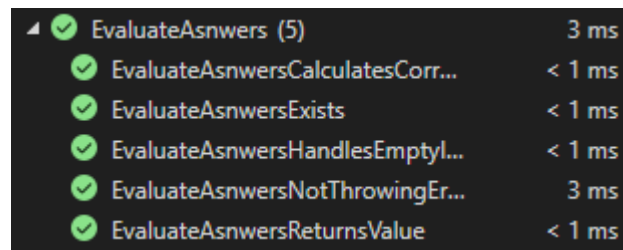


Рисунок 4.4 – Результат виконання юніт тестів

На рис. 4.5 наведено результат, який формується після відпрацювання всіх моделей.

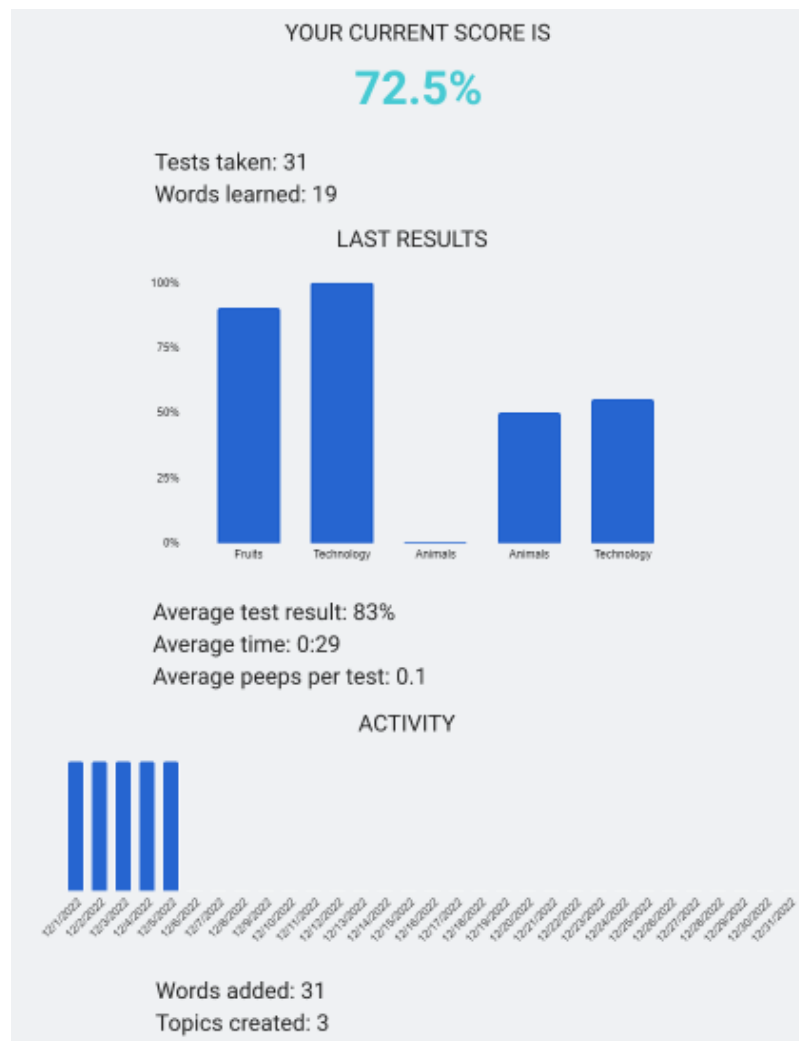
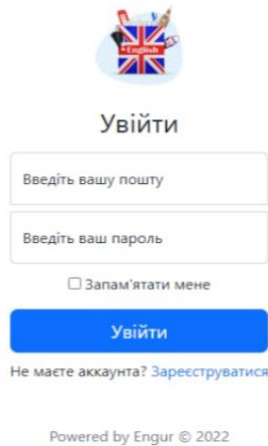


Рисунок 4.5 – Результати навчання у статистиці користувача

4.2 Керівництво оператора

Web-система для вивчення слів англійської мови надає можливість записувати та іноземні слова. Розроблена підсистема розширяє можливості системи таким чином, що враховує додаткові параметри при перевірці знань, тому користувач має можливість отримувати розширену характеристику рівня своїх знань.

На початку роботи запускається стартова сторінка, на якій відображається вікно входу в систему (рис. 4.6). Якщо користувач не має акаунта, його можна зареєструвати, натиснувши на відповідну кнопку.



English

Увійти

Введіть вашу пошту

Введіть ваш пароль

Запам'ятати мене

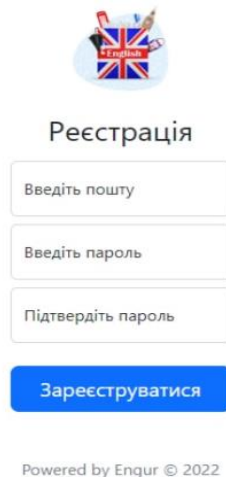
Увійти

Не маєте акаунта? [Зареєструватися](#)

Powered by Engur © 2022

Рисунок 4.6 – Форма введення даних для входу

Натиснувши кнопку «Зареєструватися», користувача перенаправляє на сторінку реєстрації (рис. 4.7), де потрібно ввести пошту та пароль. У системі діє перевірка правильності введених даних, а також перевіряється, чи немає вже в системі користувача з такою ж поштою.



English

Реєстрація

Введіть пошту

Введіть пароль

Підтвердіть пароль

Зареєструватися

Powered by Engur © 2022

Рисунок 4.6 – Форма введення даних для реєстрації

Після того, як користувач увійде в систему, його перенаправить на головну сторінку, звідки можна здійснювати навігацію системою (рис. 4.7). Інтерфейс для зручності виконаний у мінімалістичному стилі, а всі потрібні кнопки знаходяться на видному місці. У правому верхньому кутку знаходиться пошук по сайту, який допоможе знайти додані раніше модулі слів та самі слова. Також поруч знаходиться кнопка випадаючого меню користувача.

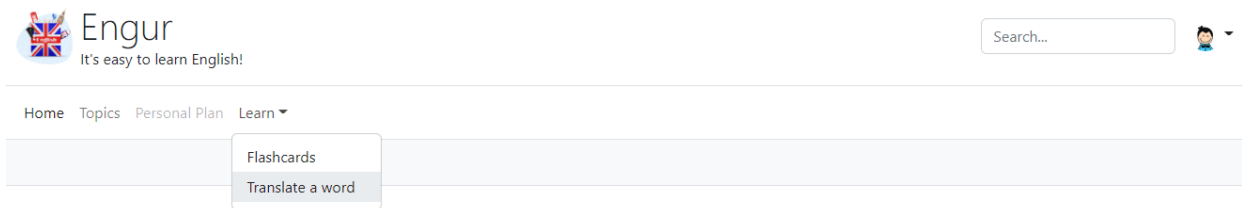


Рисунок 4.7 – Меню головної сторінки системи

Обираємо метод навчання «Translate the word». Система виконає перенаправлення, після чого обираємо тему і проходимо тест.

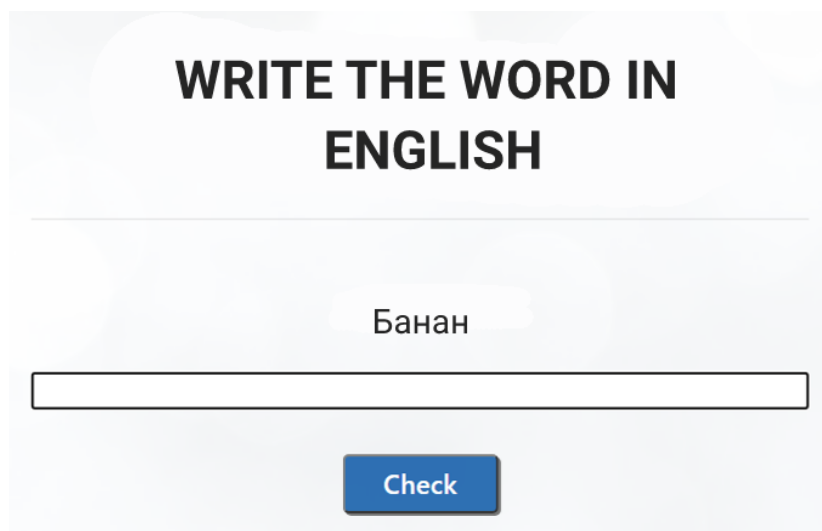


Рисунок 4.8 – Тест на правильний переклад слова

Після того, як тест буде пройдено і відображено результати, переходимо до сторінки профілю, щоб перевірити роботу розробленої підсистеми.

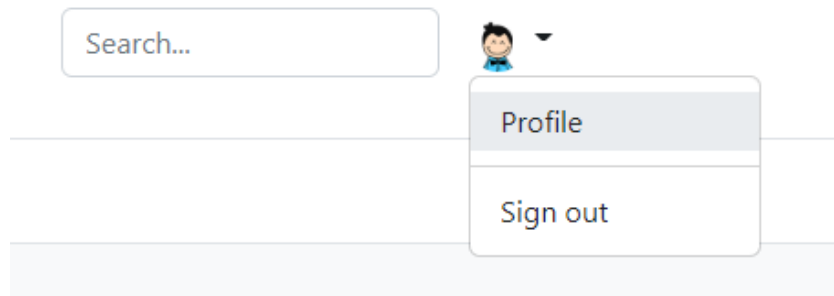


Рисунок 4.8 – Контекстне меню користувача системи

На сторінці профілю відобразяться зміни, що були внесені підсистемою в результаті проходження користувачем тесту.

Основна змінна, що розраховується за математичними моделями, побудованими в роботі, знаходиться одразу під заголовком «Your current score».

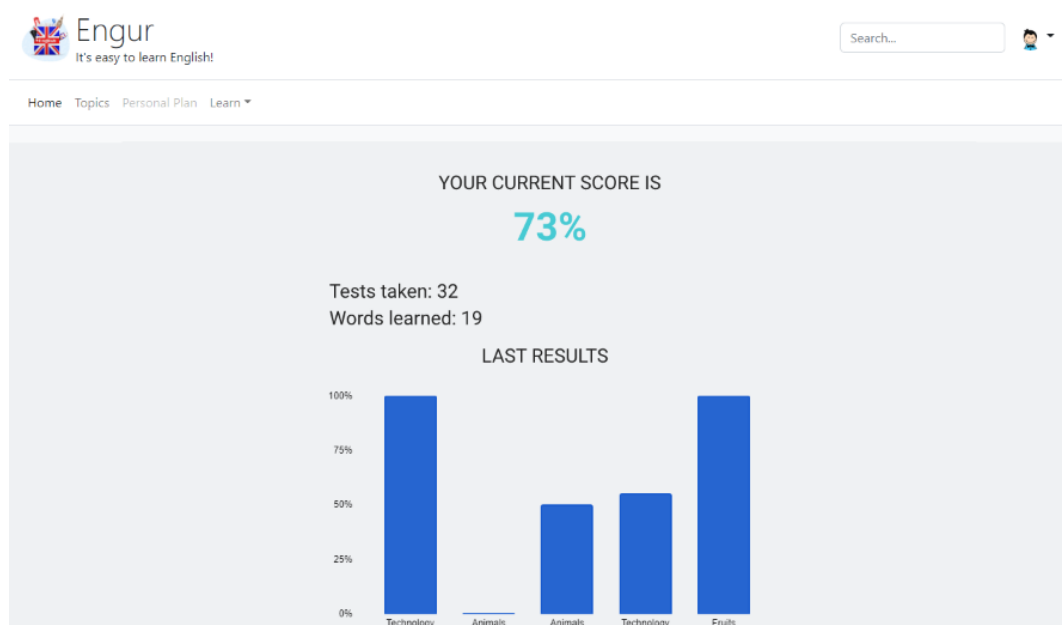


Рисунок 4.9 – Сторінка профілю користувача

4.3 Порівняння результатів роботи ПЗ із результатами провідних аналогів

Сьогодні на ринку є безліч систем для вивчення мов та слів, кожна з яких за замовчуванням має базовий набір функцій. Отже, вибір залишається робити на основі зручності інтерфейсу або функціональності чи наявності обмежень платної підписки.

Для порівняння обрано 3 провідних системи галузі: Duolingo, Lingualeo та Memrise.

Система Duolingo (4.10) спеціалізується на вивченні мов та пропонує великий набір стандартних модулів для вивчення, що включають слова, граматику та лексику.

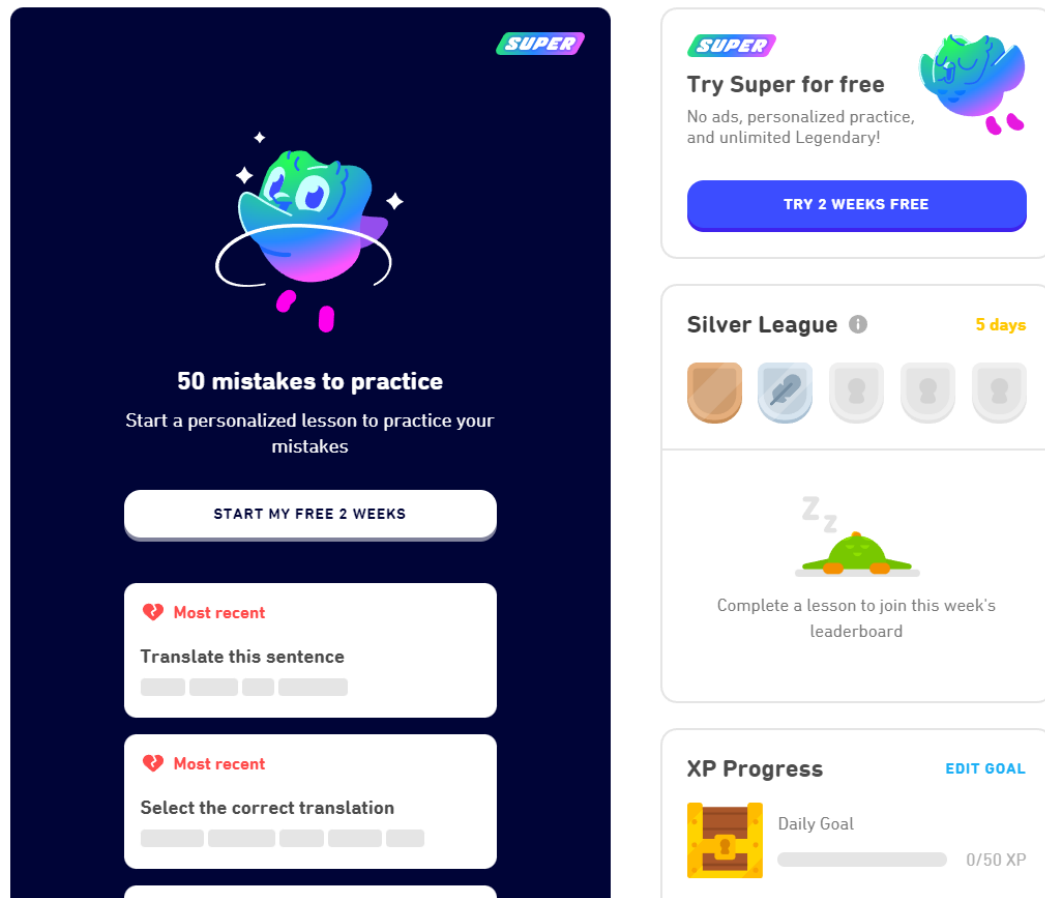


Рисунок 4.10 – Сторінка статистики Duolingo

Для надання статистики наявний лічильник неперервності занять та загальної кількості пройденого матеріалу. За додаткову плату пропонується підключити можливість практикувати слова та модулі, в яких було допущено помилки під час навчання.

Система Lingualeo також спеціалізується на вивченні мов та слів, але, на відміну від попередньої, володіє більшим різноманіттям навчальних вправ. У системі знаходяться перевірені експертами та готові до вивчення модулі слів та речень, а також є можливість вести власний словник та практикувати лише слова з нього. У системі присутня розгорнута статистика, яка рахує метрики користувача при навчанні. Ліва частина матеріалу та навчальних вправ залишається закритою до моменту придбання платної підписки.

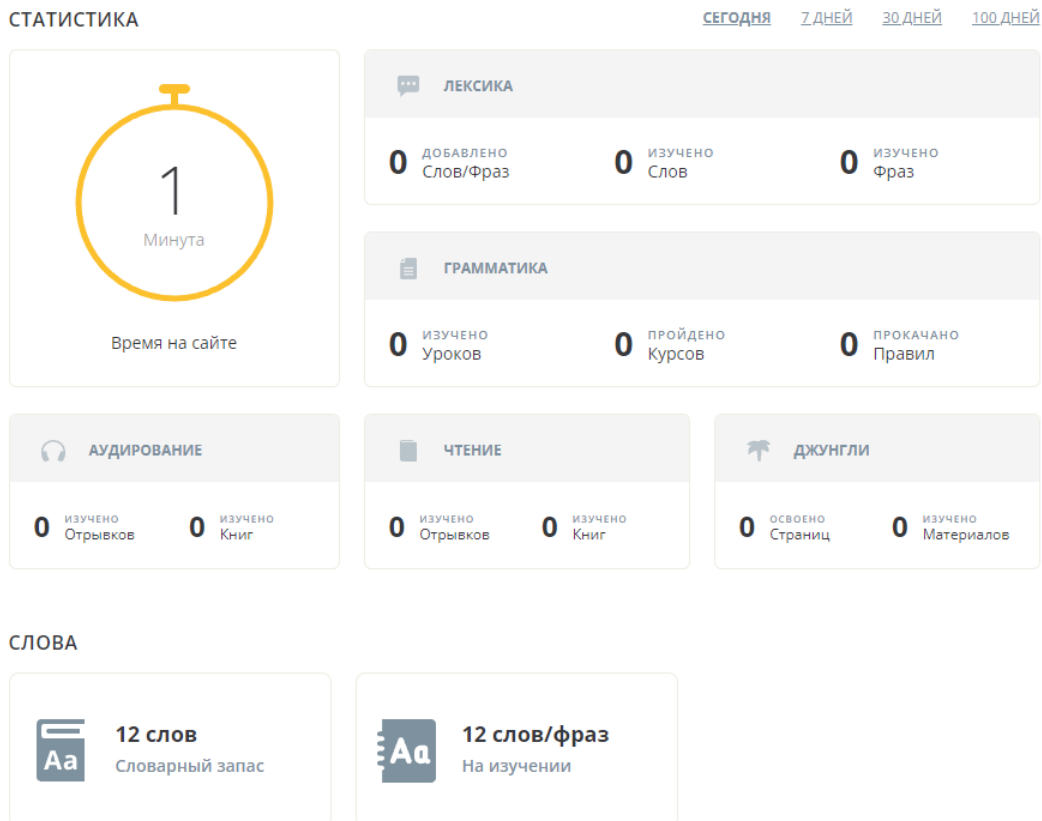


Рисунок 4.10 – Сторінка статистики Lingualeo

Система Memrise спеціалізується на вивченні слів та технологіях їхнього запам'ятовування. Для користувачів доступно декілька видів навчальних вправ на вибір. Початково користувачу пропонується створити навчальний модуль, заповнити його колекцією слів, і лише потім вивчати їх із допомогою різних режимів. В останній версії системи розробники додали стандартні модулі слів для вивчення. У вікні статистики системи присутні календар занять, метрики часу проведеного у системі та перелік складних слів, у процесі вивчення яких була допущена помилка.

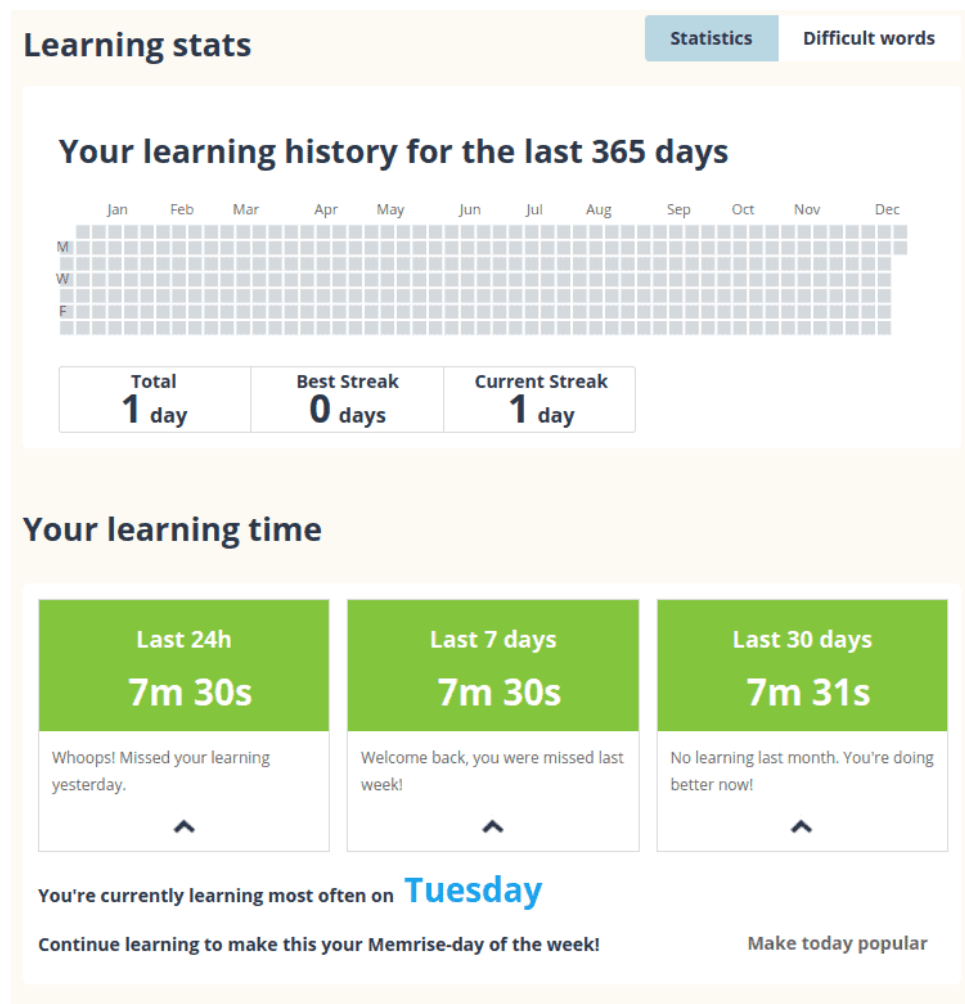


Рисунок 4.10 – Сторінка статистики Memrise

Система, що була розроблена в магістерській роботі, має всі базові функції, необхідні для вивчення слів іноземної мови. У користувача є можливість створювати свої модулі слів, ділитися ними або запозичувати колекції інших користувачів. Беручи це до уваги, можна сказати, що розроблена система підходить як для самостійного навчання, так і для роботи з викладачем. Для забезпечення процесу навчання у системі наявні: режим безпосередньо навчання – слова на картках та режим перевірки – переклад слів. Ключова відмінність роботи – підсистема оцінювання результатів навчання, що працює за власними математичними моделями і дає більш точне оцінювання знань респондента.

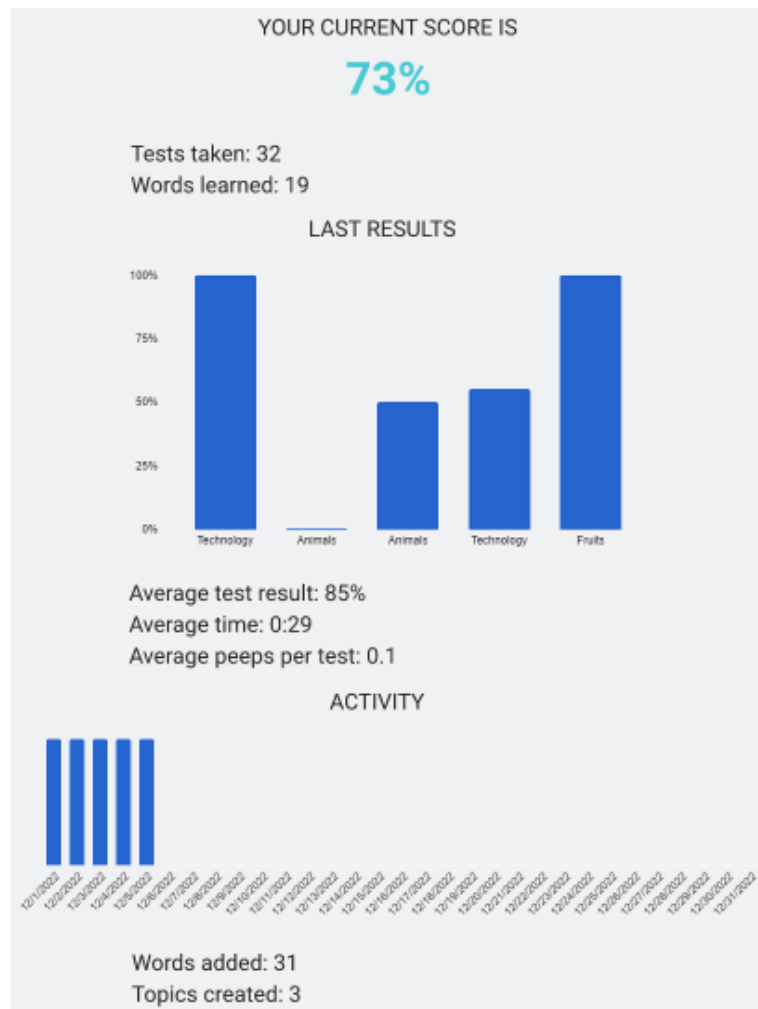


Рисунок 4.10 – Сторінка статистики розробленої системи

Висновки до розділу 4

У даному розділі магістерської роботи проведено тестування та аналіз роботи розробленого програмного забезпечення. Реалізовано автоматизоване тестування за допомогою юніт-тестів.

Детальну увагу приділено послідовності виконання модулів підсистеми оцінювання результатів навчання.

Продемонстровано роботу ключових функцій та дано детальні пояснення дій користувача та програмних модулів, які за них відповідають.

Порівняння роботи запропованої системи з аналогами дозволило виявити низку переваг першої, а саме: власна система є більш функціональною, наявним є попередній, поточний та підсумковий контроль знань, що дозволяє підвищити точність такого процесу; уможливорює ґрунтовне оцінювання результатів перевірки навчання за рахунок різноманітних форм тестування; не має рекламних акцій та додаткової оплати.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного продукту Web-система для дистанційного вивчення слів англійської мови. Частина 1. «Підсистема оцінювання якості навчання». Розробляється повноцінна система (бек + фронт) по типу Duolingo, Memrise, Lingualeo, яка дозволяє вивчати слова іноземної мови (в даному випадку англійської) з використанням методик навчання та тестування. Робота комплексна, в даній частині акцент зроблений на розробці та програмній реалізації методики оцінювання якості навчання.

Актуальність та новизна магістерської роботи полягає у розширенні можливостей системи дистанційного вивчення слів англійської мови, шляхом впровадження підсистеми оцінювання результатів навчання, що працює за власними математичними моделями, враховуючи індивідуальні фактори користувача.

Ціни аналогів: Lingualeo – 3840 грн/рік (доступ до більшої кількості навчальних матеріалів, більш цікавих та ефективніших способів вивчення, повторення помилок, персональний навчальний план).

Duolingo – 3840 грн/рік (немає реклами, необмежена кількість спроб, повторення помилок, навчальний план – узагальнений для всіх користувачів, ніяк не коригується до індивідуальних показників академічної успішності конкретного студента, на відміну від підсистеми, що розробляється)

uTalk – 14400 грн/рік (доступ до більшої кількості навчальних матеріалів, більш цікавих та ефективніших способів вивчення, але немає в наявності системи оцінювання якості навчання, чи персонального навчального плану).

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями (табл 5.1).

Таблиця 5.1 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	4	4	4
Наявність аналогів на ринку	3	3	3
Цінова політика	3	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	3	4	4
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	4	4	4
Сума	42	43	44
Середньоарифметична сума балів	$(42+43+44) / 3 = 43$		

За даними таблиці 4.1 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Середньоарифметична сума балів більше 40, отже рівень комерційного потенціалу нового програмного продукту, що розробляється, є високим, що досягається за рахунок того, що програмний продукт відрізняється від існуючих розширенням можливостей системи дистанційного вивчення слів англійської мови, шляхом впровадження підсистеми оцінювання якості навчання, що працює за власними математичними моделями та дозволить впровадити на своїй основі інші системи, як наприклад коригування процесу навчання.

5.2 Прогнозування витрат на виконання науково-дослідної роботи

Основна заробітна плата розробників розраховується за формулою (5.1).

$$Z_0 = \frac{M}{T_p} \times t, \quad (5.1)$$

де M – місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p – число робочих днів в місяці, 22 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.2.

Таблиця 5.2 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проєкту	25000	1136,36	30	34090,909
Програміст	22000	1000,00	30	30000,000
Всього				64090,91

В даному випадку розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Додаткову заробітну плату (5.2 і 5.3) прийнято розраховувати як 12 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_0 \cdot 12 \% / 100 \% , \quad (5.2)$$

$$Z_d = (64090,91 \cdot 12 \% / 100 \%) = 7690,91 \text{ (грн.)} \quad (5.3)$$

Згідно діючого законодавства нарахування на заробітну плату (5.4 і 5.5) складають 22 % від суми основної та додаткової заробітної плати.

$$H_z = (Z_o + Z_d) \cdot 22 \% / 100\% , \quad (5.4)$$

$$H_z = (64090,91 + 7690,91) \cdot 22 \% / 100 \% = 15792,00 \text{ (грн.)} \quad (5.5)$$

Оскільки при розробці системи витрачаються лише матеріали у вигляді пачки паперу (Maestro Standard+), то її вартість складе 195 грн.

Амортизація обладнання, що використовувалася для розробки розраховується за формулою (5.6):

$$A = \frac{Ц}{T_B} \cdot \frac{t_{\text{вик}}}{12} , \quad (5.6)$$

де $Ц$ – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років;

$t_{\text{вик}}$ – термін використання під час розробки, місяців.

Розрахуємо амортизаційні витрати на комп'ютер (5.7), балансова вартість якого становить 34154 грн., термін корисного використання згідно податкового законодавства – 2 роки, а термін фактичного використання – 1,36 міс.

$$A = \frac{34154}{2} \times \frac{1,36}{12} . \quad (5.7)$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до табл. 5.3.

Таблиця 5.3 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та периферія (Asus X556UQ, Lenovo M500 Wireless, Samsung SCX-4200)	34154	2	1,36	1940,568
Офісне обладнання (меблі)	20000	4	1,36	568,182
Приміщення	850000	20	1,36	4829,545
Всього				7338,30

Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн (Microsoft Windows 10 Pro - 7 899,00 грн., Microsoft SQL Server – 7 686,00 грн), то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $V_{нем.ак.} = 15585$ грн. Також потрібно включити у вартість нематеріальних ресурсів і вартість підписки на такий нематеріальний актив як Visual Studio Business IDE – 1 655,00 грн/міс. Актив використовувався 1,36 місяця, тому вартість підписки склала 2250,8 грн.

Витрати на електроенергію розраховуються за формулою (5.8 і 5.9):

$$V_e = B \times P \times \Phi \times K_n, \quad (5.8)$$

$$V_e = 0,9 \cdot 0,45 \cdot 8 \cdot 30 \cdot 6,2 = 602,64 \text{ (грн)}, \quad (5.9)$$

де B – вартість 1 кВт-години електроенергії для 1 класу підприємства, $B = 6,2$ грн./кВт;

P – встановлена потужність обладнання, кВт. $P = 0,45$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

K_n – коефіцієнт використання потужності, $K_n = 0,9$.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників (5.10 і 5.11):

$$I_g = (Z_o + Z_p) \cdot \frac{H_{iv}}{100\%}, \quad (5.10)$$

$$I_g = 64090,91 * 70\% / 100\% = 44863,64 \text{ (грн)}, \quad (5.11)$$

де H_{iv} – норма нарахування за статтею «Інші витрати».

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників (5.12 і 5.13):

$$H_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.12)$$

$$H_{нзв} = 64090,91 * 100\% / 100\% = 64091 \text{ (грн)}, \quad (5.13)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні витрати».

Сума всіх попередніх статей витрат (5.14) дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 64090,91 + 7690,91 + 15792,00 + 195 + 15585 + 2250,8 + 7338,30 + 602,64 + 44863,64 + 64091 = 222500,10 \text{ (грн)} \quad (5.14)$$

Загальні витрати на завершення науково-дослідної роботи та оформлення її результатів розраховуються ZB (5.15 і 5.16).

$$ZB = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (5.15)$$

де η – коефіцієнт, який характеризує етап виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка знаходиться на стадії дослідного зразка:

$$ZB = 222500,10 / 0,5 = 445000 \text{ (грн)}. \quad (5.16)$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

У випадку розробки та вдосконалення програмного засобу для використання масовим споживачем, майбутній економічний ефект (5.17) буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.17)$$

де $\pm\Delta C_o$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

C_o – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $C_o = C_b \pm \Delta C_o$;

C_b – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 2000 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 300 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 3000 шт., протягом другого року – на 2000 шт., протягом третього року на 1000 шт. В Україні на 25.08.2022 р. є понад 23,5 тис. закладів освіти [32]. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\begin{aligned} \Delta\Pi_I &= (0 \times 300 + (2000 + 300) \times 3000 \times 0,8333 \times 0,45) \times \\ &\quad \times (1 - 0,18) = 1844999,926 \text{ (грн)}, \end{aligned}$$

$$\begin{aligned}\Delta\Pi_2 &= (0 \times 300 + (2000 + 300) \times (3000+2000) \times 0,8333 \times 0,45) \times \\ &\quad \times (1 - 0,18) = 3536249,859 \text{ (грн)}, \\ \Delta\Pi_3 &= (0 \times 300 + (2000 + 300) \times (3000+2000+1000) \times 0,8333 \times 0,45) \times \\ &\quad \times (1 - 0,18) = 4243499,830 \text{ (грн)}\end{aligned}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 9624749,62 грн.

Розраховуємо приведену вартість збільшення всіх чистих прибутків ПП (5.18), що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.18)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якого виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\begin{aligned}ПП &= (1844999,926 / (1 + 0,1)^1) + (3536249,859 / (1 + 0,1)^2) + \\ &\quad + (4243499,830 / (1 + 0,1)^3) = 7787997,435 \text{ (грн)}.\end{aligned}$$

Далі розраховують величину початкових інвестицій PV (5.19 і 5.20), які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} \times ЗВ, \quad (5.19)$$

$$PV = 2 \times 445000 = 890000,40 \text{ (грн)}, \quad (5.20)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{инв}=2\dots5$, але може бути і більшим;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки, грн.

Тоді абсолютний економічний ефект E_{abc} (5.21 і 5.22) або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = ПП - PV, \quad (5.21)$$

$$E_{abc} = 7787997,435 - 890000,40 = 6897997,04 \text{ (грн)} \quad (5.22)$$

Оскільки $E_{abc} > 0$ то вкладання коштів та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_g . Для цього використаємо формули (5.23 і 5.24):

$$E_g = \sqrt[T_{ж}]{1 + \frac{E_{обс}}{PV}} - 1, \quad (5.23)$$

$$E_g = \sqrt[3]{(1 + 6897997,04/890000,40) - 1} = 1,061, \quad (5.24)$$

де $T_{ж}$ – життєвий цикл наукової розробки, роки.

Визначимо мінімальну ставку дисконтування (4.25 і 4.26), яка у загальному вигляді визначається за формулою:

$$\tau_{min} = d + f, \quad (5.25)$$

$$\tau_{min} = 0,14 + 0,05 = 0,19, \quad (5.26)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = (0,09...0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай $f = (0,05...0,5)$.

Так як $E_g > \tau_{min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проєкту інвестицій за формулою (4.27 і 4.28):

$$T_{ок} = \frac{1}{E_g}, \quad (5.27)$$

$$T_{ок} = 1 / 1,061 = 0,94 (p). \quad (5.28)$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,94 роки, то фінансування даної наукової розробки є доцільним.

Висновок до розділу 5

Економічна частина даної роботи містить розрахунок витрат на розроблення нового програмного продукту, сума яких складає 445000 гривень. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення.

У результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною є дешевшим за аналог і є конкурентоспроможним.

Таким чином, оскільки період окупності такої розробки складає близько 0,94 р., то він є меншим за нормативно визначений термін (3 роки), що свідчить про економічну доцільність інвестиційних вкладень у запропоновану систему дистанційного навчання.

ВИСНОВОК

У магістерській кваліфікаційній роботі було спроектовано та розроблено web-систему для дистанційного вивчення слів англійської мови в межах підсистеми оцінювання якості навчання.

Проаналізовано існуючі підходи до організації контролю та оцінювання результатів навчання, в результаті чого запропоновано підхід до побудови власної web-системи дистанційного навчання.

Вивчено недоліки та переваги наявних комп'ютерних систем оцінювання якості дистанційного навчання, що дозволило обґрунтувати вимоги до розроблення власної системи дистанційного навчання.

Розроблено структуру підсистеми оцінювання якості навчання та наведено інтерфейс для її інтеграції до загальної web-системи вивчення слів англійської мови. Запропонована підсистема складається з трьох модулів, що працюють у послідовному режимі: перевірка, оцінювання та облік результатів навчання.

На базі аналізу існуючих математичних моделей було побудовано моделі для перевірки результатів навчання, оцінювання його якості та процесу обліку результатів навчання. Обґрунтовано вибір двопараметричної моделі Бірнбаума як основи для розроблення математичної моделі оцінювання результатів навчання та її переваги над одно- та трипараметричними моделями. Модель перевірки результатів навчання базується на лінійній моделі, вихідний параметр якої є вхідним параметром моделі процесу оцінювання якості навчання. Модель процесу оцінювання якості навчання розраховує уточнені результати перевірки знань респондента, враховуючи вагові коефіцієнти як виконаних завдань, так і метрик респондента. Для обліку результатів навчання створено концептуальну та логічну моделі даних, що базуються на реляційній моделі даних.

Обґрунтовано вибір інструментів для програмної реалізації розробки, для клієнтської та серверної частин окремо, а також для бази даних. Для клієнтської частини та відображення інтерфейсу користувача було вирішено розробляти Angular-застосунок. Для серверної частини ключовою технологією обрано .NET та стек його суміжних фреймворків. Сервер бази даних обрано Microsoft SQL Server.

У результаті роботи було розроблено web-застосунок та базу даних. Основний застосунок базується на тривірневій архітектурі та виконує основну логіку та обчислення над даними, а також надає інтерфейс керування програмою клієнтській частині.

Розроблено нормалізовану базу даних, що керується безпосередньо кодом додатку, але також має можливість ручного керування за допомогою Microsoft SQL Management Studio. Під час розроблення програмного комплексу було враховано можливість подальшого горизонтального масштабування системи, шляхом додавання до вже існуючої підсистеми оцінювання якості навчання інших підсистем, наприклад, підсистеми побудови навчального плану.

Здійснено тестування та аналіз роботи розроблені системи. Спочатку було проведено автоматизоване тестування за допомогою юніт-тестів, де детальну увагу приділено послідовності та коректності виконання модулів системи. Також було продемонстровано роботу ключових функцій та надано детальні пояснення дій програмних модулів, які за них відповідають.

Ключовими перевагами розроблюваної системи є наявність трьох основних та додаткового індивідуального типу перевірки знань, різноманітність форм завдань, аналіз результатів перевірки та представлення розширеної статистики респондента у вигляді інфографіків, можливість роботи з викладачем, а також відсутність реклами та платних обмежень.

Отже, задачі, поставлені у магістерській кваліфікаційній роботі, було виконано в повному обсязі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Система дистанційної колективної самопідготовки / О. Д. Азаров та ін. *Інформаційні технології та комп'ютерна інженерія*. 2016. № 2. Т. 2. С. 15–20.
2. Биков В. Ю. Теоретико-методологічні засади моделювання навчального середовища сучасних педагогічних систем : зб. наук. праць. Київ : Інститут засобів навчання АПН України, 2005. 272 с.
3. Simonson M., Zvacek S. M., Smaldino S. *Teaching and Learning at a Distance: Foundations of Distance Education*. 7th Ed. Charlotte : Information Age Publishing, 2019. 366 p.
4. Маркова Є. С. Інформаційні технології навчання : навч. посіб. Запоріжжя : Просвіта, 2012. 118 с.
5. Інформаційні та комунікаційні технології навчання в системі загальної середньої освіти зарубіжних країн / Гриценчук О. О. та ін. Київ : Пед. думка, 2012. 176 с.
6. Супрун П. Б. Особливості розробки веб-додатків на ASP.NET Core. *LI науково-технічна конференція ВНТУ* : Електронне наукове видання матеріалів конференції, м. Вінниця, 2022. URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2022/paper/view/14483/12251> (дата звернення 19.09.2022).
7. Ткаченко Л. В., Хмельницька О. С. Особливості впровадження дистанційного навчання в освітній процес закладу вищої освіти. *Педагогіка формування творчої особистості у вищій і загальноосвітній школах*. 2021. № 75. Т. 3. С. 91–96.
8. Організація дистанційного навчання. Створення електронних навчальних курсів та електронних тестів / Вишнівський В. В. та ін. Київ : ДУТ, 2014. 140 с.
9. Дистанційне навчання. Як організувати навчання вдома і не зійти з розуму / Вайзман Р. та ін. Київ : Альпіна паблішер, 2021. 240 с.

10. Буйницька О. Інформаційні технології та технічні засоби навчання : навч. посіб. Київ : Центр навч. літератури, 2019. 240 с.
11. Memrise – онлайн сервіс для вивчення слів іноземних мов. *Memrise* : веб-сайт. URL: <https://www.memrise.com/> (дата звернення 24.09.2022).
12. Lingualéo – онлайн сервіс для вивчення англійської та інших іноземних мов. *Lingualéo* : веб-сайт. URL : <https://lingualéo.com/> (дата звернення 24.09.2022).
13. Duolingo – самий швидкий шлях вивчити іноземну мову. *Duolingo* : веб-сайт URL : <https://www.duolingo.com/> (дата звернення 24.09.2022).
14. Винарчук Т. М. Веб-статистика як інструмент аналізу сайту. *Народна освіта*. 2014. № 23, Т. 6. URL: https://www.narodnaosvita.kiev.ua/?page_id=2424 (дата звернення 12.10.2022).
15. Bliss K. Libertiny J. What is Mathematical Modelling. 2th Ed. COMAP & SIAM, 2019. 236 p.
16. Yodgorov G., Jurakulov T. Mathematical modeling of learning processes based on the theory of control. *Conference Proceedings*. 2021. № 1, Vol. 2365. URL: <https://aip.scitation.org/doi/10.1063/5.0057821> (accessed: 12.10.2022).
17. Zhelezniakova E., Silichova T. Mathematical modeling as a means of control and evaluation of the quality of the educational process. *Modestum*. 2020. № 2, Vol. 16. URL: http://repository.hneu.edu.ua/bitstream/123456789/25791/1/Zhelezniakova_Silichova%20%281%29.pdf (accessed: 14.10.2022).
18. Sun B. Mathematical Models of Learning Efficiency. *Modestum*. 2017. № 7, Vol. 13. P. 4261–4270.
19. Ivanichkina L., Neporada A. Mathematical Methods and Models of Improving Data Storage Reliability Including Those Based on Finite Field Theory. *Contemporary Engineering Sciences*. 2014. № 28, Vol. 7. P. 1589–1602.

20. An introduction to web development technologies. *Tiller* : website. URL: <https://tillerdigital.com/blog/an-introduction-to-web-development-technologies/> (accessed: 12.10.2022)
21. ng-book: The Complete Guide to Angular 4. Coury F. and oth. 4th Ed. Amazon Digital Services, 2017. 597 p.
22. Richter J. CLR via C# (Developer Reference). Microsoft Press, 2012. 896 p.
23. Mark J. Price. C# 11 and .NET 7 – Modern Cross-Platform Development Fundamentals: Start building websites and services with ASP.NET Core 7, Blazor, and EF Core 7. Packt Publishing, 2022. 818 p.
24. Мартін Р. Чиста Архітектура. Київ : Фабула, 2019. 416 p.
25. Прайс М. C# 9 та .NET 5. Розробка та оптимізація. Packt, 2021. 832 с.
26. Hughes A. Microsoft SQL Server. *TeachTarget* : website. URL: <https://www.techtarget.com/searchdatamanagement/definition/SQL-Server/> (accessed: 12.10.2022).
27. Багаторівнева архітектура. *Metanit* : веб-сайт. URL: <https://metanit.com/sharp/mvc5/23.5.php/> (accessed: 15.10.2022).
28. Entity Framework Core. *Microsoft* : website. URL: <https://learn.microsoft.com/ru-ru/ef/core/> (accessed: 15.10.2022).
29. Quickstart: Install and use a NuGet package in Visual Studio. *Microsoft* : website. URL: <https://learn.microsoft.com/en-us/nuget/quickstart/install-and-use-a-package-in-visual-studio/> (accessed: 15.10.2022).
30. Фрімен Е., Робсон Е. Характеристики Head First. Патерни проектування. Київ : Фабула, 2021. 688 с.
31. Unit of Work in Repository Pattern. *C# Corner* : website. URL: <https://www.c-sharpcorner.com/UploadFile/b1df45/unit-of-work-in-repository-pattern/> (accessed: 16.10.2022).

32. Who Disposes Your Repository. *On Code Design* : website. URL: <https://oncodedesign.com/who-disposes-your-repository/> (accessed: 16.10.2022).

33. Скільки шкіл в Україні готові до очного навчання: відповідь міністра освіти. *TCH* : веб-сайт. URL: <https://tsn.ua/ukrayina/skilki-shkil-v-ukrayini-gotovi-do-ochnogo-navchannya-vidpovid-ministra-osviti-2143093.html/> (дата звернення 20.11.2022).

ДОДАТОК А

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри обчислювальної техніки

_____ проф., д.т.н. О. Д. Азаров

«__» _____ 2022 року

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання комплексної магістерської кваліфікаційної роботи
Web-система для дистанційного вивчення слів англійської мови.

Частина 1 «Підсистема оцінювання якості навчання»

08-23.КМКР.032.00.000 ТЗ

Науковий керівник к.т.н., проф. каф. ОТ

_____ Азарова А. О.

Студента групи 1КІ-21м

_____ Супруна П. Б.

1 Підставою для виконання комплексної магістерської кваліфікаційної роботи є наказ про затвердження теми дипломної роботи, а також актуальність розробки системи для вивчення слів іноземної мови, яка би забезпечувала ефективний процес навчання іноземної мови, оцінюючи показники його академічної успішності.

2 Мета і призначення КМКР:

- метою є розробка веб-системи для вивчення слів іноземної мови та підсистеми оцінювання якості навчання;
- призначенням розробки є виконання комплексної магістерської кваліфікаційної роботи з можливістю подальшого впровадження та масштабування.

3 Вихідні дані для виконання КМКР:

- розробити веб-систему для дистанційного вивчення слів англійської мови, а також підсистему для оцінювання якості навчання;
- основна технологія .NET Core;
- середовище програмування Microsoft Visual Studio 2022;
- мова програмування C#, SQL, TypeScript.

4 Технічні вимоги до виконання КМКР:

- виведення математичної моделі оцінювання якості навчання.
- впровадження підсистеми для оцінювання якості навчання.

5 Етапи КМКР та очікувані результати (табл. А.1).

Таблиця А.1 – Етапи роботи

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Дослідження методів дистанційного вивчення іноземної мови.	01.09.2022	09.09.2022	Розділ 1
2	Математичне моделювання процесу оцінювання якості навчання	12.09.2022	26.09.2022	Розділ 2

Продовження таблиці А.1

3	Аналіз та вибір технологій проектування та розробки веб-додатків	3.10.2022	10.10.2022	Розділ 3 частково
4	Програмна реалізація системи	17.10.2022	31.10.2022	Розділ 3 повністю
5	Підготовка економічної частини	7.11.2022	14.11.2022	Розділ 4
6	Оформлення матеріалів до захисту КМКР	21.11.2022	1.12.2022	Пояснювальна записка, графічний матеріал, лістинг, презентація

6 Матеріали, що подаються до захисту КМКР:

- пояснювальна записка КМКР;
- графічні і ілюстративні матеріали;
- протокол попереднього захисту КМКР на кафедрі;
- відгук наукового керівника;
- рецензія на виконану роботу;
- анотації до КМКР українською та іноземною мовами;
- нормоконтроль про відповідність оформлення КМКР діючим вимогам.

7 Порядок контролю виконання та захисту КМКР:

- виконання етапів графічної та розрахункової документації КМКР контролюється науковим керівником згідно зі встановленими термінами;
- захист КМКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8 Вимоги до оформлення КМКР викладені в методичних вказівках до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп'ютерна інженерія, ДСТУ 3008 : 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання», ДСТУ 8302 : 2015 «Бібліографічні посилання. Загальні положення та правила складання».

ДОДАТОК Б

Структура системи

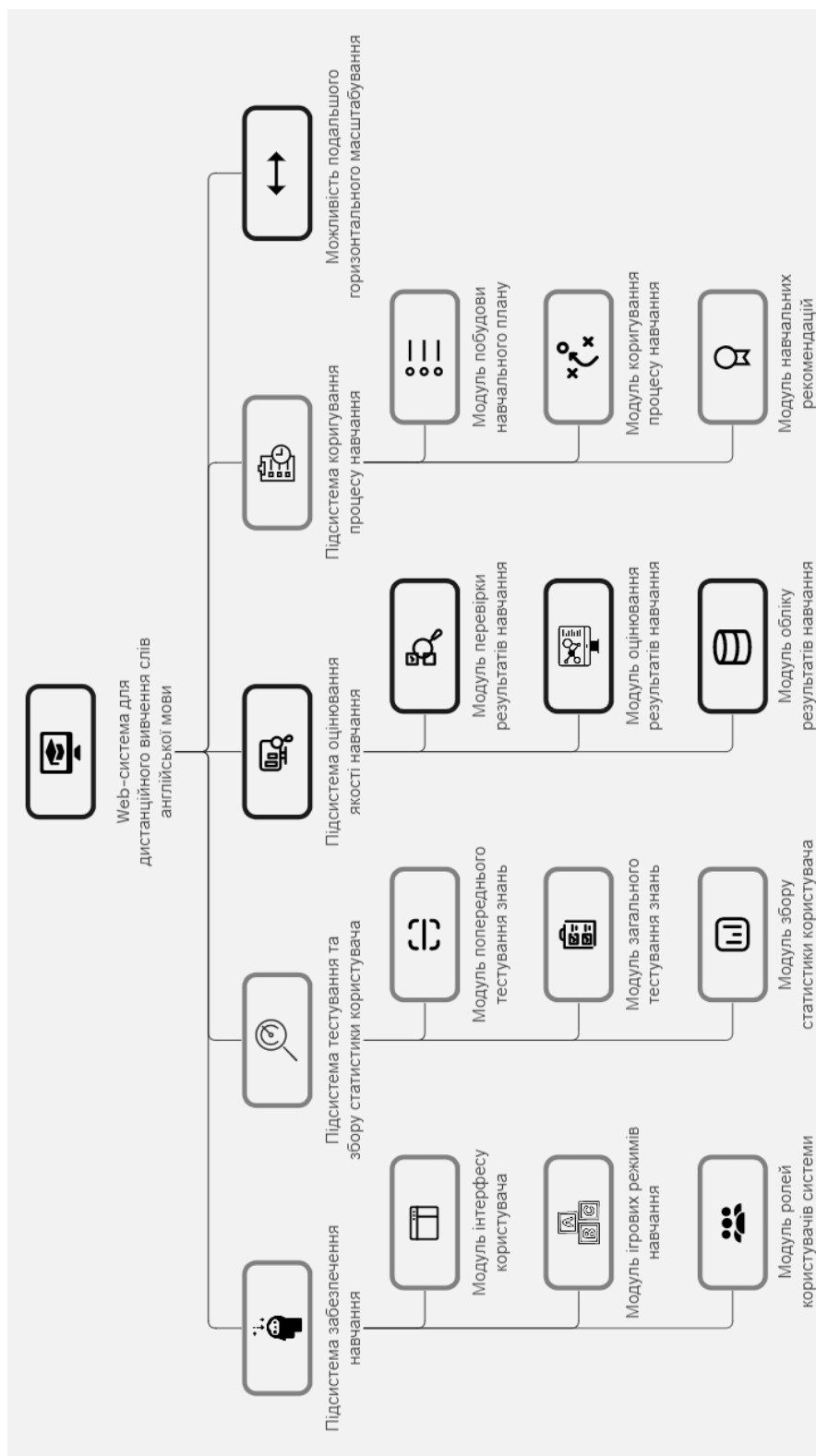


Рисунок Б.1 — Структурна схема системи

ДОДАТОК В

Алгоритм роботи системи

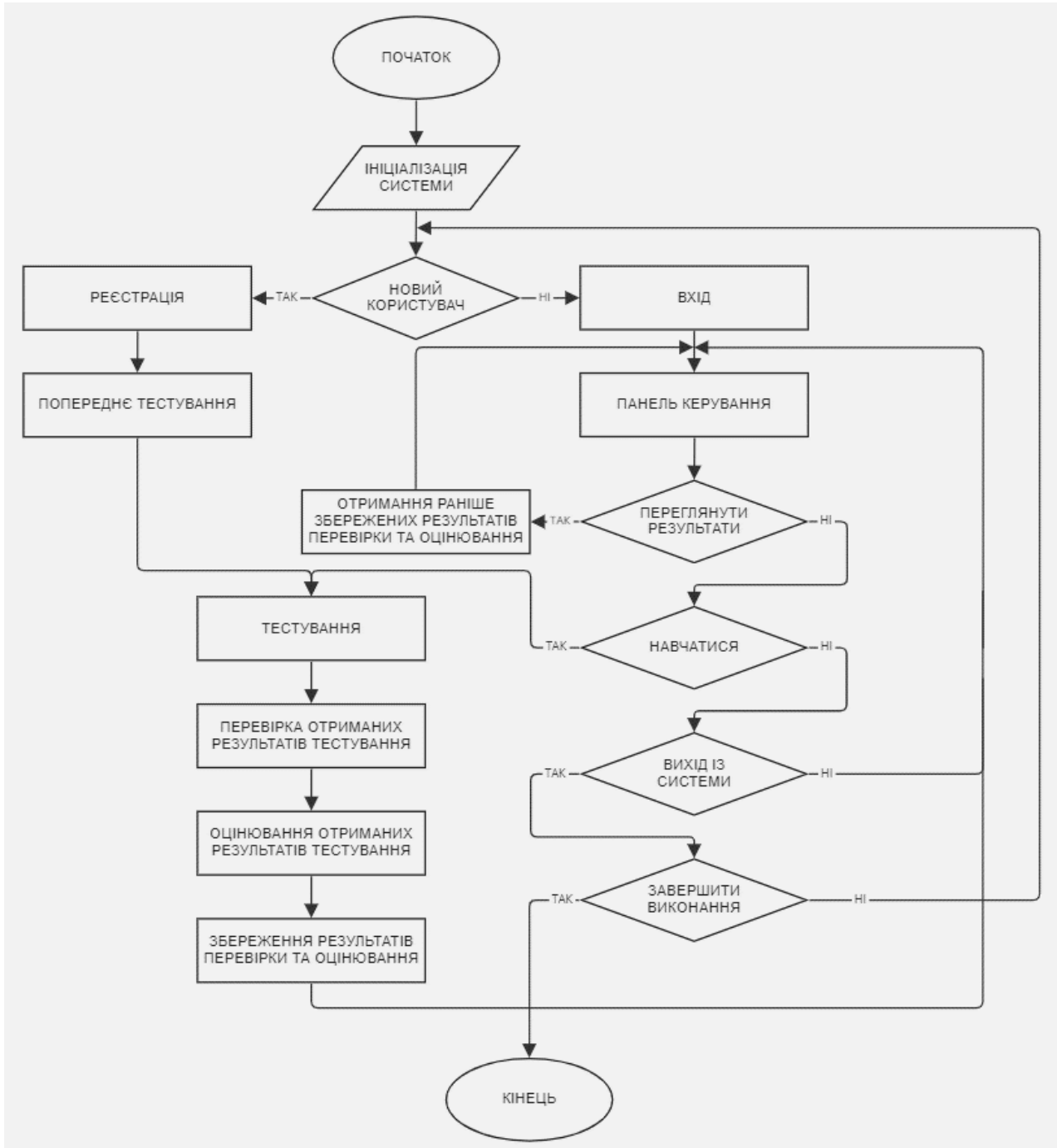


Рисунок В.1 — Алгоритм роботи інтегрованої підсистеми оцінювання якості навчання

ДОДАТОК Г

Лістинг репозиторіїв

```
public interface IWordRepository : IDisposable
{
    Task<Word?> GetByIdAsync(int id);
    Task<IEnumerable<Word>> GetAllAsync();
    Task CreateAsync(Word entity);
    Task DeleteByIdAsync(int id);
    void Update(Word entity);
}

public interface ITopicRepository : IDisposable
{
    Task<Topic?> GetByIdWithWordsAsync(int id);
    Task<IEnumerable<Topic>> GetAllAsync();
    Task CreateAsync(Topic entity);
    Task DeleteByIdAsync(int id);
    void Update(Topic entity);
}

public class WordRepository : IWordRepository
{
    private readonly EngurDbContext _context;
    public WordRepository(EngurDbContext context)
    {
        _context = context;
    }
    public async Task CreateAsync(Word word)
```

```
{
    if (word == null)
    {
        throw new ArgumentNullException(nameof(word));
    }
    await _context.Words.AddAsync(word);
}

public async Task DeleteByIdAsync(int id)
{
    var word = await _context.Words.FirstOrDefaultAsync(w => w.Id == id);
    if (word != null)
    {
        _context.Words.Remove(word);
    }
}

public async Task<IEnumerable<Word>> GetAllAsync()
{
    return await _context.Words.ToListAsync();
}

public async Task<Word?> GetByIdAsync(int id)
{
    return await _context.Words.FirstOrDefaultAsync(t => t.Id == id);
}

public void Update(Word word)
{
    _context.Words.Update(word);
}
```

```
public void Dispose()
{
    _context.Dispose();
}
}

public class TopicRepository : ITopicRepository
{
    private readonly EngurDbContext _context;
    public TopicRepository(EngurDbContext context)
    {
        _context = context;
    }
    public async Task CreateAsync(Topic topic)
    {
        if (topic == null)
        {
            throw new ArgumentNullException(nameof(topic));
        }
        await _context.Topics.AddAsync(topic);
    }
    public async Task DeleteByIdAsync(int id)
    {
        var topic = await _context.Topics.FirstOrDefaultAsync(t => t.Id == id);
        if (topic != null)
        {
            _context.Topics.Remove(topic);
        }
    }
}
```

```
}  
public async Task<IEnumerable<Topic>> GetAllAsync()  
{  
    return await _context.Topics.ToListAsync();  
}  
public async Task<Topic?> GetByIdWithWordsAsync(int id)  
{  
    return await _context.Topics  
        .Include(t => t.Words)  
        .FirstOrDefaultAsync(t => t.Id == id);  
}  
public void Update(Topic topic)  
{  
    _context.Topics.Update(topic);  
}  
public void Dispose()  
{  
    _context.Dispose();  
}  
}
```

ДОДАТОК Д
Лістинг сервісів

```
public interface IWordService
{
    Task<WordDto?> GetByIdAsync(int id);
    Task<IEnumerable<WordDto>> GetAllAsync();
    Task CreateAsync(WordDto word);
    Task DeleteByIdAsync(int id);
    Task UpdateAsync(WordDto word);
}

public interface ITopicService
{
    Task<TopicDto?> GetByIdWithWordsAsync(int id);
    Task<IEnumerable<TopicDto>> GetAllAsync();
    Task CreateAsync(TopicDto word);
    Task DeleteByIdAsync(int id);
    Task UpdateAsync(TopicDto word);
}

public class WordService : IWordService
{
    private readonly IUnitOfWork _unitOfWork;
    public WordService(IUnitOfWork uow)
    {
        _unitOfWork = uow;
    }
    public async Task CreateAsync(WordDto word)
```

```
{
    await
_unitOfWork.WordRepository.CreateAsync(WordDto.MapFromDto(word));
    await _unitOfWork.SaveChangesAsync();
}
public async Task DeleteByIdAsync(int id)
{
    await _unitOfWork.WordRepository.DeleteByIdAsync(id);
    await _unitOfWork.SaveChangesAsync();
}
public async Task<IEnumerable<WordDto>> GetAllAsync()
{
    var words = await _unitOfWork.WordRepository.GetAllAsync();
    return WordDto.MapToDtos(words);
}
public async Task<WordDto?> GetByIdAsync(int id)
{
    var word = await _unitOfWork.WordRepository.GetByIdAsync(id);
    if (word == null)
    {
        throw new ArgumentException(nameof(word));
    }
    return WordDto.MapToDto(word);
}
public async Task UpdateAsync(WordDto word)
{
    _unitOfWork.WordRepository.Update(WordDto.MapFromDto(word));
}
```

```
        await _unitOfWork.SaveChangesAsync();
    }
}

public class TopicService : ITopicService
{
    private readonly IUnitOfWork _unitOfWork;
    public TopicService(IUnitOfWork uow)
    {
        _unitOfWork = uow;
    }
    public async Task CreateAsync(TopicDto topic)
    {
        await
        _unitOfWork.TopicRepository.CreateAsync(TopicDto.MapFromDto(topic));
        await _unitOfWork.SaveChangesAsync();
    }
    public async Task DeleteByIdAsync(int id)
    {
        await _unitOfWork.TopicRepository.DeleteByIdAsync(id);
        await _unitOfWork.SaveChangesAsync();
    }
    public async Task<IEnumerable<TopicDto>> GetAllAsync()
    {
        var topics = await _unitOfWork.TopicRepository.GetAllAsync();
        return TopicDto.MapToDtos(topics);
    }
    public async Task<TopicDto?> GetByIdWithWordsAsync(int id)
```

```
{
    var topic = await _unitOfWork.TopicRepository.GetByIdWithWordsAsync(id);
    if(topic == null)
    {
        throw new ArgumentException(nameof(topic));
    }
    return TopicDto.MapFromDto(topic);
}
public async Task UpdateAsync(TopicDto topic)
{
    _unitOfWork.TopicRepository.Update(TopicDto.MapFromDto(topic));
    await _unitOfWork.SaveChangesAsync();
}
}
public class WordDto
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Translation { get; set; }
    public int TopicId { get; set; }
    public static WordDto MapToDto(Word word)
    {
        return new WordDto()
        {
            Id = word.Id,
            Name = word.Name,
            Translation = word.Translation,
```



```
        TopicId = word.TopicId
    };
}
public static IEnumerable<WordDto> MapToDtos(IEnumerable<Word> words)
{
    var wordDtos = new List<WordDto>();
    if(words != null)
    {
        foreach (var word in words)
        {
            wordDtos.Add(MapToDto(word));
        }
    }
    return wordDtos;
}
public static Word MapFromDto(WordDto word)
{
    return new Word()
    {
        Id = word.Id,
        Name = word.Name,
        Translation = word.Translation,
        TopicId = word.TopicId
    };
}
}
public class TopicDto
```

```
{
    public int Id { get; set; }
    public string Name { get; set; }
    public IEnumerable<WordDto>? Words { get; set; }
    public static TopicDto MapToDto(Topic topic)
    {
        return new TopicDto()
        {
            Id = topic.Id,
            Name = topic.Name,
            Words = WordDto.MapToDtos(topic.Words)
        };
    }
    public static IEnumerable<TopicDto> MapToDtos(IEnumerable<Topic> topics)
    {
        var topicDtos = new List<TopicDto>();
        foreach (var topic in topics)
        {
            topicDtos.Add(MapToDto(topic));
        }
        return topicDtos;
    }
}
```

ДОДАТОК Е

ЛІСТИНГ КОНТРОЛЕРІВ

```
[Route("api/[controller]")]
[ApiController]
[Authorize]
public class WordController : ControllerBase
{
    private readonly IWordService _wordService;
    public WordController(IWordService wordService)
    {
        _wordService = wordService;
    }
    [HttpGet]
    public async Task<IEnumerable<WordDto>> GetAllWords()
    {
        return await _wordService.GetAllAsync();
    }
    [HttpGet("{id}")]
    public async Task<IResult> GetWordById(int id)
    {
        var word = await _wordService.GetByIdAsync(id);
        return word == null ? Results.NotFound() : Results.Ok(word);
    }
    [HttpPost]
    public async Task<IResult> CreateWord(WordDto wordDto)
    {
```

```
        await _wordService.CreateAsync(wordDto);
        return Results.Ok();
    }
    [HttpPut]
    public async Task<IResult> UpdateWord(WordDto wordDto)
    {
        await _wordService.UpdateAsync(wordDto);
        return Results.Ok();
    }
    [HttpDelete("{id}")]
    public async Task<IResult> DeleteWord(int id)
    {
        await _wordService.DeleteByIdAsync(id);
        return Results.Ok();
    }
}

[Route("api/[controller]")]
[ApiController]
[Authorize]
public class TopicController : ControllerBase
{
    private readonly ITopicService _topicService;
    public TopicController(ITopicService topicService)
    {
        _topicService = topicService;
    }
    [HttpGet]
```

```
public async Task<IEnumerable<TopicDto>> GetAllTopics()
{
    return await _topicService.GetAllAsync();
}

[HttpGet("{id}")]
public async Task<IResult> GetTopicByIdWithWordsAsync(int id)
{
    var topic = await _topicService.GetByIdWithWordsAsync(id);
    return topic == null ? Results.NotFound() : Results.Ok(topic);
}

[HttpPost]
public async Task<IResult> CreateTopic(TopicDto topicDto)
{
    await _topicService.CreateAsync(topicDto);
    return Results.Ok();
}

[HttpPut]
public async Task<IResult> UpdateTopic(TopicDto topicDto)
{
    await _topicService.UpdateAsync(topicDto);
    return Results.Ok();
}

[HttpDelete("{id}")]
public async Task<IResult> DeleteTopic(int id)
{
    await _topicService.DeleteByIdAsync(id);
    return Results.Ok();
}
```

```

    }
}
[Route("api/[controller]")]
[ApiController]
public class AccountController : ControllerBase
{
    private readonly UserManager<IdentityUser> _userManager;
    private readonly RoleManager<IdentityRole> _roleManager;
    private readonly IConfiguration _configuration;
    public AccountController(
        UserManager<IdentityUser> userManager,
        RoleManager<IdentityRole> roleManager,
        IConfiguration configuration)
    {
        _userManager = userManager;
        _roleManager = roleManager;
        _configuration = configuration;
    }
    [HttpPost]
    [Route("login")]
    public async Task<IResult> Login([FromBody] AccountDto dto)
    {
        var user = await _userManager.FindByNameAsync(dto.Username);
        if (user != null && await _userManager.CheckPasswordAsync(user,
dto.Password))
        {
            var userRoles = await _userManager.GetRolesAsync(user);

```

```

var authClaims = new List<Claim>
{
    new Claim(ClaimTypes.Name, user.UserName),
    new Claim(JwtRegisteredClaimNames.Jti, Guid.NewGuid().ToString()),
};
foreach (var userRole in userRoles)
{
    authClaims.Add(new Claim(ClaimTypes.Role, userRole));
}
var token = GetToken(authClaims);
return Results.Ok(new
{
    token = new JwtSecurityTokenHandler().WriteToken(token),
    expiration = token.ValidTo
});
}
return Results.Unauthorized();
}
[HttpPost]
[Route("register")]
public async Task<IResult> Register([FromBody] AccountDto dto)
{
    var userExists = await _userManager.FindByNameAsync(dto.Username);
    if (userExists != null)
    {
        return Results.BadRequest(new { Status = "Error", Message = "User already
exists!" });
    }
}

```

```
}  
IdentityUser user = new()  
{  
    Email = dto.Email,  
    SecurityStamp = Guid.NewGuid().ToString(),  
    UserName = dto.Username  
};  
var result = await _userManager.CreateAsync(user, dto.Password);  
if (!result.Succeeded)  
{  
    return Results.BadRequest(new { Status = "Error", Message = "User creation  
failed! Please check the credentials." });  
}  
if(dto.Role == "Teacher")  
{  
    if (!await _roleManager.RoleExistsAsync(UserRole.Admin))  
    {  
        await _roleManager.CreateAsync(new IdentityRole(UserRole.Admin));  
    }  
    await _userManager.AddToRoleAsync(user, UserRole.Admin);  
}  
else  
{  
    if (!await _roleManager.RoleExistsAsync(UserRole.User))  
    {  
        await _roleManager.CreateAsync(new IdentityRole(UserRole.User));  
    }  
}
```



```

        await _userManager.AddToRoleAsync(user, UserRole.User);
    }
    return Results.Ok();
}
private JwtSecurityToken GetToken(List<Claim> authClaims)
{
    var authSigningKey = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes(_configuration["JWT:Secret"]));
    var token = new JwtSecurityToken(
        issuer: _configuration["JWT:ValidIssuer"],
        audience: _configuration["JWT:ValidAudience"],
        expires: DateTime.Now.AddHours(3),
        claims: authClaims,
        signingCredentials: new SigningCredentials(authSigningKey,
SecurityAlgorithms.HmacSha256)
    );
    return token;
}
}

```

ДОДАТОК Є

Ілюстративний матеріал до захисту

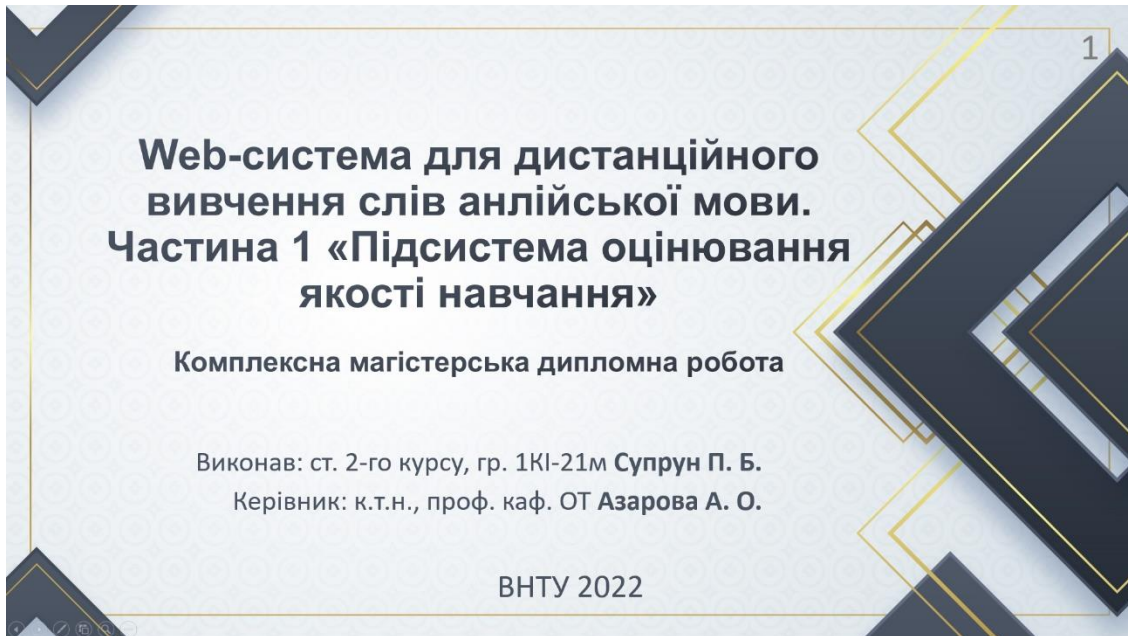


Рисунок Є.1 — Вступний слайд презентації

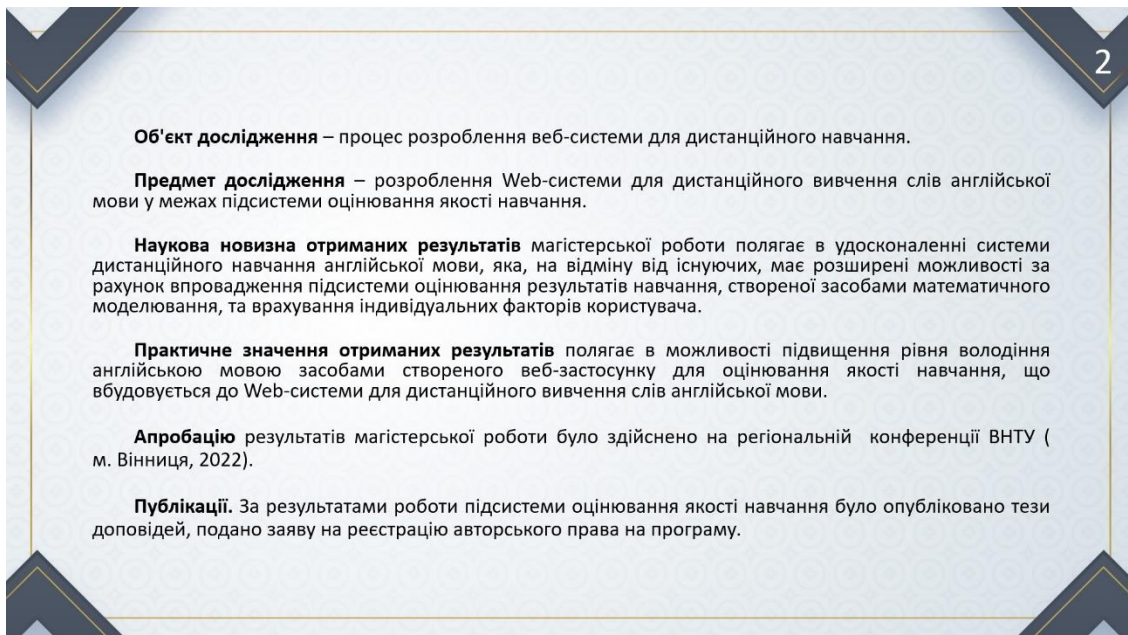


Рисунок Є.2 — Слайд з основною інформацією

Вивчення провідних систем-аналогів 3

Сьогодні на ринку є безліч систем для вивчення мов та слів, кожна з яких за замовчуванням має базовий набір функцій. Отже, вибір залишається робити на основі зручності інтерфейсу або функціональності чи наявності обмежень платної підписки.

Для порівняння обрано 3 провідних системи галузі: Duolingo, LinguaLeo та Memrise.

Рисунок Є.3 — Слайд із системами-аналогами

Підсистема оцінювання якості навчання 4

Структура та алгоритм роботи підсистеми оцінювання якості навчання

Рисунок Є.4 — Слайд зі структурою та алгоритмом підсистеми оцінювання якості навчання

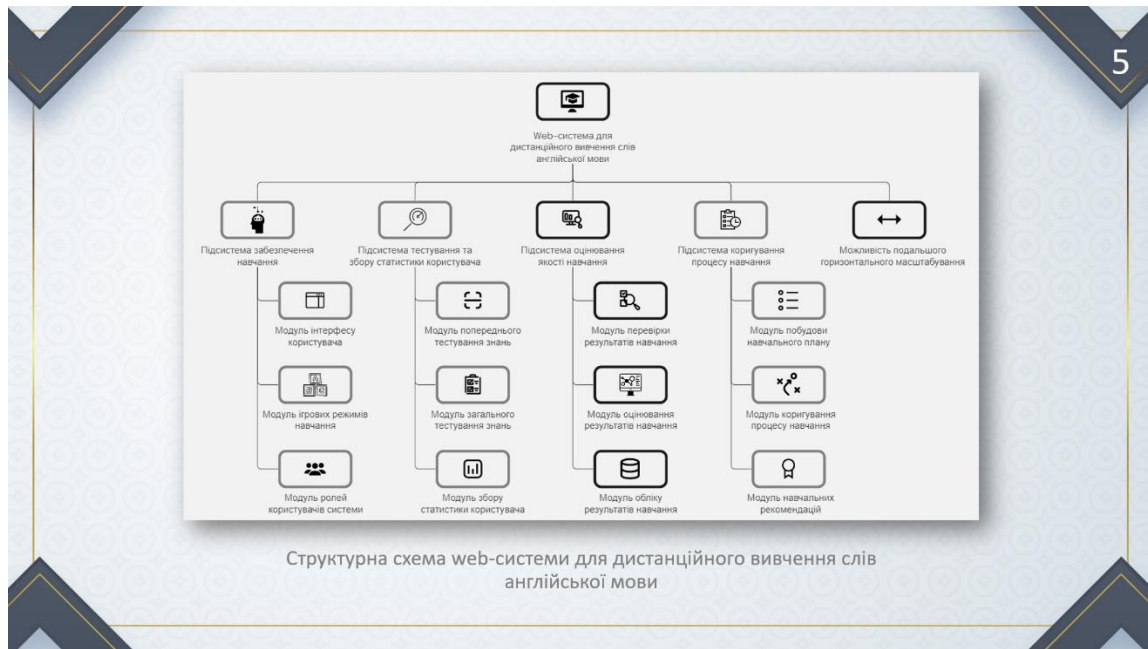


Рисунок Є.5 — Слайд зі структурою web-системи для дистанційного вивчення слів англійської мови

6

Математична модель процесу перевірки результатів навчання

Для компонента перевірки результатів навчання в магістерській роботі пропонується модель, яка оцінює відповідь суб'єкта тестування за двобальною шкалою (правильно або неправильно):

$$R = \frac{\sum_{i=1}^k R_i}{n},$$

де R_i – правильна відповідь суб'єкта тестування на i -е завдання;
 k – кількість правильних відповідей із n запропонованих ($k \leq n$).

До переваг даної моделі варто віднести простоту її реалізації.

Рисунок Є.6 — Слайд із математичною моделлю перевірки якості навчання

7

Математична модель процесу оцінювання якості навчання

Додатково для покращення процесу контролю був розроблений алгоритм, базований на класифікації завдань за їх дидактичними характеристиками. Спочатку вираховується число балів, отриманих респондентом за виконання n завдань:

$$y = \sum_{i=1}^n w_i R_i,$$

де R_i – оцінка за виконання i -го завдання;
 n – число завдань;
 w_i – вектор вагових коефіцієнтів завдань, що залежить від їх дидактичних характеристик.

Рисунок Є.7 — Слайд із початковою математичною моделлю оцінювання якості навчання

8

Математична модель процесу оцінювання якості навчання

По завершенню перевірки встановлюється середній бал A , отриманий респондентом за виконання n завдань:

$$A = \frac{y}{k_n},$$

де y – загальне число балів, отриманих респондентом;
 k_n – кількість спроб виконання n завдань, $k_n \geq n$.

На основі попередньої формули, вираховується значення уточненого середнього балу A' :

$$A' = A \times r + a_1 \frac{k_n}{n} + a_2 \frac{k_c}{n} + a_3 \frac{k_b}{n},$$

де r – коефіцієнт респондента, який призначається після проходження попередньої перевірки знань;
 k_n – кількість спроб виконання n завдань;
 k_c – кількість звернень до довідкової інформації;
 k_b – кількість завдань, виконаних з перевищенням відведеного часу ($k_b \leq n$);
 a_1, a_2, a_3 – вагові коефіцієнти, задаються статично в системі (0...1).

Рисунок Є.8 — Слайд із остаточною математичною моделлю оцінювання якості навчання

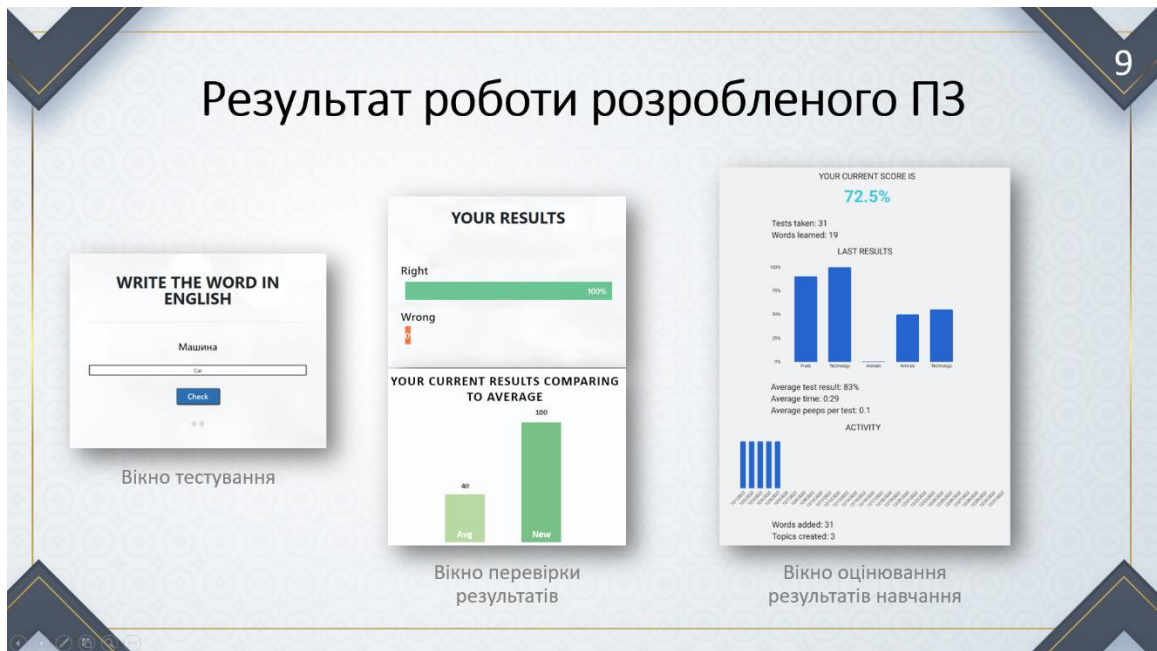


Рисунок Є.9 — Слайд результатів роботи розробленого ПЗ

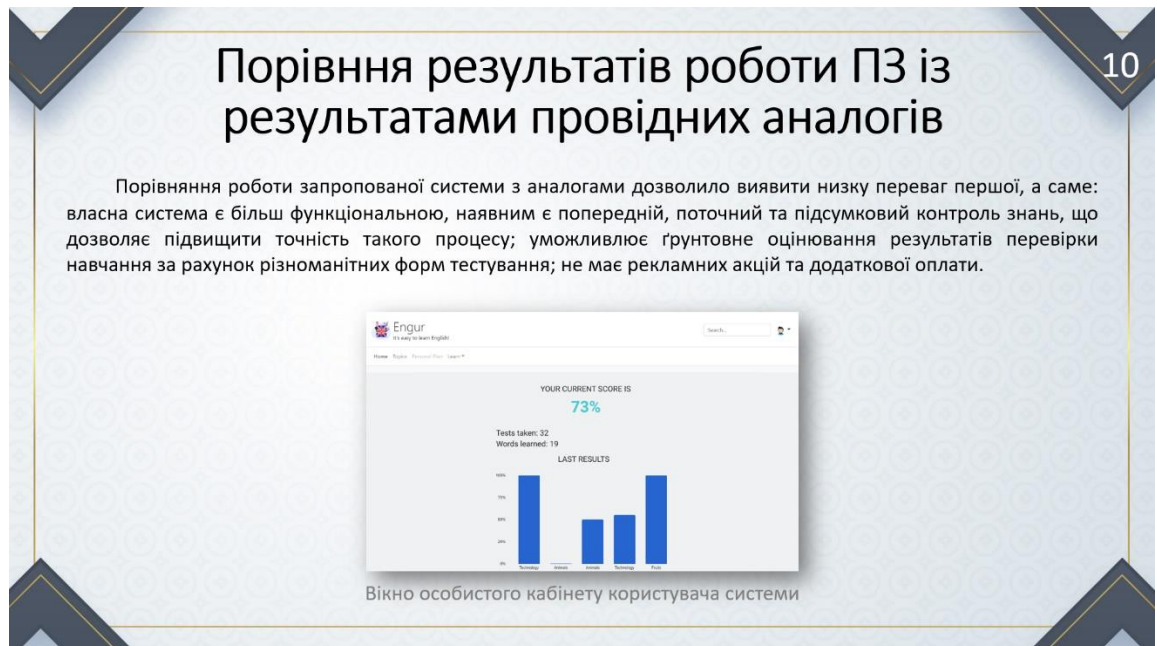
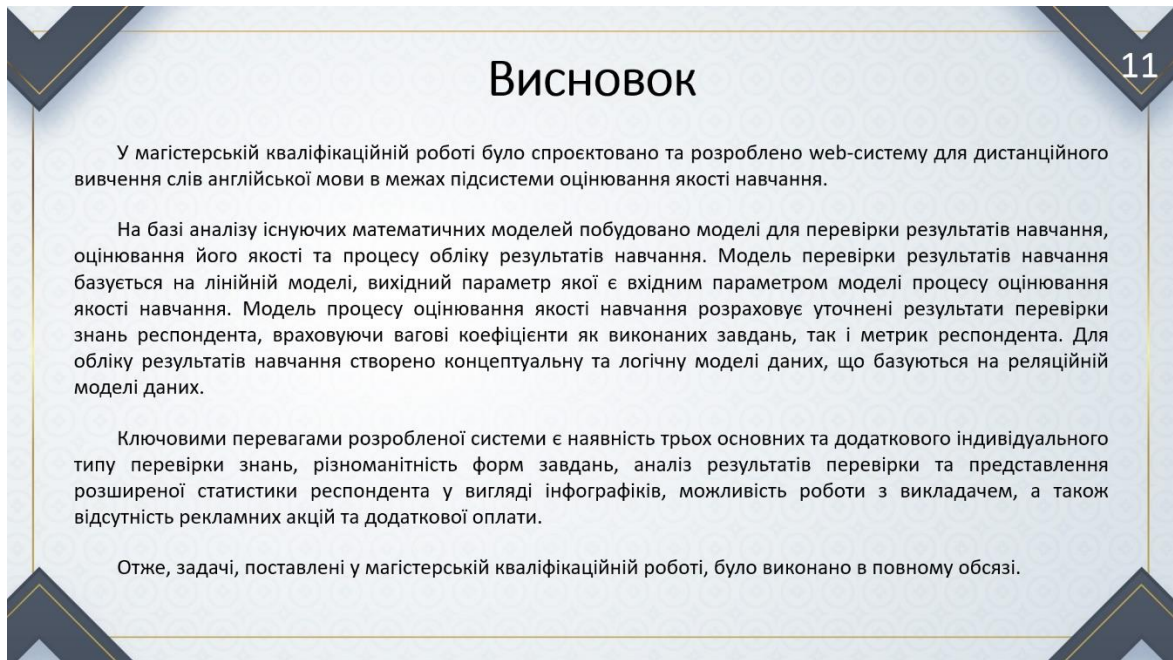


Рисунок Є.10 — Слайд з порівняннями результатів роботи ПЗ із результатами провідних аналогів



11

Висновок

У магістерській кваліфікаційній роботі було спроектовано та розроблено web-систему для дистанційного вивчення слів англійської мови в межах підсистеми оцінювання якості навчання.

На базі аналізу існуючих математичних моделей побудовано моделі для перевірки результатів навчання, оцінювання його якості та процесу обліку результатів навчання. Модель перевірки результатів навчання базується на лінійній моделі, вихідний параметр якої є вхідним параметром моделі процесу оцінювання якості навчання. Модель процесу оцінювання якості навчання розраховує уточнені результати перевірки знань респондента, враховуючи вагові коефіцієнти як виконаних завдань, так і метрик респондента. Для обліку результатів навчання створено концептуальну та логічну моделі даних, що базуються на реляційній моделі даних.

Ключовими перевагами розробленої системи є наявність трьох основних та додаткового індивідуального типу перевірки знань, різноманітність форм завдань, аналіз результатів перевірки та представлення розширеної статистики респондента у вигляді інфографіків, можливість роботи з викладачем, а також відсутність рекламних акцій та додаткової оплати.

Отже, задачі, поставлені у магістерській кваліфікаційній роботі, було виконано в повному обсязі.

Рисунок Є.11 — Слайд з висновком

ДОДАТОК Ж

Протокол перевірки кваліфікаційної роботи

Назва роботи: Web-система для дистанційного вивчення слів англійської мови.
 Частина 1. «Підсистема оцінювання якості навчання»

Тип роботи: _____ магістерська кваліфікаційна робота _____
 (БДР, МКР)

Підрозділ _____ кафедра обчислювальної техніки _____
 (кафедра, факультет)

Показники звіту подібності Unichesk

Оригінальність _____ 86.5% _____ Схожість _____ 13.5% _____

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Захарченко С.М.
 (підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unichesk щодо роботи.

Автор роботи _____ Супрун П. Б.
 (підпис) (прізвище, ініціали)

Керівник роботи _____ Азарова А. О.
 (підпис) (прізвище, ініціали)