

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з п'яти розділів, містить 26 рисунків, 20 таблиць, 2 додатки, 23 джерела.

Дипломний проект присвячений розробці програмного продукту для досягнення наступних цілей: надання викладачам сервісу з управління тестуванням; пришвидшення процесу тестування шляхом автоматизації перевірки відповідей; забезпечення для студентів сервісу з проходження тестування. До комплексу задач розробки слід віднести можливість доповнення і редагування змісту системи тестування, проходження створених тестів студентами, автоматичну перевірку відповідей студентів, отримання інформації про результати тестування.

У розділі інформаційного забезпечення описані вхідні та вихідні дані, а також описана структура БД проекту.

Розділ математичного забезпечення присвячений опису змістовної та математичної постановок задачі, обґрунтування методу розв'язання, та опису застосованих методів розв'язання.

Програмний продукт створений описаними в роботі засобами відповідно до вимог технічного завдання та описаної структури.

У технологічному розділі описано керівництво користувача і наведена інформація щодо випробування програмного продукту.

**СИСТЕМА ТЕСТУВАННЯ, ПИТАННЯ ВІДКРИТОГО ТИПУ,
СЕМАНТИЧНИЙ АНАЛІЗ, ВЕБ-ЗАСТОСУНОК**

					ДП ІС-5127.1181-с.ПЗ			
Зм.	Арк.	Прізвище	Підпис	Дата				
Розроб.		Сулейманов А.К.			<i>Семантичний аналіз відповідей природною мовою в текстових системах</i>	Літ.	Аркуш	Аркушів
Перевірив.		Звенігородський О.С.					2	81
Н. кон.		Москаленко Н.В.				<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51</i>		
Затв.		Павлов О.А.						

ABSTRACT

Structure and scope of work. The explanatory note of the graduation project consists of five sections, contains 26 drawings, 20 tables, 2 applications, 23 sources.

The graduation project is devoted to the development of a software product to achieve the following objectives: providing teachers with service management for testing; reducing the time spent on processing the results of testing by introducing automated answer verification; possibility for students to execute given tests in the context of a system. The set of development tasks should include: the possibility to supplement the content of the testing system, the possibility to edit it, possibility of tests executing for students, automated answer verification of students answers, obtaining information about the results of the testing.

The information support section describes the input and output data considered, as well as the project database structure is described.

The section of mathematical support contains the conceptual and mathematical formulation of the problem, the rationale behind the solution methods of and the description of the applied methods of solution.

The software is created using the tools described in this project, in accordance with the requirements of the terms of reference and in accordance with the structure described in the software and hardware section.

The technology section describes the user manual and provides information about software product testing.

TESTING SYSTEM, OPEN ANSWER QUESTIONS, SEMANTIC ANALYSIS, WEB APPLICATION

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	5
ВСТУП.....	6
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	8
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА.....	8
1.1.1 <i>Опис процесу діяльності.....</i>	<i>9</i>
1.1.2 <i>Опис функціональної моделі</i>	<i>10</i>
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ.....	11
1.3 ПОСТАНОВКА ЗАДАЧІ.....	12
1.3.1 <i>Призначення розробки.....</i>	<i>13</i>
1.3.2 <i>Цілі та задачі розробки</i>	<i>13</i>
Висновок до розділу	14
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....	15
2.1 ВХІДНІ ДАНІ	15
2.2 ВИХІДНІ ДАНІ	18
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ.....	19
Висновок до розділу	19
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	20
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ.....	20
3.2 МАТЕМАТИЧНА ПОСТАНОВКА	21
3.3 ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ’ЯЗАННЯ	22
3.4 ОПИС МЕТОДІВ РОЗВ’ЯЗАННЯ	24
3.4.1 <i>VLEU</i>	<i>24</i>
3.4.2 <i>Косинус подібності</i>	<i>26</i>
3.4.3 <i>Інші застосовані алгоритми</i>	<i>28</i>
Висновок до розділу	28
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	29
4.1 ЗАСОБИ РОЗРОБКИ	29
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	32
4.3 АРХІТЕКТУРА ПЗ.....	32
4.3.1 <i>Діаграма класів</i>	<i>33</i>
4.3.2 <i>Діаграма послідовності</i>	<i>35</i>

4.3.3	Діаграма компонентів.....	36
4.3.4	Специфікація функцій	36
	Висновок до розділу	42
5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ.....	43
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА.....	43
5.2	ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	52
5.2.1	Мета випробування	52
5.2.2	Загальні положення.....	53
5.2.3	Результати випробування	53
	Висновок до розділу	58
	ЗАГАЛЬНІ ВИСНОВКИ	59
	ПЕРЕЛІК ПОСИЛАНЬ	60
	ДОДАТОК А	63
	ДОДАТОК Б	73

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

БД – база даних.

СКБД – система керування базами даних.

ООП – об'єктно-орієнтоване програмування.

MVC – архітектурний шаблон проектування систем.

BLEU – алгоритм оцінки відповідності текстових уривків.

EDM – модель даних з використанням сутностей.

TF – Term Frequency, частість, з якою слово зустрічається у документі.

IDF – Inverse Document Frequency, величина, що характеризує рідкість слова серед набору документів.

AJAX – підхід, що дозволяє обмінюватись даними серверу та клієнту без необхідності останнього завантажувати сторінку заново.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Перевірка рівня знань є невід’ємною частиною сучасного освітнього процесу. Одним з найбільш популярних її видів є тестування. Більшість видів питань у тестах є «закритими», тобто мають жорстко задану кінцеву множину можливих відповідей. «Відкриті» ж питання, де відповідь дається у вигляді фрази, речення чи невеликого тексту, дозволяють оцінити ступінь знань учня більш комплексно. Втім, головний їх недолік – необхідність людської перевірки.

Отже, на сьогоднішній день організація автоматичної перевірки відповідей на відкриті питання є актуальною проблемою. У зв’язку з цим у даній роботі пропонується розглянути систему для аналізу відповідей природною мовою.

Мета дослідження: збільшити об’єктивність оцінювання в електронних тестових системах за рахунок розробки системи семантичного аналізу відповідей природною мовою.

Об’єкт дослідження: системи тестування.

Предмет дослідження: методи аналізу відповідей відкритого типу на питання тестів природною мовою.

Призначення програмного продукту - інформаційно-аналітичне забезпечення навчального процесу:

- безпосередньо під час проведення тестування;
- під час процесу оцінки наданих учнями відповідей;
- під час управління тестуванням та доповнення тестових матеріалів.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

Забезпечення зазначених процесів дозволить спростити і пришвидшити проведення контролю знань.

У даній роботі проведено розбір предметної області, задач та цілей розроблюваного програмного продукту. У відповідності до предметної області побудовано структуру даних. Розглянуто алгоритми, що використовуються для вирішення поставлених задач, обґрунтовано їх вибір та викладено принципи роботи. Спроектовано структуру програмного продукту, описано засоби розробки та обґрунтовано їх вибір. З їх використанням та у відповідності із спроектованою структурою розроблено програмний продукт; перевірено його функціональність, створено керівництво користувача.

Практичне значення отриманих результатів: розроблено веб-застосунок для організації, проведення тестування та його автоматичної перевірки.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Сьогодні тестування у сфері освіти розглядається як одна з актуальних форм контролю якості підготовки учнів та студентів, що дозволяє об'єктивно та швидко оцінити об'єм засвоєної навчальної дисципліни. Використання тестових завдань різних видів дозволяє відповідати вимогам державного освітнього стандарту. Крім того, тестування є особливо корисним в якості інструменту дистанційного контролю знань чи поточної (оперативної) перевірки повноти засвоєння студентами теми.

Тестування має наступні переваги над очним опитуванням:

- тестування є більш об'єктивним за рахунок виключення суб'єктивного фактору та стандартизації процедури перевірки;
- тести дозволяють оцінити знання рівномірно за всім пройденим курсом, в той час як особисте опитування зазвичай зачіпає дві-чотири теми;
- тестування є ефективним з точки зору витрат часу: основні витрати – на розробку якісного тесту - мають разовий характер.

Втім, тестування має і недоліки:

- відповідь дається «механічно», а візуальна пам'ять стає вирішальною при виконанні тестів;
- тести не можуть бути джерелом навчання під час перевірки, вони лише індикатор недостатності знань;
- цей тип контролю якості освіти має вразливість до вгадування, особливо у тестах невисокої складності. [1]

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

Існує багато видів завдань закритого типу, в яких до кожного питання необхідно обрати правильну відповідь з кількох запропонованих варіантів. Також широко використовуються й завдання відкритого типу, в яких тестований студент має запропонувати свою власну відповідь - слово, фразу, формулу і т.д. Вони (за класифікацією Аванесова) є одними з найбільш розповсюджених поруч з завданнями на відповідність, одиночним чи множинним вибором [2].

Відкриті питання позбавлені багатьох недоліків питань закритого типу; у той же час вони втрачають і одну з головних переваг – простоту й швидкість перевірки. Комп'ютер не здатний «усвідомити» зміст тексту, а відповідь може бути сформульована по-різному. Тому проблема розробки засобів семантичного аналізу відповідей є актуальною.

1.1.1 Опис процесу діяльності

Опишемо процес діяльності з проведення контролю якості освіти до впровадження автоматизованої системи тестування та після.

Без автоматизованої системи викладач за матеріалами тестів підготовлює витратні матеріали (роздруківки). Далі вони роздаються студентам для виконання тесту. Після виконання тесту результати збираються і вручну перевіряються викладачем. Оцінки заносяться у відповідний документ і доводяться до студентів у тій чи іншій формі.

Із застосуванням системи викладач повідомляє студентам про необхідність проходження певного тесту у системі. Студенти, пройшовши тест у браузері, надсилають результати до програмної системи, яка перевіряє їх відповіді і зберігає результати у БД. Після цього студентам одразу відображуються результати. За необхідності викладач має доступ до усіх результатів тесту.

Відповідні схеми діяльності знаходяться у графічних матеріалах.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

1.1.2 Опис функціональної моделі

Опишемо функціональну модель системи.

Актор - «Студент».

Основні функції системи:

- можливість переглянути список модулів;
- можливість пройти будь-який з існуючих тестів;
- можливість отримання звіту про пройдений тест;
- можливість отримання переліку звітів за ім'ям;

Актор - «Викладач».

Основні функції системи:

- можливість додати, змінити чи видалити модуль;
- можливість переглянути список питань модуля;
- можливість переглянути список тестів модуля;
- можливість додати, змінити чи видалити запитання;
- можливість додати, змінити чи видалити тест;
- можливість переглянути список питань тесту;
- можливість переглянути всі спроби проходження тесту.

Відповідну структурну схему вимог до функцій системи (UseCase діаграму) включено до графічних матеріалів.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

1.2 Огляд наявних аналогів

На даний момент більшість систем тестування передбачає у своєму складі так звані «закриті» питання – у цьому випадку варіанти відповідей передбачені заздалегідь та строго задані. Це можуть бути одиночні відповіді, декілька тверджень чи співставлення. Відповіді на «відкриті» питання у Moodle поділяються на два типи. «Есе» перевіряються викладачем вручну [3], а «короткі відповіді» перевіряються на точне співпадіння з одним із варіантів, заздалегідь заданих текстовим шаблоном варіантів [4]. Залучення викладача до перевірки зменшує переваги автоматизованості тестів, а шаблонний підхід не є гнучким відносно відповіді і також є дуже чутливим до способу її формулювання. Аналогів у сфері контролю якості навчання, робота яких засновується на семантичному аналізі, не спостерігається.

У даному продукті, на відміну від Moodle, передбачені лише питання «відкритого» типу і представлено менше функцій щодо їх управління, але аналіз відповідей на відкриті питання буде проводитися більш ґрунтовно, із застосуванням засобів семантичного аналізу.

Порівняння з аналогами представлено у таблиці 1.1.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Таблиця 1.1 – Аналіз аналогів

Функціональність	Організація тестування вручну	Moodle	Продукт, що створюється
Простота організації тестування	-	+	+
Широкі можливості з управління процесом контролю якості освіти	-	+	-
Питання закритого типу	+	+	-
Питання відкритого типу	+	+	+
Грунтовна перевірка питань відкритого типу засобами семантичного аналізу	-	-	+
Зберігання результатів та зручний пошук	-	+	+

В продукті, що розробляється, на відміну від інших існуючих тестових систем, є автоматична перевірка питань відкритого типу. Продукт не містить таких широких можливостей управління, як Moodle, та не має можливості включення до тестів закритих питань. Проте, виходячи з поставлених завдань та цілей, можемо зробити висновок, що така функціональність є надлишковою – отже, необхідність її реалізації відсутня.

1.3 Постановка задачі

Є множина тем, так званих «модулів», управління якими веде викладач. До кожного модуля приписаний набір питань з відповідної теми та набір тестів. Тест може містити питання свого модуля. Питання вводяться у систему разом з оціненими прикладами відповідей, модулі – разом із тематичними текстами-словниками. При проходженні студентом тесту його відповіді на питання оцінюються на основі збережених

прикладів відповідей та тематичних текстів, після чого результати оцінювання зберігаються у системі та відображаються студенту.

1.3.1 Призначення розробки

Призначенням розробки є організація можливості автоматизованої перевірки відповідей на відкриті питання з послідуною оцінкою достовірності відповіді.

1.3.2 Цілі та задачі розробки

Цілі розробки програмного продукту:

- надати зручний сервіс для викладачів з управління тестуванням;
- скоротити час, що витрачається на обробку результатів, шляхом введення автоматичної перевірки відповідей;
- забезпечити зручний сервіс для проходження тестування для студентів.

Для досягнення поставлених цілей необхідно вирішити такі задачі:

- забезпечення інтерфейсу управління тестами та їх складовими для викладача;
- здійснення аналізу наданих відповідей з наступним визначенням ступеню релевантності;
- забезпечення можливості проходження тесту з наступним збереженням його результатів та можливістю їх перегляду.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Висновок до розділу

Проаналізовано застосування тестового підходу до оцінювання рівня знань у сучасному освітньому процесі, обґрунтовано актуальність створюваного програмного продукту.

Розглянуто характерні риси наявних підходів, порівняно з системою, що розроблюється та зроблено висновки щодо їхніх можливостей у оцінюванні.

Визначено призначення розроблюваного ПП, його цілі та задачі, які необхідно розв'язати для досягнення поставлених цілей.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Оскільки на поточний момент можливості програмного забезпечення не передбачають розуміння і осмислення людської мови, для того, щоб оцінити певну відповідь, програмі необхідні вже оцінені викладачем приклади. Після знаходження співвідношення між різними показниками вже оцінених відповідей на їх основі буде можливою і оцінка нової, наданої студентом.

Оцінюється кожна відповідь, виходячи з ступеню її відповідності тематичному тексту, текстам на іншу тему та вже оціненим прикладам відповідей. Отже, ці дані й необхідні для роботи системи.

Оскільки розробляється клієнт-серверний веб-застосунок, усі дані мають зберігатися на сервері у базі даних.

Необхідні дані:

– Дані про модуль:

- 1) номер
- 2) назва;
- 3) тематичний текст – словник;

– Дані про тест:

- 1) номер;
- 2) назва;
- 3) кількість запитань, що виводиться користувачу;
- 4) модуль (посилання)
- 5) список питань (посилання)

- Дані про запитання:
 - 1) номер;
 - 2) текст запитання;
 - 3) модуль (посилання)
 - 4) ваги (для аналізу відповідей)
- Дані про приклад відповіді:
 - 1) номер;
 - 2) текст;
 - 3) оцінка;
 - 4) зразковість (чи є зразком для порівняння);
 - 5) питання (посилання);
- Дані про результат тесту:
 - 1) номер;
 - 2) час виконання;
 - 3) ім'я виконавця;
 - 4) тест (посилання)
- Дані про користувацьку відповідь:
 - 1) номер
 - 2) текст
 - 3) оцінка
 - 4) питання (посилання)
 - 5) результат тесту, до якого відноситься (посилання)

Структура сутностей бази даних у вигляді ER-діаграми наведена на рисунку 2.1.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

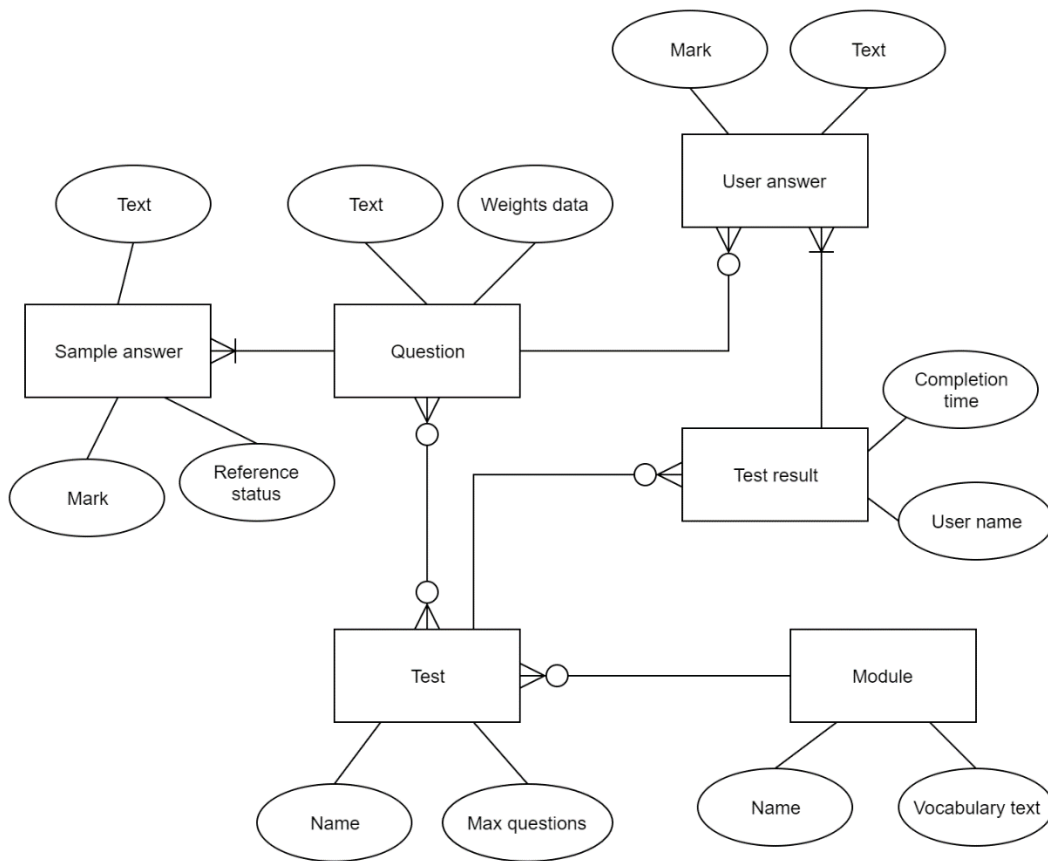


Рисунок 2.1 – Схема структурна ER-діаграми бази даних

Така організація бази даних є доцільною, оскільки забезпечує необхідний рівень нормалізації і дозволяє досить просто виконувати необхідні операції (додавання, зміна, видалення).

Оскільки веб-інтерфейс містить функціонал зміни та доповнення бази даних, до інформаційної системи від нього надходять дані; вони являють з себе фактично цілісні сутності, що представлені на діаграмі. Така поведінка забезпечується застосуванням EDM-концепції. [5]

2.2 Вихідні дані

Оскільки програмний продукт є веб-застосунком, то для всіх його операцій необхідна інформація з бази даних. В залежності від потреб з інформаційної системи запитуються різні набори даних у вигляді сутностей, які ці набори даних описують; структура цих сутностей є відповідною структурі таблиць бази даних. Таку поведінку забезпечує використання підходу EDM. [5]

Набір типів вхідних даних представлено у таблиці 2.1.

Таблиця 2.1 – Типи вихідних даних

Назва	Опис
Question	Дані питання
AnswerSample	Дані прикладу відповіді
UserAnswer	Дані відповіді користувача
TestResult	Дані результатів тесту
Test	Дані тесту
Module	Дані модуля

Вихідних документів система не має.

2.3 Опис структури бази даних

В базу даних мають бути збережені дані, що описують:

- модулі;
- питання;
- тести;
- приклади відповідей на питання;
- результати тестування;
- користувацькі відповіді.

В цілому структура БД відповідає структурі вхідних даних.

Відповідна структурна схема бази даних інформаційної системи знаходиться у графічних матеріалах.

Висновок до розділу

Визначено перелік даних, що подаються на вхід та вихід інформаційної підсистеми програмного продукту. Створено загальну схему сутностей, інформація про які буде міститися у базі даних.

Побудовано структуру бази даних відповідно до викладеного у попередніх розділах, яка є найбільш підходящою з огляду на розроблюваний програмний продукт.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

У сфері контролю якості освіти на сьогоднішній день тести є одним з основоположних інструментів. Широкого поширення набуло автоматизоване тестування. Автоматизація процесу тестування спрощує роботу викладачів і пришвидшує перевірку знань.

Розроблюваний програмний продукт має оцінювати відповіді студента, засновуючись на збережених прикладах відповідей та на тематичних текстах модулів .

Для реалізації поставлених цілей необхідно зберігати інформацію, пов'язану з тестами та їх результатами, проводити автоматичну перевірку результатів тестування та забезпечити зручний доступ до функціоналу через інтерфейс. До математичної частини відноситься перевірка відповідей студентів.

Метою є створення алгоритму визначення достовірності відповіді на основі оцінених прикладів відповідей та відомостей за темою питання.

На математичне забезпечення покладається:

- порівняння відповіді з термінологією теми питання;
- порівняння відповіді зі зразковими правильними відповідями;
- виведення оцінки на основі даних, отриманих за допомогою вищезазначених підходів.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

3.2 Математична постановка

Надана користувачем відповідь повинна являти собою фразу невеликої довжини (2 – 5 слів) англійською мовою.

Оцінка достовірності відповіді має являти собою дійсне число у діапазоні від 0 до 1.

Для отримання оцінки повинна бути оцінені достовірність відповіді відносно теми питання та відносно зразкових відповідей.

Оцінка достовірності відносно теми має проводитися з урахуванням частоти використання слів, даних у відповіді, серед текстів.

Оцінка достовірності відносно зразкових відповідей має проводитися з урахуванням частоти використання слів і словосполучень, подібності зразковим відповідям.

Загальна оцінка має виводитися на основі попередніх оцінок із урахуванням усіх оцінених відповідей, наданих викладачем в якості зразка. Ступені значущості факторів, згідно з якими виводиться остаточна оцінка, невідомі і мають бути визначені на основі оцінок відповідей, наданих викладачем.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

3.3 Обґрунтування методу розв'язання

Оцінка наданої відповіді відносно зразкових відповідей є основною складовою аналізу. Комп'ютерна програма не здатна розуміти людську мову – це є AI-повною задачею [6], тобто на поточний момент не існує технологій, здатних забезпечити розуміння природньої мови. Самостійно визначити правильність наданої відповіді може тільки людина; комп'ютер може лише оцінити близькість відповідей до правильних, що завчасно надані і оцінені людьми.

Для порівняння наданих та зразкових відповідей було обрано так званий алгоритм BLEU. Алгоритм обрано через його відносну простоту, високу швидкість, досить високий степінь кореляції оцінки цього алгоритму та людської оцінки [7, 8]. Він застосовується для оцінки близькості наданої фрази чи тексту до наданих завчасно, найчастіше – для перевірки правильності машинного перекладу у порівнянні з професіональним людським. Загальна ідея алгоритму полягає у тому, що, чим ближче запропонований варіант до зразків – тим вище оцінка [7].

Оцінка відповіді за частістю використання слів у темі потрібна для виключення можливості перебору варіантів та для забезпечення реакції системи на використання не передбачених зразками відповідей слів. В якості метрики для такої оцінки було обрано косинус подібності. Ця методика обрана, бо широко використовується у пошукових системах і показала високий рівень точності передбачення відповідності документа запиту [9]. У даному випадку у ролі запита буде виступати відповідь, а у ролі документів – сторонні тексти та текст, пов'язаний із темою питання.

Застосовано аналіз за представленими частинами мови: використання нетипових для відповіді слів буде розцінене як відхилення і відповідно відобразиться на остаточній оцінці. Цей підхід застосований

тому, що він запобігатиме розпізнаванню однокореневих слів як однакових.

Для забезпечення стійкості алгоритму до одиничних помилок у написанні слів застосовано просте порівняння тексту; у разі, якщо відмінності від однієї відповіді зі зразків мінімальні, оцінка не буде сильно зниженою.

Для забезпечення стійкості алгоритму до сприйняття відповіді, що є протилежною за змістом, враховується також кількість конструкцій, що позначають заперечення у англійській мові, таким чином змінюючи зміст фрази на протилежний. Такий підхід є можливим завдяки тому, що англійська мова у більшості випадків є добре структурованою, а побудова конструкцій заперечення у ній відповідає чітким правилам.

У постановці задачі зазначено, що співвідношення, яке пов'язує вищевказані підходи, методи і алгоритми, невідоме. Для успішного їх застосування та остаточної оцінки відповіді співвідношення має бути виведене; ця задача покладається на нейромережу прямого поширення (feed-forward) – нейромережі є типовим рішенням для задач цього типу. Цей інструмент обрано тому, що він чудово себе показує у задачах, де необхідно знайти співвідношення, за яким один набір факторів перетворюється у інший. Доведено, що мережа прямого поширення з одним прихованим шаром може наблизити будь-яку неперервну функцію багатьох змінних з будь-якою точністю [10]. Отже, за прикладами вже оцінених відповідей нейромережа «навчиться» оцінювати й нові.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

3.4 Опис методів розв'язання

3.4.1 BLEU

BLEU [7] – алгоритм для вимірювання якості тексту у порівнянні зі зразками, в основному застосовується у сфері машинного перекладу, звідки і пішла його назва. Ставши одним з перших алгоритмів, чия оцінка корелює з людською, він є досить популярним через простоту і низьку «вартість» обчислень.

Оцінка алгоритму завжди знаходиться між 0 та 1. Ця величина відображає, наскільки текст, що перевіряється, є близьким до зразків, чим ближче значення до 1 – тим ближче текст до зразків. Зі збільшенням кількості зразків оцінка зазвичай зростає.

Метрика BLEU заснована на підрахуванні кількості, у якій зустрічається у зразках кожна з n-грам тексту, що перевіряється. N-грама – послідовність з n елементів, у даному випадку – слів. Зверху величина, що відображує кількість входжень у текст, що перевіряється, обмежується максимальною кількістю входжень у зразках:

$$m_{ngram} = \max(m_{ngram}^{text}, m_{ngram}^{references}). \quad (3.1)$$

Після цього ці величини підсумовуються для всіх n-грам, для отримання оцінки сума ділиться на їх загальну кількість у тексті і множиться на так званий штраф короткості (brevity penalty):

$$P = \frac{brevity\ penalty}{ngram\ count\ in\ text} * \sum_{all\ ngrams\ in\ text} [m_{ngram}]. \quad (3.2)$$

Якщо довжина тексту, що перевіряється, більше довжини найменшого зі зразків то коефіцієнт штрафу короткості дорівнює одиниці.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

У протилежному випадку він визначається за формулою:

$$e^{1-\frac{r}{c}}, \quad (3.3)$$

де c – довжина у словах тексту, що перевіряється, а r – кількість слів у найменшому зі зразків. Приклад роботи алгоритму представлений на рисунку 3.1.

Cand. 1	the	the	the	the	the	the	the
Cand. 2	apple	is	on	a	table		
Ref. 1	there	is	an	apple	on	the	table
Ref. 2	the	apple	is	on	the	table	

<p>Candidate 1, n = 1:</p> <p>7 x "the"</p> <p>$P = \min(7, 2) / 7 = 0.285;$</p>	<p>Candidate 2, n = 1:</p> <p>"apple", "is", "on", "a", "table"</p> <p>$P = e^{(1-6/5)} * (1 + 1 + 1 + 0 + 1) / 5 = 0.655$</p>
<p>Candidate 1, n = 2:</p> <p>6 x "the the"</p> <p>$P = \min(6, 0) / 7 = 0;$</p>	<p>Candidate 2, n = 2:</p> <p>"apple is", "is on", "on a", "a table"</p> <p>$P = e^{(1-6/5)} * (1 + 1 + 0 + 0) / 4 = 0.409$</p>

Рисунок 3.1 – Приклад роботи алгоритму BLEU

Найкраще алгоритм корелює з людською оцінкою при $n = 4$ [8]; втім, недовгі відповіді, що складаються з двох-трьох слів, будуть відкинуті за такого способу оцінювання, тож для даної задачі застосовним є n не більше двох.

Також алгоритм було вдосконалено: за допомогою фреймворку WordNet [11] порівняння n -грам відбувається з урахуванням синонімів слів, що входять до їх складу – тож оцінка людської відповіді буде більш

точною у разі, якщо студент дасть відповідь з використанням синонімів слів, що входять до зразків відповідей.

3.4.2 Косинус подібності

Косинус подібності (cosine similarity) застосовується до текстів наступним чином: якщо запит і тема є взаємовідповідними, то у тексті з даної теми слова, що містяться у запиті, зустрічаються разом частіше, ніж у інших текстах. Будучи початково застосованим в якості основи пошукових систем, цей інструмент підходить для вирішення такої задачі, як встановлення відносного ступеня відповідності певної теми фразі запиту.

Оцінка відповідності документа запиту міститься на відрізку між 0 та 1 і відображає, наскільки корелюють між собою частоти застосування слів фрази-запиту у документі. Визначається вона у два етапи.

Спочатку визначаються частоти, з якими зустрічаються слова у межах кожного документа (Term Frequency, TF):

$$TF(\text{word}, \text{document}) = \frac{\text{word count in document}}{\text{words amount in document}}. \quad (3.4)$$

Після цього обраховується зворотна частість слів запиту відносно документів (Inverse Document Frequency, IDF) – вона визначає ступінь «рідкості» певного слова серед документів. Визначається вона наступним чином:

$$IDF(\text{word}) = 1 + \ln\left(\frac{N_{doc}}{N_{word}}\right), \quad (3.5)$$

де N_{doc} – загальна кількість документів, N_{word} – кількість документів, що містить шукане слово.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

Далі всі документи і запит ставляться у відповідність векторам, кількість вимірів яких дорівнює кількості різних слів запиту, а величини у векторі в кожному з них рівні добутку IDF цього слова та TF у документі (чи запиті).

Після переведення текстів у вектори визначається той вектор, що є найближчим за напрямком до вектору запиту, для цього визначається косинус кута між ними, що також зветься косинусом подібності:

$$\text{CosSim}(v1, v2) = \frac{v1 \cdot v2}{\|v1\| \cdot \|v2\|} \quad (3.6)$$

Ця величина і відображає відповідність документа і запиту (відносно інших документів, які були залучені до розрахунку IDF). Приклад роботи алгоритму представлений на рисунку 3.2.

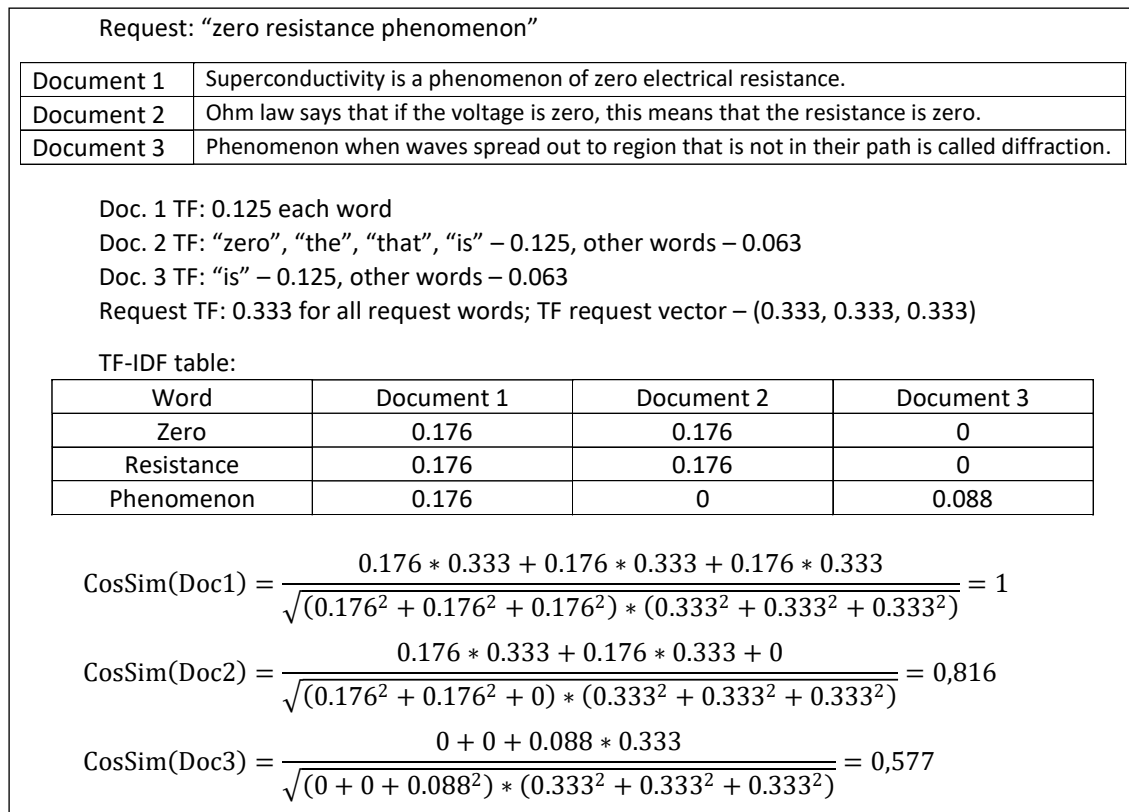


Рисунок 3.2 – Приклад роботи алгоритму, заснованого на косинусі подібності

3.4.3 Інші застосовані алгоритми

Як вже зазначено, також застосовано наступні алгоритми та засоби:

- порівняння тексту як послідовності символів; порівнюються надана відповідь та зразки правильних відповідей. Для виявлення фрагментів обрано широко розповсюджений алгоритм diff, принцип роботи якого засновано на пошуку найдовших спільних підпослідовностей [12]. Приклад роботи однієї з реалізацій алгоритму представлений на рисунку 3.3;
- порівняння частоти, з якою зустрічаються різні частини мови. Для цього застосовано такий інструмент семантичного аналізу, як таггер частин мови (POS-tagger) фреймворку OpenNLP [13];
- підрахунок кількості використаних мовних структур, що позначають заперечення (таких, як «non-» чи «dis-»), у відповіді;
- нейронна мережа прямого поширення з одним прихованим шаром для виведення остаточної оцінки з проміжних оцінок, наданих використовуваними алгоритмами.

Рисунок 3.3 – Приклад роботи однієї з реалізацій алгоритму diff

Висновок до розділу

Таким чином, у даному розділі було визначено суть завдання, його математичну постановку. Відповідно до завдання обрано алгоритми, що забезпечують виконання поставленого завдання, обґрунтовано їх вибір та викладено основні принципи їх роботи.

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

В якості мови програмування серверної частини додатку було обрано С# з використанням технології MVC ASP.NET.

С# [14] - об'єктно-орієнтована мультипарадигменна мова програмування загального призначення, розроблена у 1998-2001 році як основна мова платформи Microsoft .NET Framework, відноситься до сім'ї С-подібних мов.

Основними архітектурними характеристиками є:

- строга статична типізація, безпека типів, єдина система типів;
- потужна система обробки виключень;
- автоматичне керування пам'яттю, «збірка сміття»;
- спрямованість мови на ООП та компонентно-орієнтоване програмування;
- можливість простої інтеграції з компонентами, написаними іншими мовами та з застосуванням інших технологій.

.NET загалом і, зокрема, С#, можуть використовуватися на всіх сучасних операційних системах, що вказує на кросплатформеність, хоч в основному використання відбувається під управлінням ОС Windows в силу історичних причин та інтеграції платформи .NET у цю операційну систему.

MVC ASP.NET [15] – технологія, що надає простий і гнучкий патерноорієнтований спосіб розробки динамічних веб-додатків, засновуючись на відомому патерні MVC. Ця технологія інтегрована у платформу .NET і в основному застосовується сумісними з нею мовами. Схематичне зображення концепції патерну представлено на рисунку 4.1.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

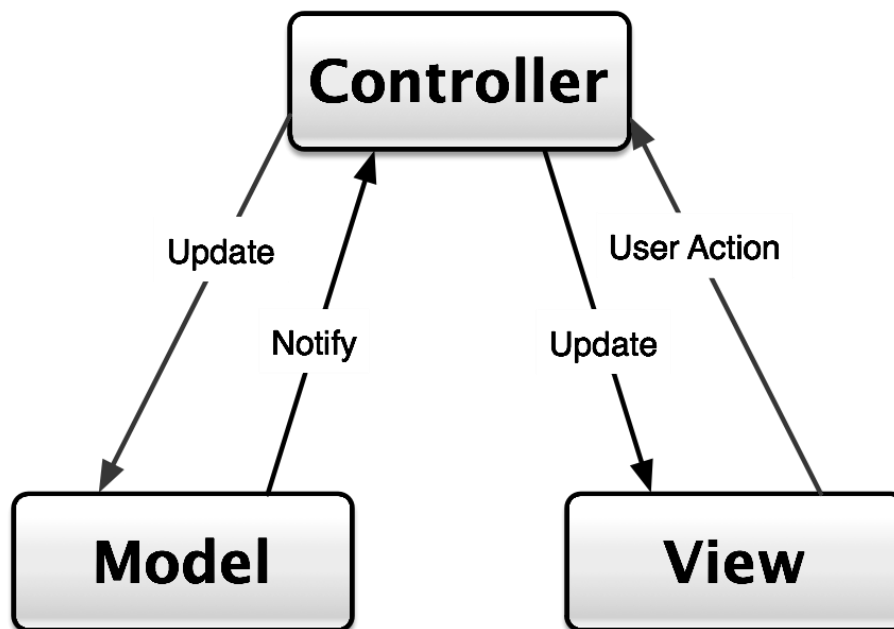


Рисунок 4.1 – Схематичне зображення концепції патерну MVC

В якості СКБД було обрано SQL Server Express, а в якості інструменту організації доступу до даних - Entity Framework.

SQL Server [16] – реляційна СКБД, основна мова запитів якої – Transact-SQL; розроблена спільно компаніями Microsoft та Sybase. Має багато сучасних засобів роботи з даними, є інтегрованою у платформу .NET, має високі показники продуктивності, успішно конкурує на ринку СКБД багато років.

Entity Framework [17] – об’єктно-орієнтована технологія доступу до даних, представляє з себе ORM-рішення для платформи .NET від Microsoft. Цей засіб дозволяє створити «безшовний» перехід між базою даних та об’єктами, з якими працює розроблюване програмне забезпечення, прискорити і полегшити розробку ПЗ, забезпечити належний рівень надійності і безпеки при взаємодії БД та програмної системи.

У користувацькому інтерфейсі також задіяні HTML, CSS, Bootstrap, JavaScript.

Змн.	Арк.	№ докум.	Підпис	Дата

HTML [18] – стандартна мова розмітки гіпертекстових документів, що застосовується у абсолютній більшості веб-сторінок, є зручною у використанні і простою для розуміння.

CSS [19] – використовувана разом з HTML мова, призначення якої – опис зовнішнього виду, стилю сторінок, написаних мовою розмітки.

Bootstrap [20] – набір інструментів (фреймворк), призначений для створення веб-додатків та сайтів; містить у собі різноманітні елементи управління та навігації, а також розширення JavaScript.

JavaScript [21] – інтерпретована динамічна мова програмування. Зазвичай є мовою сценаріїв веб-сторінок, у такому разі вона інтерпретується браузером.

В якості середовища розробки було використано Visual Studio [22], оскільки воно напряму пов'язане із середовищем .NET і є основним засобом для розробки застосунків мовою C#. За допомогою Visual Studio можна розробляти програмне забезпечення майже будь-якого типу, від консольних та віконних застосунків до веб-сайтів та мобільних додатків.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

4.2 Вимоги до технічного забезпечення

Для забезпечення стабільної роботи системи технічні засоби мають містити наступні складові:

- сервер з характеристиками:
 - 1) процесор з частотою не менше 1.5 ГГц;
 - 2) об'єм ОЗП – 4 Гб;
 - 3) об'єм вільного місця на диску – 1 Гб;
 - 4) інші параметри та складові мають другорядне значення, оскільки не впливають на роботу програми значним чином.
- ПЗ, встановлене на сервері:
 - 1) ОС Windows 7 / Windows Server 2008 чи вище;
 - 2) Internet Information Services як компонент Windows;
 - 3) .NET Framework 4.5 або вище;
 - 4) SQL Server Express 2008 або вище.

Клієнтський пристрій має бути підключений до тієї ж мережі. На ньому має бути встановлений сучасний веб-браузер з підтримкою HTML 5 та AJAX. Інші параметри та програмне забезпечення клієнтського пристрою особливої ролі не грають.

4.3 Архітектура ПЗ

У даному програмному продукті застосована клієнт-серверна архітектура.

В розділі описана структура реалізації програмного забезпечення. Проектування передбачає розділення функцій програми між підсистемами, кожна з яких відповідає за виконання певної частини завдання.

Підсистема обробки відповідей влаштована за принципом фасаду: внутрішній функціонал поєднано так, що звернення до системи відбувається через єдиний клас, що полегшує використання модуля.

Інформаційна підсистема, що відповідає за взаємодію з базою даних, побудована за патерном репозиторіїв: класи, що відповідають використовуваним об'єктам, містять лише дані, у той час як функціонал взаємодії з БД винесено до так званих «репозиторіїв». Така організація підсистеми полегшує розробку, повторне використання коду і тестування.

Архітектура підсистеми, що надає користувачем веб-інтерфейс, засновується на архітектурному шаблоні Model-View-Controller, що передбачає поділ системи на три частини – дані (модель), інтерфейс (вигляд), модуль керування (контролер). Такий шаблон забезпечує гнучкість програмного забезпечення, дозволяє повторно використовувати деякі компоненти, його використання сприяє зрозумілості й впорядкованості.

4.3.1 Діаграма класів

Структурна схема класів програмного забезпечення включена до графічних матеріалів.

На діаграмі класів зазначені наступні класи:

- “Bleu” відповідає за розрахунок BLEU-оцінки близькості наданої відповіді та зразків;
- “NGram” представляє собою використовувані у алгоритмі BLEU так звані n-грами – послідовності зі слів;
- “Cosine” проводить розрахунок відповідності відповіді відомим темам, використовуючи концепцію косинуса подібності;
- “StringAnalysis” включає у себе усі інші застосовувані підходи – подібність текстів за складом, подібність текстів за

використовуваними частинами мови та підрахунок заперечень у виразі;

- “FeedForwardNetwork” представляє собою нейронну мережу прямого поширення, що об’єднує оцінки від різних алгоритмів у загальну;
- “AnalysisFacade” представляє з себе «фасад» модуля, що дозволяє користуватись усім функціоналом підсистеми, зібраним разом;
- “Module” являє собою представлення теми тестів та питань;
- “Vocabulary” є словником цієї теми, між темою та словником завжди існує взаємно однозначна відповідність;
- “Question” представляє питання з певної теми;
- “Weights” містить збережені коефіцієнти для оцінки питання, між набором вагових коефіцієнтів та питанням завжди існує взаємно однозначна відповідність;
- “AnswerSample” являє собою приклад оціненої викладачем відповіді;
- “Test” представляє з себе збірку питань з певної теми;
- “TestResult” є записом про проходження тесту студентом;
- “UserAnswer” містить запис про відповідь студента на питання під час проходження тесту;
- “OAdbContext” огортає підключення до бази даних;
- “DbSetProvider” реалізує інтерфейс “IDbSetProvider”, огортає контекст бази даних, регулює доступ до нього і час його існування;
- “ModuleRepository” інкапсулює усю взаємодію з базою даних, що стосується об’єктів типу «модуль»;
- “QuestionRepository” інкапсулює усю взаємодію з базою даних, що стосується об’єктів типу «запитання»;

- “TestRepository” інкапсулює усю взаємодію з базою даних, що стосується об’єктів типу «тест»;
- “TestResultRepository” інкапсулює усю взаємодію з базою даних, що стосується об’єктів типу «результат тесту»;
- “ModulesController” містить у собі всю логіку інтерфейсу, що стосується модулів;
- “QuestionsController” містить у собі всю логіку інтерфейсу, що стосується запитань;
- “TestsController” містить у собі всю логіку інтерфейсу, що стосується тестів;
- “ResultsController” містить у собі всю логіку інтерфейсу, що стосується перегляду результатів тестів;
- “ModuleModel” включає дані, що стосуються модуля;
- “QuestionModel” містить дані, що стосуються питання;
- “TestModel” включає дані, що стосуються тесту;
- “TestResultModel” являє собою контейнер для даних, що стосуються результату проходження тесту;
- “UserAnswerModel” містить відповідь студента на питання під час проходження тесту.

4.3.2 Діаграма послідовності

Розглядається сценарій проходження тесту студентом. Відповідна структурна схема послідовності сценарію включена до графічних матеріалів.

Користувач, переходячи на сторінку проходження тесту, відправляє запит до веб-інтерфейсу. Після завантаження з інформаційної підсистеми інформації про тест вона огортається веб-сторінкою та повертається користувачеві.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

Користувач, відповівши на питання, відправляє відповіді на сервер. Веб-інтерфейс співставляє надані відповіді з питаннями і подає запит у інформаційну підсистему на збереження результатів спроби. Відповіді передаються далі у підсистему аналізу, звідки повертаються оцінки користувацьких відповідей. Вони зберігаються разом з відповідями та іменем користувача, після чого ідентифікатор збереженої спроби повертається до веб-інтерфейсу, який перенаправляє користувача на сторінку перегляду результатів щойно завершеної спроби.

4.3.3 Діаграма компонентів

Відповідна структурна схема компонентів включена до графічних матеріалів.

Програмний продукт має три основних структурних компоненти: обробки відповідей, взаємодії із базою даних та представлення інформації користувачеві.

“WordNet”, “DiffLib”, “OpenNLP”, “StemmersNet” [23] – засоби семантичного аналізу, що застосовуються програмним продуктом.

“AnswerAnalysis” – підсистема надання оцінок відповідям.

“DataLayer” – підсистема взаємодії з базою даних.

“OATests Web Interface” – підсистема веб-інтерфейсу.

4.3.4 Специфікація функцій

Клас “Bleu” містить методи і дані, що використовуються для оцінювання відповіді за допомогою BLEU-алгоритму. Функції класу представлені у таблиці 4.1.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

Таблиця 4.1 – Опис функцій класу Bleu

Назва функції	Призначення функції
CalculateBleu	Підраховує BLEU-оцінки для наданої відповіді за передбаченими значеннями N
BrevityPenalty	Підраховує «штраф за короткість»
ClipCount	Підраховує кількість спільних входжень n-грам у зразках та у відповіді
ClosestWordCount	Знаходить довжину зразка, найближчого до відповіді за кількістю слів
CountNGram	Підраховує оцінку для певного значення N без врахування штрафу короткості

Клас “Cosine” містить методи і дані, що використовуються для оцінювання відповіді за допомогою косинусу подібності. Функції класу представлені у таблиці 4.2.

Таблиця 4.2 – Опис функцій класу Cosine

Назва функції	Призначення функції
Compute	Підраховує ступінь відповідності наданої фрази з усіма проіндексованими документами

Клас “StringAnalysis” містить методи, що доповнюють основні алгоритми визначення оцінки. Функції класу представлені у таблиці 4.3.

Таблиця 4.3 – Опис функцій класу StringAnalysis

Назва функції	Призначення функції
MaxPosSimilarity	Знаходить ступінь подібності відповіді найближчому зразку за кількістю використаних частин мови
MaxSimilarity	Знаходить ступінь подібності відповіді найближчому зразку за написанням
NegationCount	Знаходить у наданому виразі кількість структур, що позначають заперечення

Клас “FeedForwardNetwork” представляє нейронну мережу прямого поширення. Функції класу представлені у таблиці 4.4.

Таблиця 4.4 – Опис функцій класу FeedForwardNetwork

Назва функції	Призначення функції
BackPropTrain	Навчає мережу на певній вибірці із заданою швидкістю навчання та максимальною кількістю епох
ComputeResult	Отримує значення на виходах мережі із вхідних значеннях
InitWeights	Ініціалізує ваги мережі невеликими значеннями
Serialize	Переводить дані мережі у масив байтів
Deserialize	Відновлює мережу з масиву байтів

Клас “NGram” містить методи і дані, що представляють n-граму. Функції класу представлені у таблиці 4.5.

Таблиця 4.5 – Опис функцій класу NGram

Назва функції	Призначення функції
Equals	Визначає рівність n-грам з урахуванням синонімів

Клас “ModuleRepository” містить методи, що забезпечують взаємодію з базою даних, яка стосується об’єктів типу «модуль». Функції класу представлені у таблиці 4.6.

Таблиця 4.6 – Опис функцій класу ModuleRepository

Назва функції	Призначення функції
Add	Додає новий модуль
ById	Повертає модуль за його числовим ідентифікатором
Delete	Видаляє модуль
Edit	Змінює дані модуля
GetAllModules	Повертає список усіх модулів

Клас “QuestionRepository” містить методи, що забезпечують взаємодію з базою даних, яка стосується об’єктів типу «питання». Функції класу представлені у таблиці 4.7.

Таблиця 4.7 – Опис функцій класу QuestionRepository

Назва функції	Призначення функції
Create	Додає питання і обчислює його вагові коефіцієнти
ByModule	Повертає список питань, що входять до модуля
ById	Повертає питання за його числовим ідентифікатором
Edit	Змінює дані питання, за потреби перераховує ваги
NotInTest	Повертає список питань, що не входять у тест

Клас “TestResultRepository” містить методи, що забезпечують взаємодію з базою даних, яка стосується об’єктів типу «результат тесту». Функції класу представлені у таблиці 4.8.

Таблиця 4.8 – Опис функцій класу TestResultRepository

Назва функції	Призначення функції
Save	Зберігає результат виконання і всі відповіді у ньому
ByNames	Повертає список результатів за іменем виконавця та назвою тесту
ById	Повертає результат тесту за ідентифікатором

Клас “TestRepository” містить методи, що забезпечують взаємодію з базою даних, яка стосується об’єктів типу «тест». Функції класу представлені у таблиці 4.9.

Таблиця 4.9 – Опис функцій класу TestRepository

Назва функції	Призначення функції
Add	Додає новий тест
ById	Повертає тест за його ідентифікатором
ByModule	Повертає список тестів, що входять до модуля
Delete	Видаляє тест
Edit	Змінює дані тесту
AddQuestion	Додає питання до тесту
RemoveQuestion	Видаляє питання з тесту

Клас “ModulesController” містить методи, що забезпечують роботу частини інтерфейсу, яка відповідає за дії з модулями. Функції класу представлені у таблиці 4.10.

Таблиця 4.10 – Опис функцій класу ModulesController

Назва функції	Призначення функції
Index	Повертає сторінку, що містить список модулів
Details	Повертає сторінку, що містить інформацію про модуль та дозволяє його редагувати
Create	Повертає сторінку, що дозволяє створити новий модуль
Delete	Повертає сторінку, що дозволяє видалити модуль

Клас “QuestionsController” містить методи, що забезпечують роботу частини інтерфейсу, яка відповідає за дії з питаннями. Функції класу представлені у таблиці 4.11.

Таблиця 4.11 – Опис функцій класу QuestionsController

Назва функції	Призначення функції
ByModule	Повертає сторінку, що містить список питань модуля
Details	Повертає сторінку, що містить інформацію про питання та дозволяє її редагувати
Create	Повертає сторінку, що дозволяє створити нове питання
Delete	Повертає сторінку, що дозволяє видалити питання
AnswersFromString	Отримує приклади відповідей з рядка
AsSingleString	Поєднує приклади відповідей у рядок

Клас “ResultsController” містить методи, що забезпечують роботу частини інтерфейсу, яка відповідає за дії з результатами тестів. Функції класу представлені у таблиці 4.12.

Таблиця 4.12 – Опис функцій класу ResultsController

Назва функції	Призначення функції
ById	Повертає сторінку, що містить детальну інформацію про спробу
Search	Повертає сторінку, що дозволяє знайти спробу за іменем виконавця та за назвою тесту
Filtered	Повертає частину сторінки, яка містить результати пошуку

Клас “TestsController” містить методи, що забезпечують роботу частини інтерфейсу, яка відповідає за дії з тестами. Функції класу представлені у таблиці 4.13.

Таблиця 4.13 – Опис функцій класу TestsController

Назва функції	Призначення функції
ByModule	Повертає сторінку, що містить список тестів модуля
Create	Повертає сторінку, що дозволяє створити новий тест
Details	Повертає сторінку, що містить інформацію про тест та дозволяє її редагувати
Delete	Повертає сторінку, що дозволяє видалити тест
Exercise	Повертає сторінку, що дозволяє пройти тест та надіслати на перевірку
Questions	Повертає сторінку, що дозволяє змінити список питань тесту
AddQuestion	Додає питання до тесту та повертає частину сторінки, що дозволяє видалити питання з тесту
RemoveQuestion	Видаляє питання з тесту та повертає частину сторінки, що дозволяє додати питання до тесту

Висновок до розділу

У розділі описано засоби, що застосовані для розробки програмного продукту та обґрунтовано їх вибір. Приведено вимоги до технічного забезпечення. Приведено інформацію щодо архітектури розробленого програмного забезпечення та описано діаграми, що її пояснюють. Приведено детальний перелік функцій класів, які складають систему. Таким чином, маємо опис програмного продукту, що виконує поставлені задачі.

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Креслення усіх описаних у керівництві користувача екранних форм включені до графічних матеріалів.

Сайт має змінювану викладачем множину тем, так званих «модулів». Модуль характеризується ім'ям та текстом на тему цього модуля. Кожен модуль може містити у собі набір питань та набір тестів згідно теми, що представляється цим модулем. Тест може містити питання зі свого модуля. Питання вводяться у систему викладачем з оціненими прикладами відповідей на них.

При проходженні студентом тесту його відповіді на питання оцінюються на основі збережених прикладів відповідей та тематичних текстів, після чого результати оцінювання зберігаються у системі та відображаються студенту.

При відкритті кореневої сторінки сайту в браузері користувача відображається список модулів. Користувачу пропонується переглянути деталі існуючих модулів, створити новий модуль чи переглянути список результатів зроблених спроб виконання тестів (ця опція доступна на кожній сторінці). Сторінка списку модулів представлена на рисунку 5.1.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

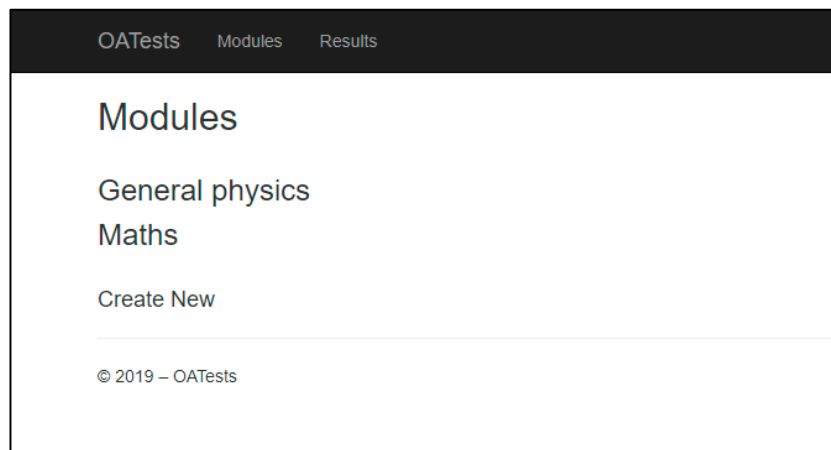


Рисунок 5.1 – Сторінка зі список модулів

При переході на сторінку «Create new» відкривається форма створення нового модуля. На цій сторінці можна ввести дані нового модуля і додати його до списку чи скасувати створення модуля і повернутися до списку модулів. Сторінка створення модуля представлена на рисунку 5.2.

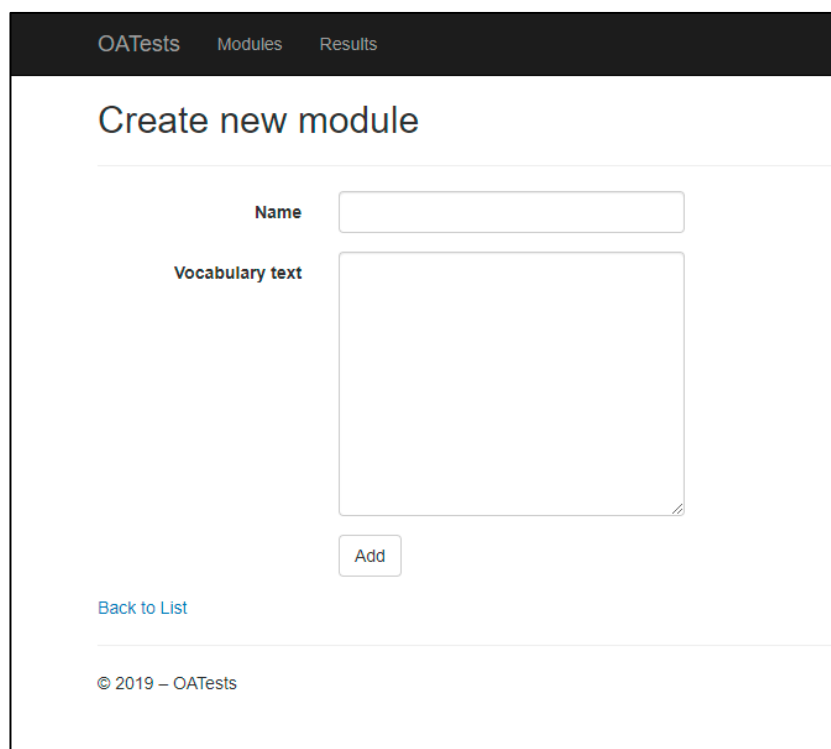


Рисунок 5.2 – Сторінка створення модуля

При поверненні до списку та переході до сторінки деталей вже створеного модуля відкривається сторінка управління модулем. З неї можна змінити дані модуля, перейти до списку питань модуля, списку тестів модуля, повернутись до списку модулів, видалити модуль. Сторінка управління модулем представлена на рисунку 5.3.

The screenshot shows a web interface for managing modules. At the top, there are navigation tabs: 'OATests', 'Modules', and 'Results'. The main heading is 'Details of module "Electromagnetics"'. Below this, there is a form with two main sections: 'Name' and 'Vocabulary text'. The 'Name' field contains the text 'Electromagnetics'. The 'Vocabulary text' field contains a paragraph of text: 'Superconductivity is a phenomenon of exactly zero electrical resistance and expulsion of magnetic flux fields occurring in certain materials, called superconductors, when cooled below a characteristic critical temperature. The simplest method to measure the electrical resistance of a sample of some material is to place it in an electrical circuit in series with a'. Below the form is a 'Save' button. At the bottom of the page, there are several links: 'Question list', 'Tests list', 'Delete', and 'Back to List'. The footer of the page reads '© 2019 – OATests'.

Рисунок 5.3 – Сторінка управління модулем

При переході до списку питань модуля можемо переглянути детально кожне з них, додати нове питання чи повернутися до модуля. Сторінка списку питань модуля представлена на рисунку 5.4.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

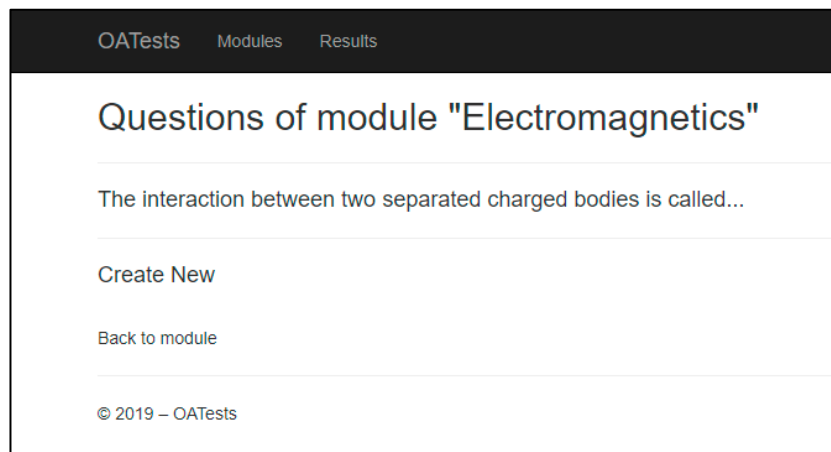


Рисунок 5.4 – Сторінка списку питань модуля

При переході на сторінку «Create new» відкривається форма створення нового запитання у обраному модулі. На цій сторінці можна ввести дані нового питання і додати його до списку питань модуля чи скасувати створення і повернутися до списку питань модуля. Сторінка створення питання представлена на рисунку 5.5.

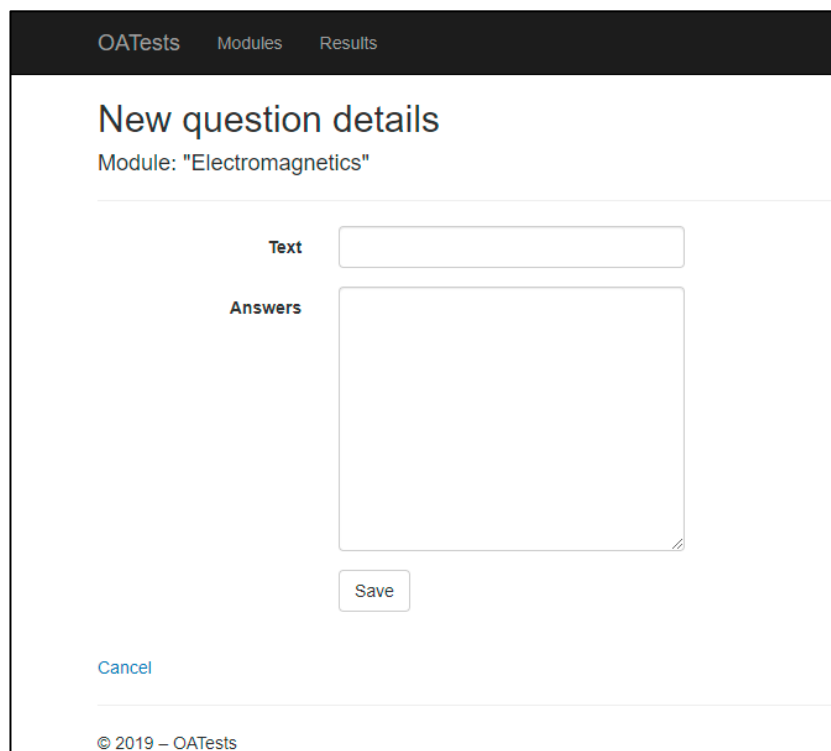


Рисунок 5.5 – Сторінка створення питання

Змн.	Арк.	№ докум.	Підпис	Дата

Приклади відповідей повинні вводитися у наступному форматі: текст відповіді має міститися у подвійних лапках. Якщо відповідь є зразковою, після лапок має йти знак оклику; у іншому разі навпроти неї має стояти оцінка від 0 до 1, розділяти цілу та дробову частину має крапка. Якщо приклад представляє невірну відповідь, оцінка якої – нуль, то оцінку можна не вказувати.

Після створення питання чи після повернення до списку питань модуля при переході до сторінки деталей питання відкривається сторінка управління питанням. З неї можна змінити дані питання, видалити його чи повернутись до списку питань модуля. При зміні даних питання приклади відповідей мають бути введені у тому ж форматі, що й при створенні питання. Сторінка управління питанням представлена на рисунку 5.6.

OATests Modules Results

Question details

Module: "Electromagnetics"

Text

Answers

- zero resistance" !
- "phenomenon of zero resistance" !
- "no resistance" !
- "no resist" !
- "zero electric resistance" 1,000
- "zero electricity resistance" 0,800
- "exectli zero resistance" 0,600
- "zro electric resist" 0,300
- "exactli zeroo rsistance" 0,300
- "resistance of"
- "exactly zero"

[Delete](#)

[Back to module question list](#)

© 2019 – OATests

Рисунок 5.6 – Сторінка управління питанням

Змн.	Арк.	№ докум.	Підпис	Дата

При поверненні до деталей модуля та переході до списку тестів відкривається сторінка, що дозволяє переглянути деталі вже створеного тесту чи створити новий. Сторінка списку тестів показана на рисунку 5.7.

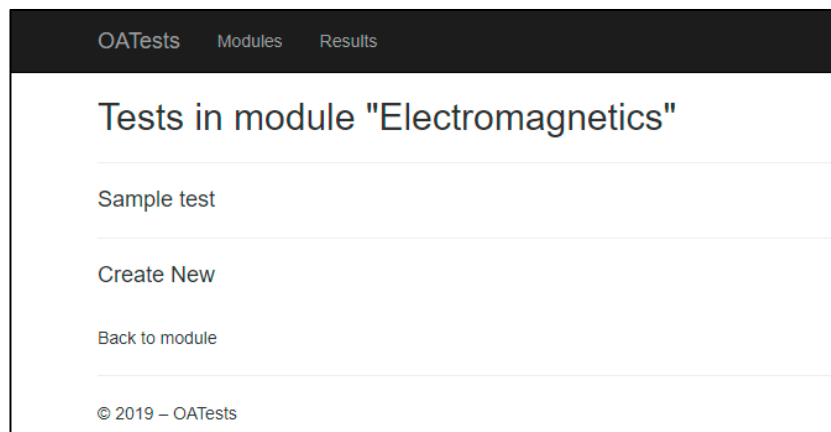


Рисунок 5.7 – Сторінка списку тестів модуля

При переході на сторінку «Create new» відкривається форма створення нового тесту у обраному модулі. Тест має назву та обмеження на максимальну кількість питань, що відображатиметься користувачеві. Якщо можливих питань в тесті буде більше, то потрібна кількість буде вибрана з загального набору випадковим чином. Сторінка створення тесту представлена на рисунку 5.8

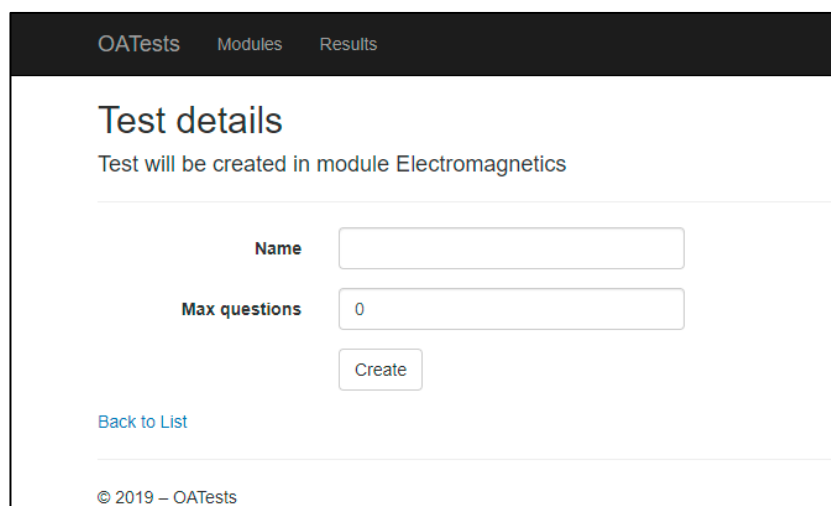


Рисунок 5.8 – Сторінка створення тесту

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

Після створення тесту чи після повернення до списку тестів модуля при переході до сторінки деталей тесту відкривається сторінка управління тестом. З неї можна змінити дані тесту, розпочати його, видалити його, перейти до переглядання результатів спроб цього тесту, перейти до управління списком питань тесту чи повернутися до списку тестів. Сторінка управління тестом представлена на рисунку 5.9.

Рисунок 5.9 – Сторінка управління тестом

При переході на сторінку управління списком питань тесту відображуються всі питання модуля, надається можливість додати їх до тесту чи видалити з нього натисненням відповідних кнопок. Сторінка управління списком питань представлена на рисунку 5.10.

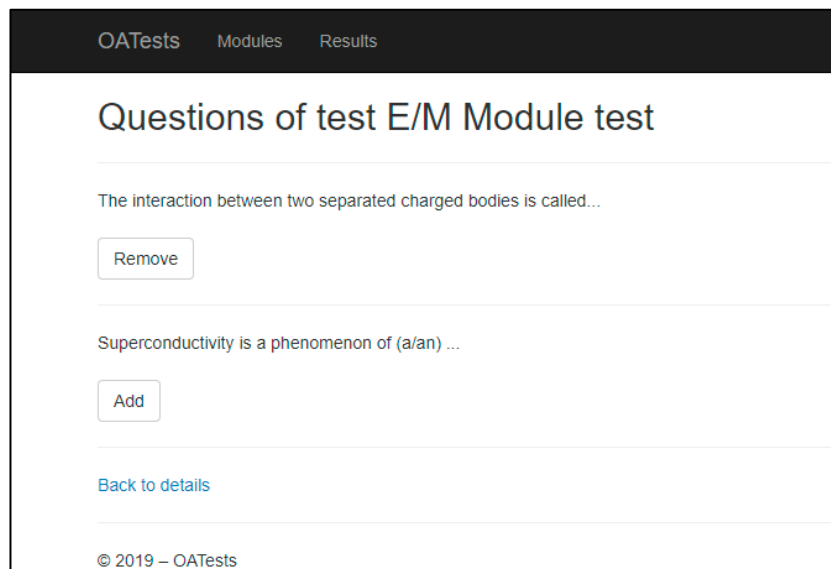


Рисунок 5.10 – Сторінка управління списом питань

При поверненні до деталей тесту та переході до його виконання відкривається сторінка, що містить питання тесту, поле імені виконавця, поля відповідей та дозволяє після надання відповідей відправити тест на перевірку. Сторінка проходження тесту представлена на рисунку 5.11.

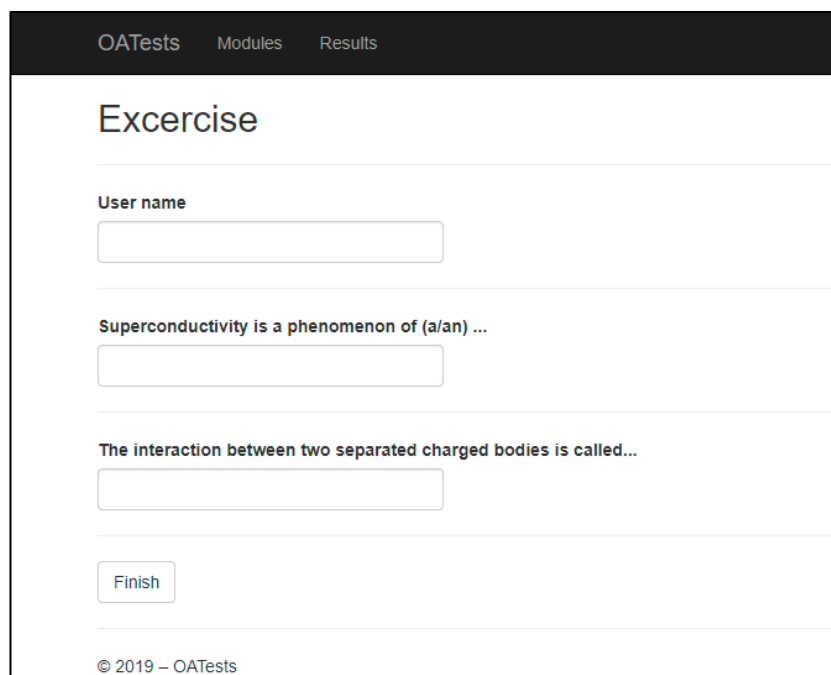


Рисунок 5.11 – Сторінка проходження тесту

Після завершення тесту відображується сторінка з результатами, з якої можна повернутися до сторінки тесту. Сторінка результатів представлена на рисунку 5.12.

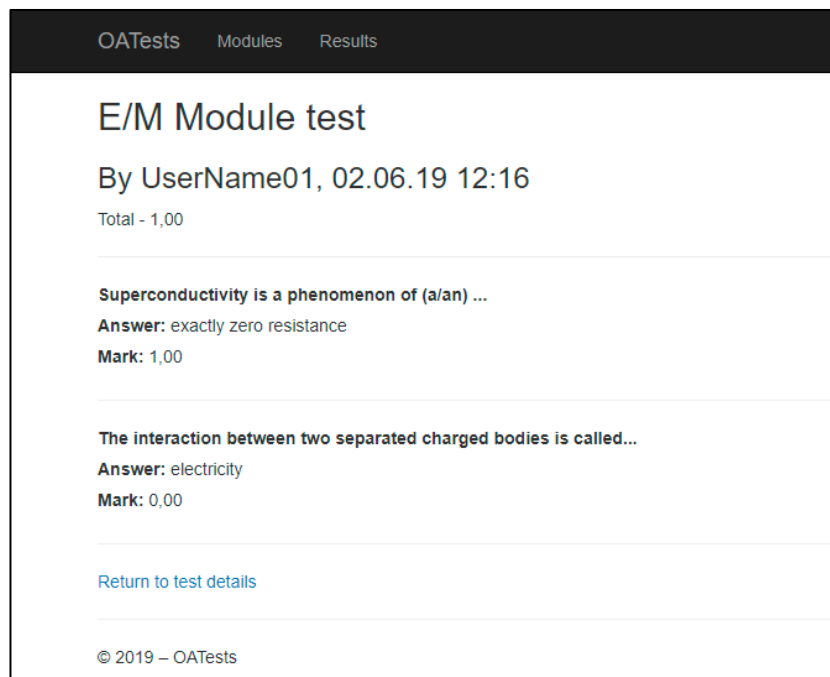


Рисунок 5.12 – Сторінка проходження тесту

Результати виконаних тестів також можна знайти, перейшовши за посиланням “Results” на верхній панелі. Сторінка пошуку результатів представлена на рисунку 5.13.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

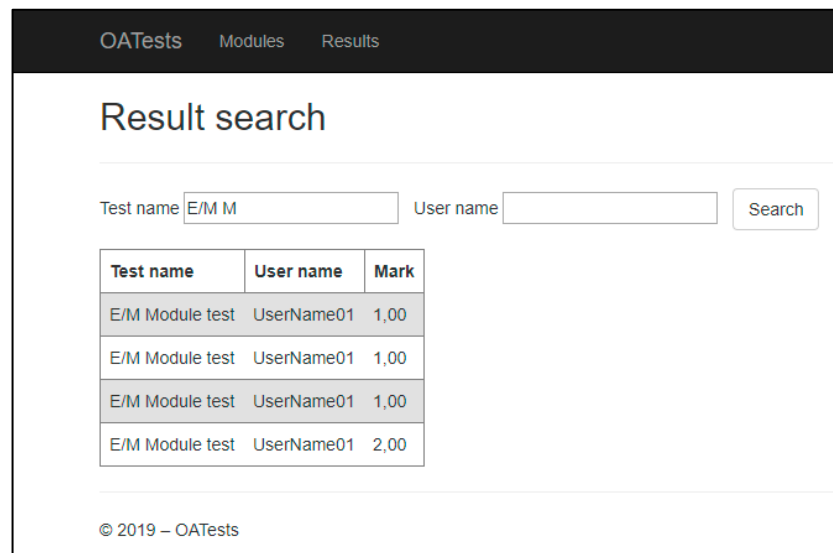


Рисунок 5.13 – Сторінка пошуку результатів тестів

Таким чином, розроблений програмний продукт успішно розв’язав наступні поставлені у дипломній роботі задачі:

- забезпечення інтерфейсу управління тестами та їх складовими для викладача;
- здійснення аналізу наданих відповідей з послідуочим визначенням ступеню їх відповідності;
- забезпечення можливості проходження тесту з послідуочим переглядом результатів.

5.2 Випробування програмного продукту

5.2.1 Мета випробування

Метою випробування є перевірка відповідності створеного програмного продукту вказаним у технічному завданні вимогам та перевірка його працездатності.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи. Вимоги до змісту документів.

5.2.3 Результати випробування

Процедура проведення випробувань та її результати представлені у таблицях 5.1 – 5.5.

Таблиця 5.1 – Результати тестування управління модулем

Дія:	Очікуваний результат:	Результат тесту:
Відкрийте сайт	Сайт відкритий та доступний; відкритий список модулів	Співпадає з очікуваним результатом
Натисніть на посилання “Create new”	Відкрита сторінка для створення модуля. Форма для введення його даних відкрита та доступна	Співпадає з очікуваним результатом
Заповніть поля модуля та натисніть кнопку “Add”	Відкритий список модулів, він містить щойно додане питання	Співпадає з очікуваним результатом
Натисніть назву модуля	Відкрита сторінка з інформацією про модуль	Співпадає з очікуваним результатом
Змініть назву модуля та його текст, натисніть “Save”	Відкритий список модулів, назва відредагованого модуля змінилася на введену	Співпадає з очікуваним результатом

Продовження таблиці 5.1

Дія:	Очікуваний результат:	Результат тесту:
Натисніть назву модуля	Відкрита сторінка з оновленою інформацією про модуль	Співпадає з очікуваним результатом
Натисніть посилання "Delete"	Відкрита сторінка підтвердження видалення модуля	Співпадає з очікуваним результатом
Натисніть кнопку "Delete"	Відкрита сторінка списку модулів, видаленого модуля в ньому немає	Співпадає з очікуваним результатом

Таблиця 5.2 – Результати тестування управління питанням

Дія:	Очікуваний результат:	Результат тесту:
Відкрийте сайт, натисніть на назву модуля, перейдіть за посиланням "Question list"	Сайт відкритий та доступний; відкрита сторінка з інформацією про модуль; відкрита сторінка зі списком питань модуля	Співпадає з очікуваним результатом
Натисніть посилання "Create new"	Відкрита сторінка створення питання	Співпадає з очікуваним результатом
Заповніть форму створення питання та натисніть "Save"	Відкрита сторінка списку питань модуля, в ньому з'явилося нове питання	Співпадає з очікуваним результатом
Натисніть текст питання	Відкрита сторінка з інформацією про питання	Співпадає з очікуваним результатом

Продовження таблиці 5.2

Дія:	Очікуваний результат:	Результат тесту:
Змініть текст питання та його відповіді, натисніть "Save"	Відкритий список питань, текст відредагованого питання змінився на введений	Співпадає з очікуваним результатом
Натисніть текст питання	Відкрита сторінка з оновленою інформацією про питання	Співпадає з очікуваним результатом
Натисніть посилання "Delete"	Відкрита сторінка підтвердження видалення питання	Співпадає з очікуваним результатом
Натисніть кнопку "Delete"	Відкрита сторінка списку питань модуля, видаленого питання немає	Співпадає з очікуваним результатом

Таблиця 5.3 – Результати тестування управління тестом

Дія:	Очікуваний результат:	Результат тесту:
Відкрийте сайт, натисніть на назву модуля, перейдіть за посиланням "Tests list"	Сайт відкритий та доступний; відкрита сторінка управління модулем; відкрита сторінка зі списком тестів модуля	Співпадає з очікуваним результатом
Натисніть на посилання "Create new"	Відкрита сторінка для створення тесту. Форма для введення його даних відкрита та доступна	Співпадає з очікуваним результатом
Заповніть форму створення тесту та натисніть "Save"	Відкрита сторінка списку тестів модуля, в ньому з'явився новий тест	Співпадає з очікуваним результатом

Продовження таблиці 5.3

Дія:	Очікуваний результат:	Результат тесту:
Натисніть назву тесту	Відкрита сторінка з інформацією про тест	Співпадає з очікуваним результатом
Змініть назву тесту та його максимальну кількість питань, натисніть “Save”	Відкритий список тестів, назва відредагованого тесту змінилась на введену	Співпадає з очікуваним результатом
Натисніть назву тесту	Відкрита сторінка з оновленою інформацією про тест	Співпадає з очікуваним результатом
Натисніть посилання “Delete”	Відкрита сторінка підтвердження видалення тесту	Співпадає з очікуваним результатом
Натисніть кнопку “Delete”	Відкрита сторінка списку тестів модуля, видаленого тесту немає	Співпадає з очікуваним результатом

Таблиця 5.4 – Результати тестування управління питаннями тесту

Дія:	Очікуваний результат:	Результат тесту:
Відкрийте сайт, натисніть на назву модуля, перейдіть за посиланням “Tests list”; Натисніть назву тесту	Сайт відкритий та доступний; відкритий список модулів; відкрита сторінка з інформацією про тест	Співпадає з очікуваним результатом
Натисніть на посилання “Manage questions”	Відкривається список питань модуля. Біля тих, що включені у тест, знаходиться кнопка “Remove”. Біля тих, що не	Співпадає з очікуваним результатом

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 5.4

Дія:	Очікуваний результат:	Результат тесту:
	включені у тест, знаходиться кнопка “Add”	
Видаліть питання, що є у тесті, відповідною йому кнопкою “Remove”	Натиснення кнопки “Remove” замінює її на “Add”	Співпадає з очікуваним результатом
Додайте питання у тест відповідною йому кнопкою “Add”	Натиснення кнопки “Add” замінює її на “Remove”	Співпадає з очікуваним результатом

Таблиця 5.5 – Результати тестування проходження тестування та пошуку результатів

Дія:	Очікуваний результат:	Результат тесту:
Відкрийте сайт, натисніть на назву модуля, перейдіть за посиланням “Tests list”; Натисніть назву тесту	Сайт відкритий та доступний; відкритий список модулів; відкрита сторінка з інформацією про тест	Співпадає з очікуваним результатом
Натисніть на посилання “Start”	Відкривається сторінка з формою виконання тесту з питаннями, полями для введення їх відповідей, полем для імені і кнопкою “Finish”	Співпадає з очікуваним результатом
Заповніть усі поля згідно їх змісту та натисніть кнопку “Finish”	Відкривається сторінка з результатами тесту	Співпадає з очікуваним результатом

Продовження таблиці 5.5

Натисніть на посилання “Results”	Відкривається сторінка пошуку результатів тестів	Співпадає з очікуваним результатом
Введіть початок імені, під яким проходився тест, та/чи початок назви тесту і натисніть “Search”	Серед виведених результатів є запис щойно завершеної спроби виконання тесту	Співпадає з очікуваним результатом

Висновок до розділу

Таким чином, у розділ було включено керівництво користувача, що описує, як користуватися програмним продуктом. Зазначено мету проведення випробувань, керівні документи для нього. Випробування проведено успішно. Отже, розроблений програмний продукт у повній мірі є працездатним і розв’язує поставлені задачі.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

ЗАГАЛЬНІ ВИСНОВКИ

В даній роботі було описано призначення розробки системи тестування на основі питань відкритого типу та актуальність цього програмного продукту в сфері контролю якості освіти, визначено цілі розробки та задачі для їх розв'язання. Розглянуто особливості існуючих підходів до вирішення задач. Визначено перелік даних програмного продукту, що подаються на вхід та вихід та побудовано структуру бази даних. Описано використовувані алгоритми та дано обґрунтування їх використанню. Описано застосовувані засоби розробки, обґрунтовано їх застосування; доведено структуру проекту, його архітектуру. Створено керівництво користувача, успішно проведене тестування.

Виходячи з цього, можна зробити наступні висновки.

Автоматичний аналіз питань відкритого типу природною мовою є актуальним і дає змогу підвищення ефективності електронних тестових систем.

Застосований алгоритм BLEU при обмеженні проблемної області питання і кількості слів у відповіді має переваги при аналізі відповідей перед іншими алгоритмами, що спрощують як розробку інформаційного забезпечення системи для програміста, так і наступне його використання для викладача.

Створений програмний продукт у повній мірі задовольняє висунутим до нього вимогам і розв'язує поставлені перед ним задачі. Втім, в подальшому є можливим удосконалення системи. Може бути переглянута та розширена список застосовуваних алгоритмів, проведене подальше їх вдосконалення відповідно до предметного середовища, вдосконалення способу виведення кінцевої оцінки.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

ПЕРЕЛІК ПОСИЛАНЬ

1. Попов А.В. – Тестирование как метод контроля качества знаний студентов // Труды Санкт-Петербургского государственного институт культуры, 2013.
2. Аванесов В. С. Форма тестовых заданий. — Москва: Центр тестирования, 2006.
3. Essay question type - Moodle Docs [Электронный ресурс] // Режим доступа: https://docs.moodle.org/37/en/Essay_question_type
Дата доступа: 25.04.2019
4. Short-Answer question type – Moodle Docs [Электронный ресурс] // Режим доступа:
https://docs.moodle.org/37/en/Short-Answer_question_type
Дата доступа: 16.12.2018.
5. Entity Data Model | Microsoft Docs [Электронный ресурс] // Режим доступа:
<https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/entity-data-model>
Дата доступа: 30.03.2017.
6. Stuart C. Shapiro – Artificial Intelligence // Encyclopedia of Artificial Intelligence, Second Edition – New York: John Wiley & Sons Inc., 1992.
7. Papineni, K. та ін. – BLEU: a method for automatic evaluation of machine translation // Proceedings of the 40th Annual meeting of the ACL – Philadelphia, 2002
8. Coughlin, D. – Correlating Automated and Human Assessments of Machine Translation Quality // MT Summit IX – New Orleans, USA, 2003

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

9. Felipe Bravo-Marquez та ін. – A Text Similarity Meta-Search Engine Based on Document Fingerprints and Search Results Records // 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology – IEEE, Lyon, France, 2011
10. Cybenko, G.V. – Approximation by Superpositions of a Sigmoidal function // Mathematics of Control, Signals and Systems, vol. 2 - Springer-Verlag New York Inc., 1989.
11. WordNet | A Lexical Database for English [Електронний ресурс] // Режим доступу: <https://wordnet.princeton.edu/>
Дата доступу: 25.05.2019
12. E. Myers – An O(ND) Difference Algorithm and Its Variations // Algorithmica. 1 (2) – Department of Computer Science, University of Arizona, 1986
13. AlexPoint / OpenNlp - Open source NLP tools [Електронний ресурс] // Режим доступу: <https://github.com/AlexPoint/OpenNlp>
Дата доступу: 22.05.2019
14. C# Guide | Microsoft Docs [Електронний ресурс] // Режим доступу: <https://docs.microsoft.com/en-us/dotnet/csharp/>
Дата доступу: 30.01.2018
15. ASP.NET MVC Pattern | .NET [Електронний ресурс] // Режим доступу: <https://dotnet.microsoft.com/apps/aspnet/mvc>
Дата доступу: 20.05.2019
16. Microsoft Data Platform | Microsoft [Електронний ресурс] // Режим доступу: <https://www.microsoft.com/en-us/sql-server/default.aspx>
Дата доступу: 21.05.2019

17. Entity Framework | Microsoft Docs [Електронний ресурс] // Режим доступу:
<https://docs.microsoft.com/en-us/ef/>
 Дата доступу: 18.05.2019
18. W3C HTML [Електронний ресурс] // Режим доступу:
<https://www.w3.org/html/>
 Дата доступу: 28.05.2019
19. Cascading Style Sheets [Електронний ресурс] // Режим доступу:
<https://www.w3.org/Style/CSS/Overview.en.html>
 Дата доступу: 26.05.2019
20. Bootstrap [Електронний ресурс] // Режим доступу:
<https://getbootstrap.com/>
 Дата доступу: 24.05.2019
21. Javascript | MDN [Електронний ресурс] // Режим доступу:
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
 Дата доступу: 21.05.2019
22. Visual Studio IDE [Електронний ресурс] // Режим доступу:
<https://visualstudio.microsoft.com>
 Дата доступу: 17.05.2019
23. StemmersNet – CodePlex Archive [Електронний ресурс] // Режим доступу:
<https://archive.codeplex.com/?p=stemmersnet>
 Дата доступу: 29.04.2019

ДОДАТОК А

Тексти програмного коду

Система тестування "Open Answers Tests"

(найменування програми (документа))

DVD-R

(вид носія даних)

9 арк, 98.3 Кб

(Обсяг програми (документа) , арк., Кб)

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64


```

        for (int j = 0; j < references.Count; j++)
        {
            refWords[j] = references[j].ToWords();
            refWordsCounts[j] = refWords[j].Length;
        }

        for (int n = 1; n <= nMax; n++)
        {
            Dictionary<NGram, int>[] refNGramCounts = new Dictionary<NGram,
int>[references.Count];

            for (int j = 0; j < references.Count; j++)
            {
                int lim = refWords[j].Length - n + 1;
                Dictionary<NGram, int> nGramDict =
                    Enumerable.Range(0, lim).
                        Select(i => new NGram(refWords[j].Skip(i).Take(n).ToArray())).
                        GroupBy(i => i).ToDictionary(g => g.Key, g => g.Count());
                refNGramCounts[j] = nGramDict;
            }
            refNGramCountsByN[n - 1] = refNGramCounts;
        }
    }

    private readonly int maxGramSize;
    private readonly Dictionary<NGram, int>[][] refNGramCountsByN;
    private readonly int[] refWordsCounts;

    private double CountNGram(int n, string candidate, bool calculateBP, out double
brevityPenalty)
    {
        Dictionary<NGram, int>[] refNGramCounts = refNGramCountsByN[n - 1];
        string[] candWords = candidate.ToWords();
        int candNGramCount = candWords.Length - n + 1;
        Dictionary<NGram, int> candDict =
            Enumerable.Range(0, candNGramCount).
                Select(i => new NGram(candWords.Skip(i).Take(n).ToArray())).
                GroupBy(i => i).ToDictionary(g => g.Key, g => g.Count());
        int clippedCount = ClipCount(candDict, refNGramCounts);
        if (calculateBP)
        {
            int r = ClosestWordCount(refWordsCounts, candWords.Length);
            brevityPenalty = BrevityPenalty(candWords.Length, r);
        }
        else
        {
            brevityPenalty = 0;
        }
        return ((double)clippedCount) / candNGramCount;
    }

    private static double BrevityPenalty(int c, int r)
    {
        return (c < r) ? Math.Exp(1.0 - ((double)r) / c) : 1.0;
    }

    private static int ClosestWordCount(int[] refWordCounts, int candWordCount)
    {
        int leastDiff = int.MaxValue;
        int best = 0;
    }

```



```

        public static int IndexOfMax<T>(this IEnumerable<T> source) where T :
        IComparable<T>
        {
            if (source == null)
            {
                throw new ArgumentNullException(nameof(source));
            }
            IEnumerator<T> e = source.GetEnumerator();
            if (!e.MoveNext())
            {
                throw new ArgumentException("Sequence is empty");
            }
            int i = 0, j = 0;
            T max = e.Current;
            while (e.MoveNext())
            {
                j++;
                T c = e.Current;
                if (c.CompareTo(max) > 0)
                {
                    i = j;
                    max = c;
                }
            }
            return i;
        }

        private const string delimiter = ",. -\t\\(\|)\|[\\]";
        private const string removed = "[\d'\"\"'`~]";
        public static readonly IStemmer Stemmer = new EnglishStemmer();

        public static string[] ToStems(this string str)
        {
            return ToWords(str).Select(w => Stemmer.Stem(w)).ToArray();
        }
        public static string[] ToWords(this string str)
        {
            return Regex.Replace(str.Trim().ToLower(), removed, string.Empty).
                Split(delimiter.ToCharArray(),
                    StringSplitOptions.RemoveEmptyEntries);
        }
    }
}

```

AnswerAnalysis/NGram.cs

```

using Syn.WordNet;
using System;
using System.IO;
using System.Linq;

namespace AnswerAnalysis
{
    Internal class NGram : IEquatable<NGram>
    {
        public override int GetHashCode()
        {
            return 0;
        }
    }
}

```

```

public override bool Equals(object obj)
{
    if (obj is NGram nGram)
    {
        return Equals(nGram);
    }
    else
    {
        return false;
    }
}

public bool Equals(NGram other)
{
    if (Length != other.Length)
    {
        return false;
    }
    for (int i = 0; i < Length; i++)
    {
        string word = words[i];
        string bWord = other.words[i];
        if (word == bWord ||
            wordNet.GetSynSets(word).
                SelectMany(s => s.Words).
                SelectMany(w => w.Split('_')).
                Distinct().
                Select(w =>
                {
                    int pos = w.IndexOf('(');
                    var wr = (pos == -1) ? w : w.Substring(0, pos);
                    return Extensions.Stemmer.Stem(wr);
                }).Contains(Extensions.Stemmer.Stem(bWord)))
        {
            continue;
        }
        return false;
    }
    return true;
}

private string[] words;
private static readonly string[] wordNetFiles = new string[8]
{
    "data_adj",
    "data_adv",
    "data_noun",
    "data_verb",
    "index_adj",
    "index_adv",
    "index_noun",
    "index_verb"
};

private static readonly WordNetEngine wordNet;

private static readonly string dirPath =
Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.CommonApplicationData),
"OATests", "WordNet");

```

```

static NGram()
{
    wordNet = new WordNetEngine();
    Directory.CreateDirectory(dirPath);
    foreach (string resName in wordNetFiles)
    {
        string filename = Path.Combine(dirPath, resName.Replace('_', '.'));
        if (!File.Exists(filename))
        {
            byte[] bytes = Properties.Resources.ResourceManager.GetObject(resName)
as byte[];
            File.WriteAllBytes(filename, bytes);
        }
    }
    wordNet.LoadFromDirectory(dirPath);
}

public int Length => words.Length;

public NGram(string[] words)
{
    this.words = words ?? throw new ArgumentNullException(nameof(words));
}
}
}

```

AnswerAnalysis/StringAnalysis.cs

```

using DiffLib;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text.RegularExpressions;
using OpenNLP.Tools.Tokenize;
using OpenNLP.Tools.PosTagger;
using System.IO;

namespace AnswerAnalysis
{
    static partial class StringAnalysis
    {
        private static readonly string dirPath =
Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.CommonApplicationData),
"OATests", "OpenNLP");
        private static readonly string posPath = Path.Combine(dirPath,
"EnglishPOS.nbin");
        private static readonly string tokPath = Path.Combine(dirPath,
"EnglishTok.nbin");

        private static void SaveBinAsFiles()
        {
            Directory.CreateDirectory(dirPath);
            if (!File.Exists(posPath))
            {
                File.WriteAllBytes(posPath, Properties.Resources.EnglishPOS);
            }
            if (!File.Exists(tokPath))
            {
                File.WriteAllBytes(tokPath, Properties.Resources.EnglishTok);
            }
        }
    }
}

```


ДОДАТОК Б

Приклади вхідних даних до системи

Система тестування "Open Answers Tests"

(найменування програми (документа))

DVD-R

(вид носія даних)

7 арк, 21.2 Кб

(Обсяг програми (документа) , арк.,) Кб)

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		74

module1/text.txt

Superconductivity is a phenomenon of exactly zero electrical resistance and expulsion of magnetic flux fields occurring in certain materials, called superconductors, when cooled below a characteristic critical temperature. The simplest method to measure the electrical resistance of a sample of some material is to place it in an electrical circuit in series with a current source and measure the resulting voltage across the sample. The resistance of the sample is given by Ohm law. If the voltage is zero, this means that the resistance is zero. The transformer is basically a pair of flux-coupled coils, one with N_p turns connected to the source of alternating voltage, the other with N_s turns connected to the load that actually consumes the energy delivered from the source. All of the flux that passes through any turn in the primary or secondary coils passes through all of the turns in both coils. The flux is usually coupled by wrapping the coils around e.g. a torus of soft iron that traps flux, laminated to prevent eddy currents. Insulators are not ever perfect, because electrons as charge carriers are not bound to the conducting substrate by an infinite potential energy barrier. In a sufficiently large field electrons are torn from their parent atoms and insulators suddenly become conductors, a process called dielectric breakdown. In short, p-type semiconductor has fewer free electrons than normal semiconductor. P stands for positive, meaning a lack of electrons, although the material is actually neutral. The locations where electrons are missing are called holes. This material can conduct electricity well, because electrons can move into the holes, making a new hole somewhere else in the material. Any extra electrons introduced into this region by the circuit will fill some or all of the holes. In n-type semiconductor, the situation is reversed. The material has an more free electrons than normal semiconductor. N stands for negative, meaning an excess of electrons, although the material is actually neutral. When a p-type semiconductor is attached to an n-type semiconductor, some of the free electrons in the n-type move across to the p-type semiconductor. They fill the available holes near the junction. This means that the region of the n-type semiconductor nearest the junction has no free electrons. This makes this n-type semiconductor positively charged. It used to be electrically neutral, but has since lost electrons. The region of p-type semiconductor nearest the junction now has no holes. This makes the p-type semiconductor negatively charged. It used to be electrically neutral, but has since gained electrons. You can explain the high resistance in a different way. A free electron in the n-type semiconductor will be repelled from the p-type semiconductor because of its negative charge. The electron will not go into the depletion band, and certainly wont cross the band to the p-type semiconductor. You may ask, But wont a free electron in the p-type semiconductor be attracted across the band, carrying a current? But there are no free electrons in p-type semiconductor, so no current of this kind can flow. Certain materials, when cooled to extremely low absolute temperatures, become superconductors. Superconductivity is a more or less purely quantum mechanical phenomenon and hence is beyond the scope of this book – basically a fraction of the electronic charge starts to behave collectively like a macroscopic quantum orbital that can transport electronic charge without resistance. Superconductors can be thought of as being diamagnetic – indeed perfectly diamagnetic as they tolerate no magnetic or electric field inside at all, but it isnt exactly the same mechanism as merely opposing an applied field via induction; a superconductor actively ejects any existing magnetic field as it is cooled across the transition temperature where superconductivity appears, even if that field is not changing. One visible sign of this ejection is that superconductors placed above a permanent magnet float, suspended by its perfectly opposed magnetic field. Electrostatics is a branch of physics that studies electric charges at rest. Since classical physics, it has been known that some materials such as amber attract lightweight particles after rubbing. Electric field lines are useful for visualizing the electric field. Field lines begin on positive charge and terminate on negative charge. They are parallel to the direction of the electric field at each point, and the density of these field lines is a measure of the magnitude of the electric field at any given point. In other words, electrostatics does not require the absence of magnetic fields or electric currents. Rather, if magnetic fields or electric currents do exist, they must not change with time, or in the worst-case, they must change with time only very slowly. In some problems, both electrostatics and magnetostatics may be required for accurate predictions, but the coupling between the two can still be ignored. Electrostatics and magnetostatics can both be seen as Galilean limits for electromagnetism. There are many examples of electrostatic phenomena, from those as simple as the attraction of the plastic wrap to one's hand after it is removed from a package to the apparently spontaneous explosion of grain silos, the damage of electronic components during manufacturing, and photocopier & laser printer operation. Electrostatics involves the buildup of charge on the surface of objects due to contact with other surfaces. Although charge exchange happens whenever any two surfaces contact and separate, the effects of charge exchange are usually only noticed when at least one of the surfaces has a high resistance to electrical flow. This is because the charges that transfer are trapped there for a time long enough for their effects to be observed. These charges then remain on the object until they either bleed off to ground or are quickly neutralized by a discharge: e.g., the familiar phenomenon of a static 'shock' is caused by the neutralization of charge built up in the body from contact with insulated surfaces.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

module1/question1.txt

Title : "Superconductivity is a phenomenon of (a/an) ..."

Given answers:

- "zero electrical resistance" !
- "exactly zero resistance" !
- "exactly zero electric resistance" !
- "zero resistance" !
- "phenomenon of zero resistance" !
- "no resistance" !
- "no resist" !
- "zero electric resistance" 1.0
- "zero electricity resistance" 0.8
- "exectli zero resistance" 0.6
- "zro electric resist" 0.3
- "exactli zeroo rsistance" 0.3
- "resistance of"
- "exactly zero"
- "resistance zero"
- "and electric"
- "electric resistance"
- "electrical resistance"
- "exactly this"
- "exactly resistance"
- "exactly resist"
- "zero electric"
- "zero some text"
- "pressure of electricity"
- "non-zero electrical resistance"
- "non-zero resistance"
- "not zero electrical resistance"
- "not zero resistance"

module1/question2.txt

Title : "The interaction between two separated charged bodies is called..."

Given answers:

- "electrostatic force" !
- "electric force" !
- "electrostatics phenomenon" !
- "electrostatic interaction" 0.9
- "electricity" 0.5
- "electic force" 0.5
- "elections"
- "electrocar"
- "electron"
- "elevator"
- "law of"
- "some sample text"
- "Iksadlk ladklaklj akslj"
- "nouton law"
- "and force"
- "archimedes law"
- "exactly this"
- "pressure of electricity"

module2/text.txt

Archimedes' law states that the upward buoyant force that is exerted on a body immersed in a fluid, whether fully or partially submerged, is equal to the weight of the fluid that the body displaces and acts in the upward direction at the center of mass of the displaced fluid. Archimedes' principle is a law of physics fundamental to fluid mechanics. It was formulated by Archimedes of Syracuse. Practically, Archimedes' principle allows the buoyancy of an object partially or fully immersed in a fluid to be calculated. The downward force on the object is simply its weight. The upward, or buoyant, force on the object is that stated by Archimedes' principle, above. Thus, the net force on the object is the difference between the magnitudes of the buoyant force and its weight. If this net force is positive, the object rises; if negative, the object sinks; and if zero, the object is neutrally buoyant - that is, it remains in place without either rising or sinking. In simple words, Archimedes' principle states that, when a body is partially or completely immersed in a fluid, it experiences an apparent loss in weight that is equal to the weight of the fluid displaced by the immersed part of the body. Solid mechanics also known as Mechanics of solids is the branch of continuum mechanics that studies the behavior of solid materials, especially their motion and deformation under the action of forces, temperature changes, phase changes, and other external or internal agents. It has specific applications in many other areas, such as understanding the anatomy of living beings, and the design of dental prostheses and surgical implants. One of the most common practical applications of solid mechanics is the Euler-Bernoulli beam equation. Solid mechanics extensively uses tensors to describe stresses, strains, and the relationship between them. Fluid mechanics is the branch of physics concerned with the mechanics of fluids (liquids, gases, and plasmas) and the forces on them. It has applications in a wide range of disciplines, including mechanical, civil, chemical and biomedical engineering, geophysics, astrophysics, and biology. Fluid Mechanics can also be defined as the science which deals with the study of behaviour of fluids either at rest or in motion. It can be divided into fluid statics, the study of fluids at rest; and fluid dynamics, the study of the effect of forces on fluid motion. It is a branch of continuum mechanics, a subject which models matter without using the information that it is made out of atoms; that is, it models matter from a macroscopic viewpoint rather than from microscopic. Fluid mechanics, especially fluid dynamics, is an active field of research, typically mathematically complex. Many problems are partly or wholly unsolved, and are best addressed by numerical methods, typically using computers. A modern discipline, called computational fluid dynamics (CFD), is devoted to this approach. Particle image velocimetry, an experimental method for visualizing and analyzing fluid flow, also takes advantage of the highly visual nature of fluid flow. The study of fluid mechanics goes back at least to the days of ancient Greece, when Archimedes investigated fluid statics and buoyancy and formulated his famous law known now as the Archimedes' principle, which was published in his work On Floating Bodies—generally considered to be the first major work on fluid mechanics. Fluid statics or hydrostatics is the branch of fluid mechanics that studies fluids at rest. It embraces the study of the conditions under which fluids are at rest in stable equilibrium; and is contrasted with fluid dynamics, the study of fluids in motion. Hydrostatics offers physical explanations for many phenomena of everyday life, such as why atmospheric pressure changes with altitude, why wood and oil float on water, and why the surface of water is always level whatever the shape of its container. Hydrostatics is fundamental to hydraulics, the engineering of equipment for storing, transporting and using fluids. It is also relevant to some aspects of geophysics and astrophysics (for example, in understanding plate tectonics and anomalies in the Earth's gravitational field), to meteorology, to medicine (in the context of blood pressure), and many other fields. Fluid dynamics is a subdiscipline of fluid mechanics that deals with fluid flow—the science of liquids and gases in motion. Fluid dynamics offers a systematic structure—which underlies these practical disciplines—that embraces empirical and semi-empirical laws derived from flow measurement and used to solve practical problems. The solution to a fluid dynamics problem typically involves calculating various properties of the fluid, such as velocity, pressure, density, and temperature, as functions of space and time. It has several subdisciplines itself, including aerodynamics (the study of air and other gases in motion) and hydrodynamics (the study of liquids in motion). Fluid dynamics has a wide range of applications, including calculating forces and movements on aircraft, determining the mass flow rate of petroleum through pipelines, predicting evolving weather patterns, understanding nebulae in interstellar space and modeling explosions. Some fluid-dynamical principles are used in traffic engineering and crowd dynamics. Newton's laws of motion are three physical laws that, together, laid the foundation for classical mechanics. They describe the relationship between a body and the forces acting upon it, and its motion in response to those forces. More precisely, the first law defines the force qualitatively, the second law offers a quantitative measure of the force, and the third asserts that a single isolated force doesn't exist. These three laws have been expressed in several ways, over nearly three centuries, and can be summarised as follows: In an inertial frame of reference, an object either remains at rest or continues to move at a constant velocity, unless acted upon by a force. In an inertial frame of reference, the vector sum of the forces F on an object is equal to the mass m of that object multiplied by the acceleration a of the object: $F = ma$. (It is assumed here that the mass m is constant – see below.) When one body exerts a force on a second body, the second body simultaneously exerts a force equal in magnitude and opposite in direction on the first body.

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

module2/question1.txt

Title : "Law that keeps things float in fluids is called..."

Given answers:

- "archimedes principle" !
- "archimedes law" !
- "law of archimedes" !
- "principle of archimedes" !
- "archimedes force" !
- "force of archimedes" !
- "archimed law" 0.9
- "law of the archimed" 0.8
- "archimed principle" 0.8
- "archimed" 0.5
- "archimedes" 0.6
- "archemedes law" 0.8
- "archemed law" 0.7
- "fluid floating"
- "law of equivalent exchange"
- "so law archimed"
- "floating principle"
- "archimedes text words"
- "Iksadlk ladklaklj akslj"
- "archimedes law no"
- "law of"
- "and principle"

module2/question2.txt

Title : "Three physical laws that form the foundation for classical mechanics are called..."

Given answers:

- "Newton's laws of motion" !
- "Newton laws" !
- "laws of newton" !
- "newtons laws" !
- "newtons principles" !
- "newton" 0.6
- "neuton" 0.4
- "neuton laws" 0.8
- "laws of the england"
- "newton text"
- "law of equivalent exchange"
- "law and new ton"
- "force principle"
- "Iksadlk ladklaklj akslj"
- "nouton law"
- "and force"
- "archimedes law"
- "andrew's keys of house"

module3/text.txt

where x represents an unknown, and a, b, and c represent known numbers, then the equation is linear, not quadratic, as there is no term. The numbers a, b, and c are the coefficients of the equation and may be distinguished by calling them, respectively, the quadratic coefficient, the linear coefficient and the constant or free term. In mathematics, a surface integral is a generalization of multiple integrals to integration over surfaces. It can be thought of as the double integral analogue of the line integral. Given a surface, one may integrate over its scalar fields (that is, functions which return scalars as values), and vector fields (that is, functions which return vectors as values). Surface integrals have applications in physics, particularly with the theories of classical electromagnetism. In mathematics, the Pythagorean theorem, also known as Pythagoras' theorem, is a fundamental relation in Euclidean geometry among the three sides of a right triangle. It states that the square of the hypotenuse (the side opposite the right angle) is equal to the sum of the squares of the other two sides. The theorem can be written as an equation relating the lengths of the sides a, b and c, often called the "Pythagorean equation". the theorem is named after the ancient Greek mathematician Pythagoras The theorem has been given numerous proofs – possibly the most for any mathematical theorem. They are very diverse, including both geometric proofs and algebraic proofs, with some dating back thousands of years. The theorem can be generalized in various ways, including higher-dimensional spaces, to spaces that are not Euclidean, to objects that are not right triangles, and indeed, to objects that are not triangles at all, but n-dimensional solids. The Pythagorean theorem has attracted interest outside mathematics as a symbol of mathematical abstruseness, mystique, or intellectual power; popular references in literature, plays, musicals, songs, stamps and cartoons abound. In mathematics, the polar coordinate system is a two-dimensional coordinate system in which each point on a plane is determined by a distance from a reference point and an angle from a reference direction. The reference point (analogous to the origin of a Cartesian coordinate system) is called the pole, and the ray from the pole in the reference direction is the polar axis. The distance from the pole is called the radial coordinate, radial distance or simply radius, and the angle is called the angular coordinate, polar angle, or azimuth. which is the standard formula for the area of a surface described this way. One can recognize the vector in the second line above as the normal vector to the surface. Although the quadratic formula provides an exact solution, the result is not exact if real numbers are approximated during the computation, as usual in numerical analysis, where real numbers are approximated by floating point numbers (called "reals" in many programming languages). In this context, the quadratic formula is not completely stable. Note that because of the presence of the cross product, the above formulas only work for surfaces embedded in three-dimensional space. A quadratic equation with real or complex coefficients has two solutions, called roots. These two solutions may or may not be distinct, and they may or may not be real. This can be seen as integrating a Riemannian volume form on the parameterized surface, where the metric tensor is given by the first fundamental form of the surface. Euclidean geometry is a mathematical system attributed to Alexandrian Greek mathematician Euclid, which he described in his textbook on geometry: the Elements. Euclid's method consists in assuming a small set of intuitively appealing axioms, and deducing many other propositions (theorems) from these. Although many of Euclid's results had been stated by earlier mathematicians, Euclid was the first to show how these propositions could fit into a comprehensive deductive and logical system. The Elements begins with plane geometry, still taught in secondary school (high school) as the first axiomatic system and the first examples of formal proof. It goes on to the solid geometry of three dimensions. Much of the Elements states results of what are now called algebra and number theory, explained in geometrical language. For more than two thousand years, the adjective "Euclidean" was unnecessary because no other sort of geometry had been conceived. Euclid's axioms seemed so intuitively obvious (with the possible exception of the parallel postulate) that any theorem proved from them was deemed true in an absolute, often metaphysical, sense. Today, however, many other self-consistent non-Euclidean geometries are known, the first ones having been discovered in the early 19th century. An implication of Albert Einstein's theory of general relativity is that physical space itself is not Euclidean, and Euclidean space is a good approximation for it only over short distances (relative to the strength of the gravitational field). Euclidean geometry is an axiomatic system, in which all theorems ("true statements") are derived from a small number of simple axioms. Until the advent of non-Euclidean geometry, these axioms were considered to be obviously true in the physical world, so that all the theorems would be equally true. However, Euclid's reasoning from assumptions to conclusions remains valid independent of their physical reality. The parallel postulate (Postulate 5): If two lines intersect a third in such a way that the sum of the inner angles on one side is less than two right angles, then the two lines inevitably must intersect each other on that side if extended far enough.

module3/question1.txt

Title : "Regular geometry that is built on a five axioms is called..."

Given answers:

- "Euclidean" !
- "Euclidean geometry" !
- "geometry of euclidean" !
- "geometry of euclid" 0.9
- "euclid" 0.5
- "euclidian" 0.8
- "geometry"
- "geo metry"
- "Iksadlk ladklaklj aksj"
- "figures of"
- "and or"
- "geometry geometry"
- "Pythagorean theorem"
- "non-euclidean geometry"
- "lobachevsky"
- "euclidean euclidean"
- "theorem giometry"
- "flux capacitor"

module3/question2.txt

Title : "Fundamental relation which states that the square of the hypotenuse is equal to the sum of the squares of the other two sides is known as..."

Given answers:

- "pythagorean theorem" !
- "pythagoras theorem" !
- "theorem of pythagoras" !
- "theorem pythagoras" 0.9
- "theorem pythagorean" 0.9
- "pythagorean" 0.8
- "pythagoras" 0.8
- "pythagor" 0.5
- "pythagorean relation" 0.4
- "theorem theorem"
- "theorem of theorem"
- "pythagoras pythagor"
- "geo metry"
- "Iksadlk ladklaklj aksj"
- "figures of"
- "and or"
- "non-euclidean geometry"
- "lobachevsky"
- "fundamental relation"

synonymQuestionSample.txt

Title : "Synonym sample question"

Given answers:

"big thing" !

"great thing" !

"major thing" !

"high thing" !

"massive thing" !

"thing thing"

"spsodfkj skldfjsdlkf"

"sadjksdlk"

"sample text"

"bigg thing" 0,8

Tested answer : "large thing"

					ДП ІС-5127.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		81

Сторінка управління питаннями тесту

OATests Modules Results

Questions of test E/M Module test

The interaction between two separated charged bodies is called...

Remove

Superconductivity is a phenomenon of (a/an) ...

Add

[Back to details](#)

Сторінка управління тестом

OATests Modules Results

Details of test "E/M Module test"

Name:

Max questions:

Save

Start

Results

Manage questions

Delete

Back to List

OATests Modules Results

Excercise

User name:

Superconductivity is a phenomenon of (a/an) ...

The interaction between two separated charged bodies is called...

Finish

Сторінка проходження тестування

Сторінка відображення результатів спроби

OATests Modules Results

E/M Module test

By UserName01, 02.06.19 12:16

Total - 1,00

Superconductivity is a phenomenon of (a/an) ...

Answer: exactly zero resistance

Mark: 1,00

The interaction between two separated charged bodies is called...

Answer: electricity

Mark: 0,00

[Return to test details](#)

					ДП ІС-5127.1181-с.КЕ						
					Креслення вигляду екранних форм	Літера		Маса	Масштаб		
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Сулейманов А.К.									
Перевірив		Звенігородський О.С.									
Т. кон.						Аркуш 1		Аркушів 1			
Н. кон.		Москаленко Н.В.			Семантичний аналіз відповідей природною мовою в текстових системах					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51	
Затвердив		Звенігородський О.С.									

Рішення з математичного забезпечення

Математична модель

Дано:

$U_{\text{кор}}$ – рядок користувацької відповіді;

S_i - приклад правильної відповіді, $i = \overline{1, m}$;

m - кількість прикладів правильних відповідей;

A_j - приклад не повністю правильної відповіді;

$r_j \in [0; 1]$ – оцінка такої відповіді;

$j = \overline{1, n}$, n - кількість прикладів відповідей;

t – кількість модулів;

q_t – кількість різних слів у тексті t -го модуля;

$k_{t q_t}$ – кількість використань q_t -го слова у тексті t -го модуля.

Функція $P(S, K, t, u)$, що визначає валідність відповіді u у відповідності з множиною правильних відповідей S , множиною текстів модулів K та номером модуля t , до якого належить питання, невідома.

Необхідно оцінити значення $P(S, K, t, U_{\text{кор}})$.

Демонстраційний плакат до дипломного проекту

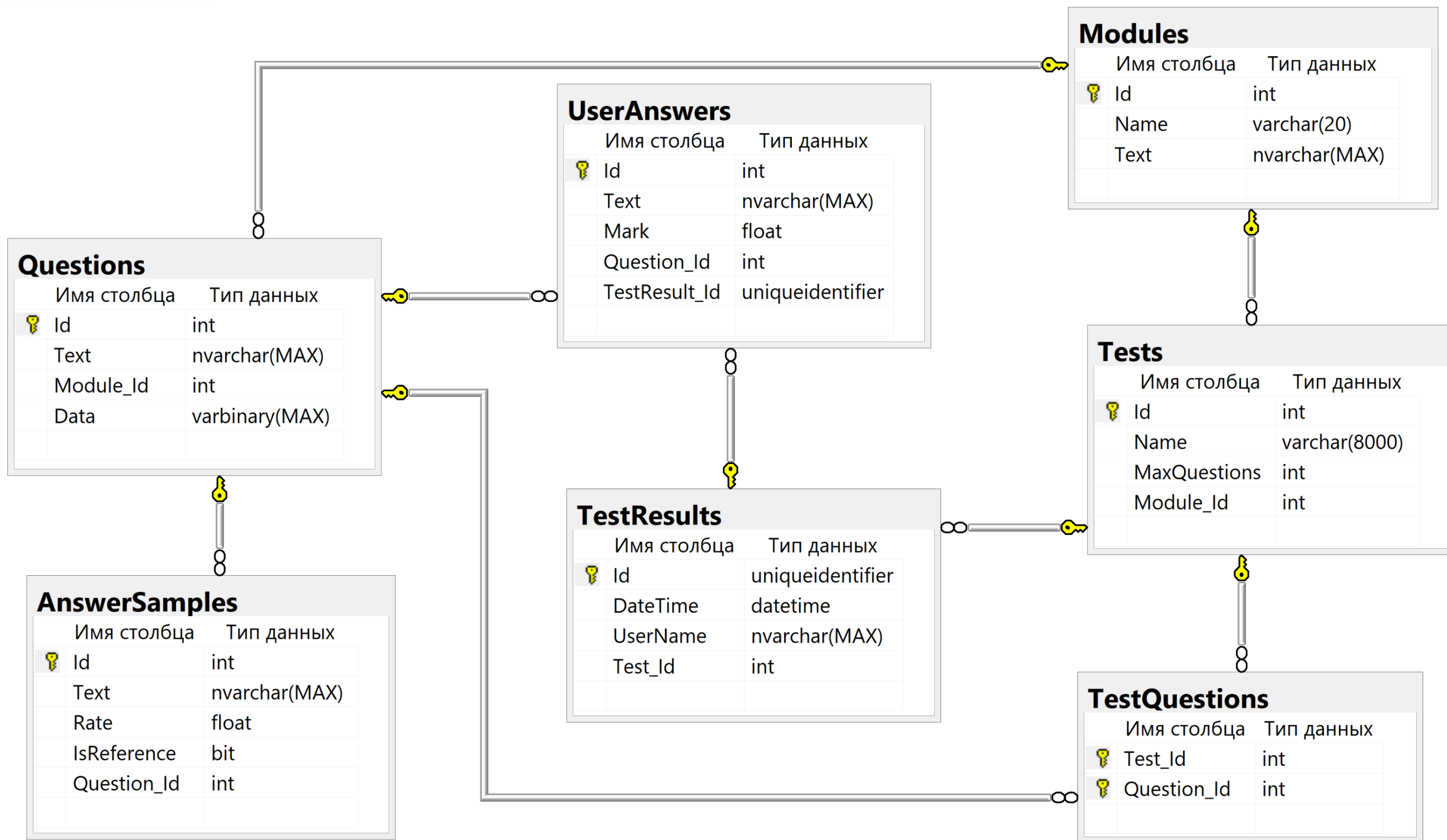
„Семантичний аналіз відповідей природною мовою в текстових системах ”

Виконав студент гр. ІС-51

Сулейманов А.К.

Керівник ДП

Звенігородський А.С.



					ДП ІС-5127.1181-с.СБД					
					Схема бази даних			Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата						
Розробив		Сулейманов А.К.						Аркуш 1	Аркушів 1	
Перевірив		Звенигородський О.С.								
Т. кон.					Семантичний аналіз відповідей природною мовою в текстових системах			КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51		
Н. кон.		Москаленко Н.В.								
Затвердив		Звенигородський О.С.								

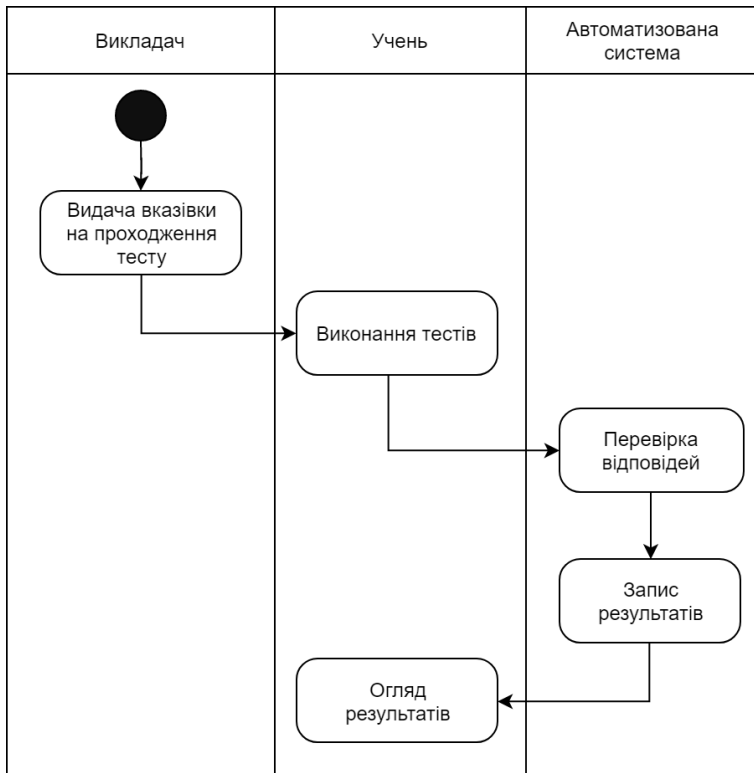


Схема структурна діяльності при тестуванні у ручному режимі

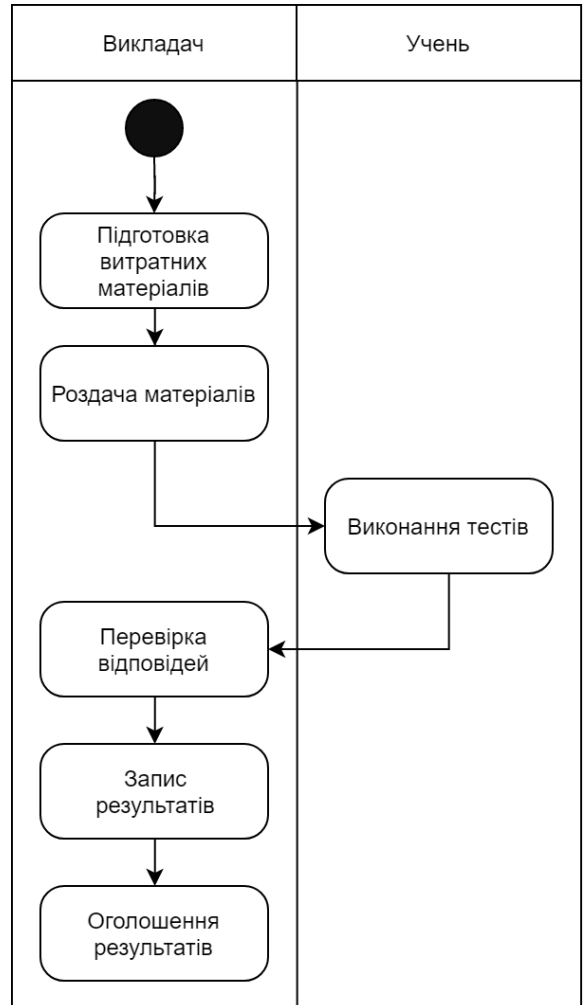
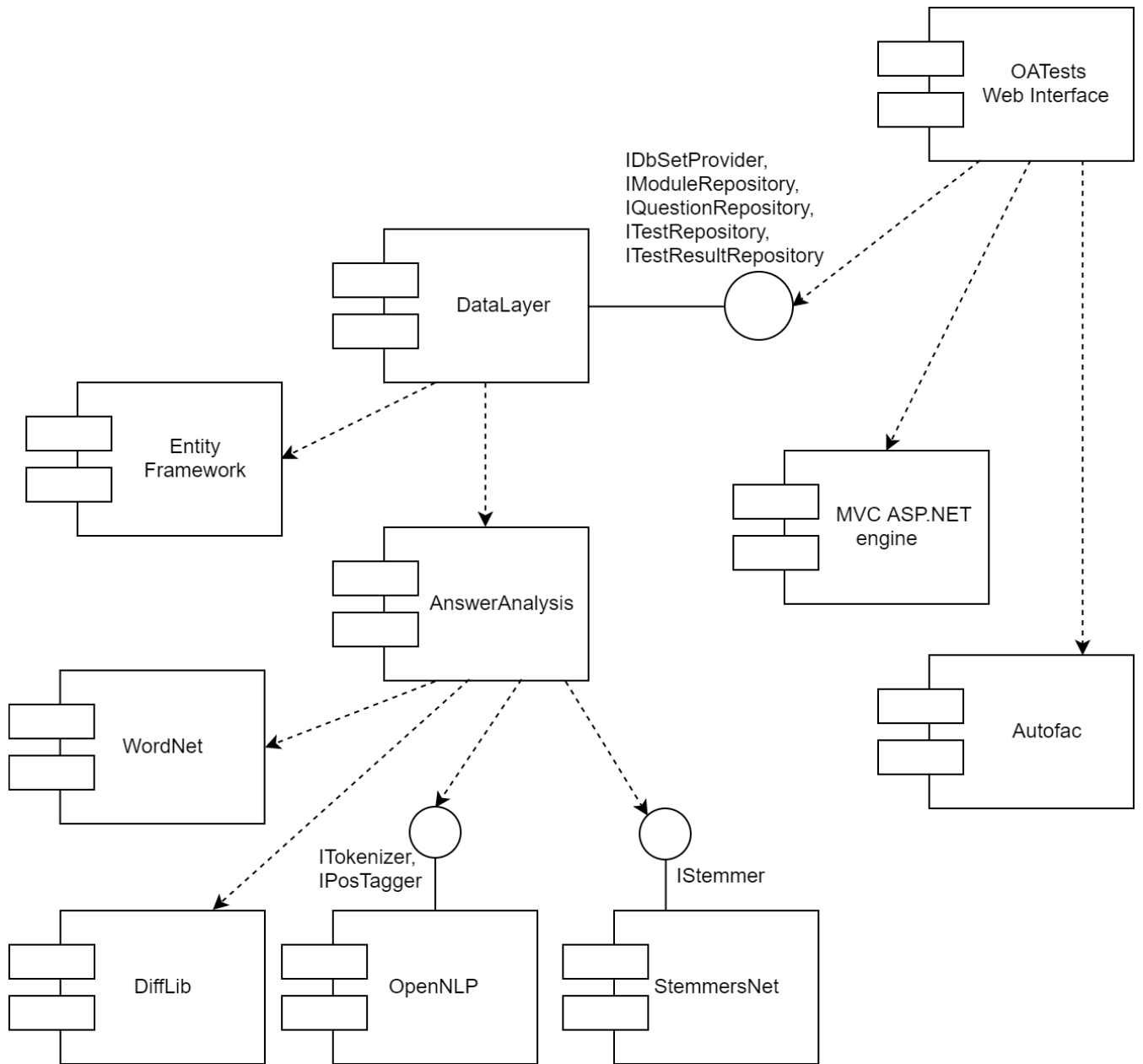
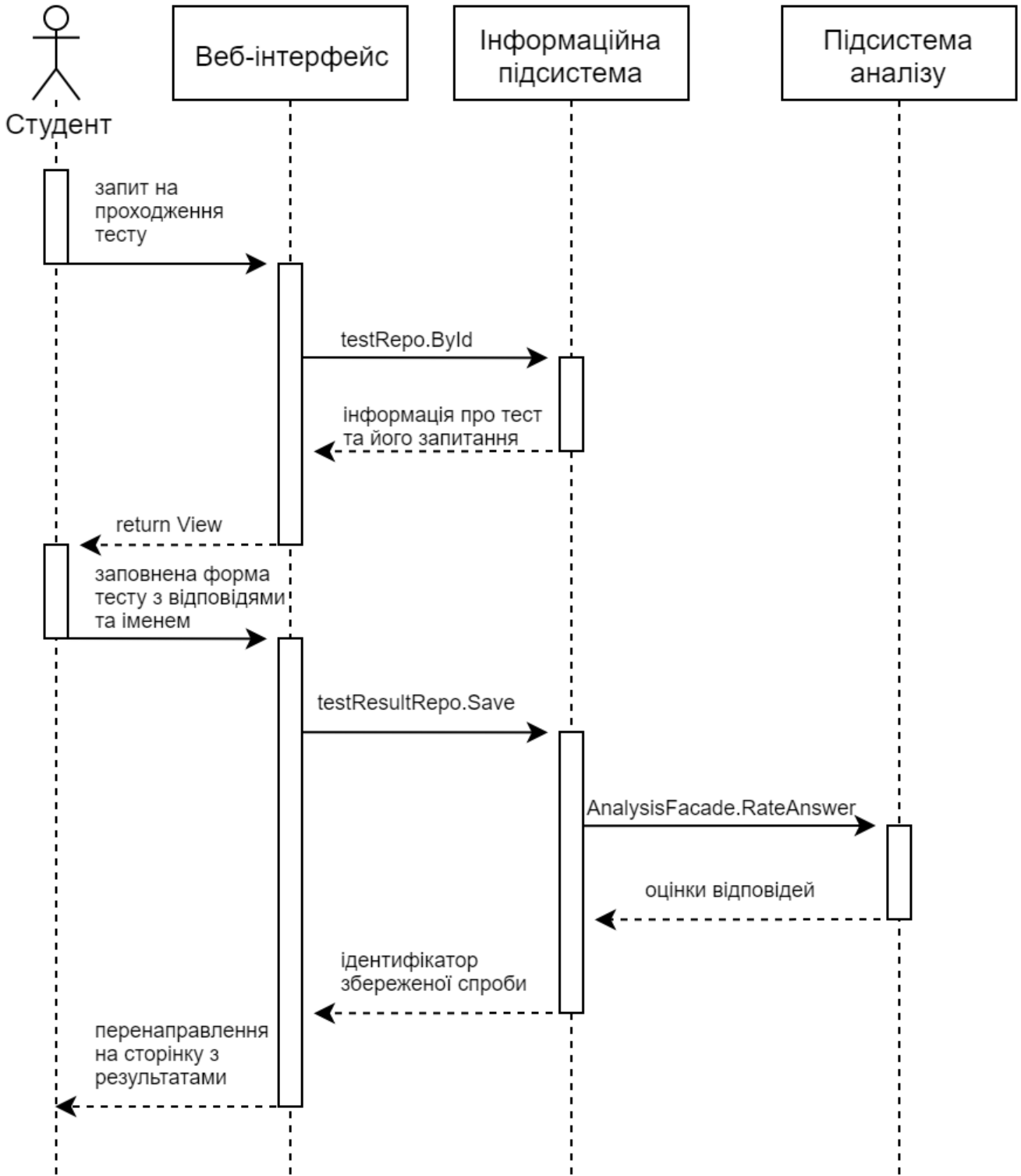


Схема структурна діяльності із запровадженням автоматизованих засобів тестування

					ДП ІС-5127.1181-с.ССД					
					Схема структурна діяльності			Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата						
Розробив	Сулейманов А.К.									
Перевірив	Звенигородський О.С.									
Т. кон.								Аркуш 1	Аркушів 1	
Н. кон.	Москаленко Н.В.				<i>Семантичний аналіз відповідей природною мовою в текстових системах</i>			<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51</i>		
Затвердив	Звенигородський О.С.									



					ДП ІС-5127.1181-с.ССК		
					Схема структурна компонентів програмного забезпечення		
Зм.	Арк.	№ документа	Підпис	Дата			
Розробив		Сулейманов А.К.					
Перевірів		Звенигородський О.С.					
Т. кон.					Аркуш 1	Аркушів 1	
Н. кон.		Москаленко Н.В.			<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51</i>		
Затвердив		Звенигородський О.С.					
					<i>Семантичний аналіз відповідей природною мовою в текстових системах</i>		



					ДП ІС-5127.1181-с.ССП			
					Схема структурна послідовності			
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Сулейманов А.К.						
Перевірив		Звенигородський О.С.						
Т. кон.					Аркуш 1		Аркушів 1	
Н. кон.		Москаленко Н.В.			КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51			
Затвердив		Звенигородський О.С.						
					Семантичний аналіз відповідей природною мовою в текстових системах			