

Міністерство освіти і науки України  
Рівненський державний гуманітарний університет  
Кафедра інформаційних технологій та моделювання

**Кваліфікаційна робота**

за освітнім ступенем «бакалавр»

на тему:

**НАВЧАЛЬНА КОМП'ЮТЕРНА ПРОГРАМА  
«АНГЛІЙСЬКА ДЛЯ ПРОГРАМІСТІВ»**

**Виконав:**

здобувач ІV курсу

групи ПЗ-41

спеціальності 121 «Інженерія  
програмного забезпечення»

Гарбарчук І.С.

**Науковий керівник:**

к.т.н., доцент Бабич С.М.

## ЗМІСТ

ВСТУП .....	3
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ТА ПОСТАНОВКА ЗАВДАННЯ .....	6
1.1. Огляд сучасних методик вивчення іноземних мов.....	6
1.2. Порівняльний аналіз програм для вивчення англійської мови .....	11
1.3. Постановка завдання.....	16
РОЗДІЛ 2. ПРОЄКТУВАННЯ ДОДАТКУ З ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ.....	19
2.1. Визначення функціональних та нефункціональних вимог до додатку .	19
2.2. Розробка проєкту додатку з вивчення англійської мови .....	20
2.3. Обґрунтування вибору засобів розробки.....	25
РОЗДІЛ 3. РОЗРОБКА ДОДАТКУ З ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ..	28
3.1. Проєктування інтерфейсу користувача .....	28
3.2. Створення системи з вивчення англійської мови .....	33
3.3. Тестування роботи програми .....	43
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
ДОДАТКИ.....	56

## ВСТУП

**Актуальність роботи.** Сучасний світ інформаційних технологій вимагає від фахівців у галузі ІТ не лише глибоких технічних знань, але й високого рівня володіння англійською мовою, яка є міжнародним стандартом спілкування в цій сфері. Знання англійської мови не тільки розширюють можливості комунікації фахівців, але й відкривають двері до світових технологічних та інноваційних трендів, можливості читати спеціалізовану літературу. Багато програмних бібліотек, документація, форуми та онлайн-ресурси для програмістів доступні саме англійською мовою. Розуміння цієї документації допомагає програмістам краще розуміти функції та можливості різних інструментів.

Велика частина програмного забезпечення розробляється в міжнародних командних середовищах, де англійська є основною мовою комунікації. Щоб ефективно співпрацювати з колегами з усього світу, важливо володіти англійською мовою. До того ж знання англійської мови розширює можливості для кар'єрного зростання. Багато вакансій в інформаційних технологіях вимагають від програмістів доброго рівня володіння англійською мовою.

Однак, попри важливість англійської мови, багато ІТ-професіоналів стикаються з труднощами у її вивченні та практиці застосування. У них може не вистачати часу для відвідування курсів, а використання допоміжних програм досить часто не приносить бажаного результату. Це пов'язано з тим, що вони в більшості орієнтовані на використання загальної лексики та виразів. У комп'ютерній літературі за роки розвитку галузі склався свій сленг та ідіоми. Їх знання є необхідною складовою при вивченні англійської мови.

У цьому контексті виникає актуальна задача розробки програми, спрямованої на покращення мовних навичок серед фахівців ІТ. Враховуючи особливості термінології та вимог до комунікації в цій галузі,

розробка такої програми має велике значення для успішної кар'єри та професійного росту спеціалістів у сфері інформаційних технологій.

Розробка програмної системи для вивчення англійської мови буде сприяти розвитку здатностей ІТ-фахівців до успішного використання англійської мови в їхній повсякденній практиці та професійному зростанні.

**Метою дослідження** є розробка ефективної програми для поліпшення мовних компетенцій в ІТ-фахівців.

Для досягнення мети потрібно вирішити такі **завдання**:

1. дослідити методики вивчення іноземних мов з використанням технічних засобів;
2. проаналізувати сучасні програмні засоби для вивчення іноземних мов ІТ-фахівцями;
3. визначити модель вивчення англійської мови за допомогою програми;
4. провести проектування системи вивчення англійської мови ІТ-фахівцями;
5. розробити програмний додаток вивчення англійської мови ІТ-фахівцями.

**Об'єктом дослідження** є процес розробки програмної системи для вивчення англійської мови ІТ-фахівцями.

**Предмет дослідження** – інструменти розробки програмної системи для вивчення англійської мови ІТ-фахівцями.

Для вивчення проблеми були використані загальнонаукові **методи**, серед яких методи: аналізу, синтезу, узагальнення та порівняння, наукові методи, такі як: моніторинг, аналіз документів, методи об'єктно-орієнтованого проектування та алгоритмічного програмування.

**Практичне значення дослідження.** Розроблена програма може використовуватися фахівцями як десктопний додаток, який допомагатиме їм у вивченні фахових термінів на англійській мові.

**Апробація і впровадження результатів дослідження.** Основні теоретичні і практичні результати дослідження доповідалися та обговорювалися на XXIV Всеукраїнській науково-технічній конференції молодих вчених, аспірантів та студентів «Стан, досягнення та перспективи інформаційних систем і технологій» (м. Одеса, 18-19 квітня 2024 р.) [1], звітній науковій конференції викладачів, співробітників і здобувачів вищої освіти Рівненського державного гуманітарного університету за 2023 рік (м. Рівне, 16-17 травня 2024 року), а також були представлені на I етапі Всеукраїнського конкурсу науково-дослідних студентських робіт і відзначені дипломом III ступеня (м. Рівне, РДГУ, 2024 р.).

**Структура роботи.** Робота складається з вступу, трьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи становить 53 сторінки. Вона містить 33 рисунки та 5 таблиць. Список використаних джерел включає 16 найменувань. Обсяг додатків – 18 сторінок.

## РОЗДІЛ 1

### АНАЛІТИЧНИЙ ОГЛЯД ТА ПОСТАНОВКА ЗАВДАННЯ

#### 1.1. Огляд сучасних методик вивчення іноземних мов

Світ помітно змінився в останні десятиліття, і володіння іноземними мовами стає в ньому нагальною потребою. А для фахівців з області інформаційних технологій (ІТ) вивчення англійської мови стає невід'ємною частиною професійного розвитку. Англійська є мовою комунікації у глобальному ІТ-середовищі, де спілкування з колегами, клієнтами та партнерами іноді відбувається виключно англійською. Знання цієї мови дозволяє ІТ-професіоналам легше обмінюватися ідеями, взаємодіяти з міжнародними командами та впроваджувати передові технології на міжнародному ринку.

Крім того, багато інновацій та технологічних ресурсів, таких як документація, онлайн-курси та форуми, доступні виключно англійською мовою. Здатність розуміти та користуватися цими ресурсами відкриває для фахівців з ІТ широкі можливості для самостійного навчання та оновлення своїх навичок, сприяючи стійкому професійному зростанню в динамічному світі технологій.

Знання іноземної мови можна окреслити такими етапами [2]:

- вміння правильно прочитати, у разі потреби, та зрозуміти за допомогою словника назви книг, адреси на конвертах, посилках, текст накладних тощо, що потрібно для окремих категорій бібліотечних службовців, для працівників зв'язку, транспорту, для кваліфікованих робітників різних виробництв;

- вміння висловити своє бажання та поставити найпростіші питання (можливо з невеликими неточностями), зрозуміти відповіді на такі питання. Подібне вміння можна було б назвати "туристською мовою", але сфера його застосування є, насправді, набагато ширшою. При поїздки за

кордон це вміння слід поєднати з вмінням читати та орієнтуватися у всіх написах, а також по можливості в газетних заголовках;

- вміння точно зрозуміти всякий нехудожній текст стосовно будь-якої проблеми, не розуміючи окремі слова і лише зрідка користуючись словником. Такого роду вміння необхідне науковцям, інженерам, студентам, усім тим, кому необхідно стежити за іноземною літературою у тій чи іншій галузі;

- вміння підтримувати розмову на будь-яку тему, роблячи при цьому невеликі помилки, проте мова цілком зрозуміла як з погляду фонетики, так і з погляду лексики і граматики. Такі навички необхідні людям, змушеним вести більш менш відповідальні розмови з іноземцями;

- вміння грамотно писати наукові та технічні статті, ділові папери та листи може бути необхідно попередньої категорії осіб;

- вміння вільно і тонко розуміти найважчі тексти, між іншим, художні, газетні та інші. Воно необхідне письменникам, критикам, літературознавцям-публіцистам, політичним діячам і насамперед, викладачам іноземних мов та перекладачам;

- вміння добре писати відповідальні документи, літературні статті тощо. необхідно для дипломатичних працівників та журналістів, що пишуть для іноземних видань;

- вміння вільно та абсолютно правильно, з погляду фонетики, говорити публічно необхідно дипломатичним працівникам та всім тим, хто виступає публічно.

Отже, знання іноземної мови можуть бути дуже диференційовані залежно від практичних потреб. Щодо фахівців в області ІТ, то вони мають, як мінімум, вміти розуміти нетехнічний текст, оскільки більшість літератури, блогів, документації з їх фаху представлена на англійській мові. Але, у більшості випадків, враховуючи, що робота над проектами може відбуватися в багатонаціональних командах, ІТ фахівці потребують вміння вести розмову, вміння грамотно писати тести англійською мовою.

Зараз існує декілька методик, які повинні забезпечити ефективне вивчення іноземної і, зокрема, англійської мови. У даній роботі під методикою будемо розуміти сукупність засобів і методів навчання.

Аудіолінгвальний підхід (Audiolingual approach) зародився в 1950-х роках, ґрунтуючись на припущенні, що мова – це набір патернів, які можна вивчити багаторазовим повторенням. Для їхнього освоєння учні слухали тексти та діалоги та повторювали за аудіозаписом типові пропозиції. У мовах справді існують стандартні шаблони фраз та обміну фразами на кшталт "Дякую – Будь ласка". Однак творці методу не врахували, що бездумне заучування пропозицій призводить до ефекту папуги, який повторює, але не породжує сам. Нині метод у чистому вигляді не існує [3].

Граматично-перекладна методика (Grammar-translation method), відома ще з XVI століття, ґрунтується на першості граматики як серцевини мови та перекладу як головного способу її освоєння. Метод дуже добре сприяє розвитку вміння розуміти тексти, однак, вона не забезпечує ефективний розвиток комунікаційних навичок.

Метод повного занурення (Total immersion) означає занурення людини у мовне середовище. Оскільки людина не має можливості спілкуватись рідною мовою, вона вимушено вбудовується в іноземний контекст. Мовне середовище безумовно дає великі можливості для навчання, але все ж таки для абсолютної більшості людей, крім занурення, потрібні регулярні уроки, які додадуть структурності та суттєво прискорять процес [3].

Dogme є відгалуженням комунікативної методики та водночас філософський рух. Воно зародилося як реакція на надлишок стандартизованих курсів і підручників, коли вчителі зрозуміли, що за типовою програмою вони втратили в паперах учнів. Dogme має на увазі роботу без опори на матеріал: учитель та учні приходять у клас з порожніми руками, а в процесі спілкування народжується тема уроку. На таких заняттях учні мають можливість самостійно визначати тематику та



напрямок свого навчання, вести живі бесіди з однокласниками та підвищувати рівень, постійно працюючи над помилками. Однак Dogme вимагає колосальних зусиль та професіоналізму з боку вчителя: він повинен з ходу вміти пояснити будь-яку граматичну тему, направити діалог у потрібне русло, знати безліч способів роботи над помилками, на ходу вигадувати завдання для закріплення вивченого, вміти без тестів відстежувати прогрес учнів. Противники Dogme критикують такий підхід за створення штучних обмежень та зайве ускладнення і так непростого процесу навчання.

Аналіз останніх робіт щодо вивчення іноземної мови показує, що багато методів, які виявляються неспроможними при застосуванні в чистому вигляді, дають хороші результати, якщо доповнювати ними основне навчання. Один із них – винахід початку ХХ століття – метод фізичного реагування (Total physical response). Він базується на концепції, що учню, перш ніж заговорити іноземною мовою, необхідно пройти так званий "період мовчання", коли слухатиме багато іноземної мови. У цей період вчитель розмовляє з учнями, використовуючи в основному дієслова, і показує рухи, називаючи їх мовою, що вивчається. Пізніше учні привчаються реагувати на команди вчителя – "Підведіться", "Сядьте", "Поплескайте в долоні". Особливість цієї методики в тому, що вона задіює кінестетику та м'язову пам'ять, що дуже рідко застосовується у вивченні іноземних мов. Зрозуміло, що цей метод не дає можливості вільно заговорити, тому його почали комбінувати з комунікативним підходом. Особливо добрі результати це дає під час навчання дітей [4].

Методикою з доведеною ефективністю є комунікативна методика (Communicative approach). Це продумана система розвитку мовних навичок через ситуації, наближені до реального життя. Учні не зубрять граматику заради граматики і не читають нескінченні тексти вголос – вони розігрують вправи, які максимально повторюють ситуації, у яких вони використовуватимуть іноземну мову в реальному житті. При цьому

розвивається не тільки розмовна мова, а й письмова мова, читання, аудіювання, адже все вказане входить у поняття "комунікація" [5, 6].

Комунікативний підхід відомий із кінця 60-х. За минулі десятиліття методика доопрацьовувалась і вдосконалювалася, а її ефективність була доведена науково на дослідженні не одного покоління. Вона стала базою багатьох навчальних курсів з англійської мови від західних видавництв на кшталт Oxford University Press, Macmillan, Cambridge University Press, а також визначила формат відомих міжнародних іспитів на знання англійської мови, таких як IELTS і TOEFL.

Лексичний підхід (Lexical approach), який виник близько 30 років тому, багато в чому спирається на комунікативну методику, але інакше розставляє пріоритети. велика увага приділяється вивченню лексики. Підхід ґрунтується на двох важливих спостереженнях. По-перше, слова існують не поодиночі, а у вигляді стійких виразів, які сприймаються як єдине ціле (наприклад, "Доброго ранку!", "Як справи?", "Не могли б ви..."). По-друге, більшість наших розмов рідною мовою складається з таких виразів, а граматику має другорядну роль. На уроках, що наслідують лексичний підхід, студенти навчаються розпізнавати та використовувати частотні словосполучення. Зазначена методика досить часто застосовується при підготовці до міжнародних іспитів, оскільки дозволяє за відносно короткий проміжок часу істотно розширити словниковий запас і зробити мову більш різноманітною [3, 5].

Метод навчання з урахуванням завдань (Task-based learning) – є варіацією комунікативного підходу. Якщо на класичних уроках учні спочатку дізнаються правило, а потім вчать його застосовувати, то метод навчання на основі завдань йде від зворотного: спочатку вам дають завдання, а вже в процесі його виконання вчитель допомагає освоїти лексику і граматику, яких бракує. При навчанні за цим методом як завдання використовується групові проекти (наприклад, вас можуть попросити вибрати 5 головних подій року та зробити презентацію для

всього класу). Такий підхід часто виявляється більш ефективним, ніж традиційний, оскільки учні від початку розуміють, як можна застосувати нові знання, і матеріал запам'ятовується швидше та легше. Крім того, робота над груповими проєктами може бути дуже цікавою. Переважна більшість сучасних комунікативних підручників включають уроки, побудовані за принципом навчання на основі завдань [5].

Для досягнення кращих результатів у більшості методик з вчителем пропонується використовувати додаткові інструменти для вивчення англійської мови.

Мабуть, найпоширеніша сьогодні доповнююча методика – самонавчання (Self-Study). Зараз існує безліч сайтів та додатків, які дають нові слова, пояснюють правила граматики, пропонують безліч вправ на відпрацювання, організовують чат з іншими учнями, записують зразки вимови та багато іншого. Завдання в таких системах організовані як короткі відрізки, які можна швидко проходити, отримуючи заохочення (бали чи рівень). Заохочення створює приємний ефект для нашого мозку, і викликають бажання виконувати нові завдання. Хоча цей метод не дає такого приросту знань, як робота з викладачем, однак він є безцінним для домашньої роботи, для повторення і просто як спосіб забезпечити більш частий контакт з мовою, що вивчається [6].

## **1.2. Порівняльний аналіз програм для вивчення англійської мови**

Існує велика кількість різноманітних програмних додатків для вивчення англійської мови. Їх можна розділити на такі групи:

1. Лексичні тренажери – для задовільного сприйняття нетехнічних текстів англійською мовою, на думку вчених Б. Лауфера та І. Нейшн, люди, що вивчають іноземну мову, повинні знати принаймні від 5000 лексичних одиниць [7, 8]. Тому такі програми можна використовувати для

відпрацювання лексичного матеріалу. Основним недоліком таких програм є часта відсутність контексту у вживанні слів та рекомендацій щодо вживання даного слова.

2. Граматичні тренажери. Граматика є невід'ємним аспектом мови, можливість вивчати граматичні конструкції та покращувати їх застосування на практиці можна також здійснювати за допомогою комп'ютера. На даний момент спеціальні програми надають можливість вивчити іноземну мову, маючи практичні способи визначення помилок та їх виправлення у тексті.

3. Додатки для розвитку навичок читання – нові технології також торкнулися і процесу навчання читання, зробивши його більш інтерактивним і доступним для тих, хто вивчає іноземну мову.

4. Програми для розвитку навичок аудіювання. На даний момент популярність завойовує відеозаписи та подкасти. Подкасти – це звукові файли в стилі радіотелепередач в Інтернеті, які забезпечують легкий та зручний доступ до навчальних матеріалів, як в освітній установі, так і за її межами.

5. Програми, спрямовані на розвиток навичок говоріння. У даний час вивчаючи іноземну мову можуть з легкістю зануритися у віртуальне середовище спілкування іноземною мовою через програмні додатки.

Розглянемо декілька найбільш популярних додатків, які використовуються для самостійного вивчення англійської мови.

Memrise – це програма, що працює на принципі словникових карток (рис. 1.1). Вона використовує повторення з інтервалами, щоб нові слова і фрази "закріпилися" у довгостроковій пам'яті. Багато карток створено користувачами, тому можна вибрати різну тематику. Для платних користувачів також є функція розпізнавання мовлення, яка дозволяє відпрацьовувати вимову, і колекція відео з носіями мови, які вимовляють слова чи фрази під назвою "навчайтеся з місцевими". Програма використовується для збільшення словникового запасу [9].

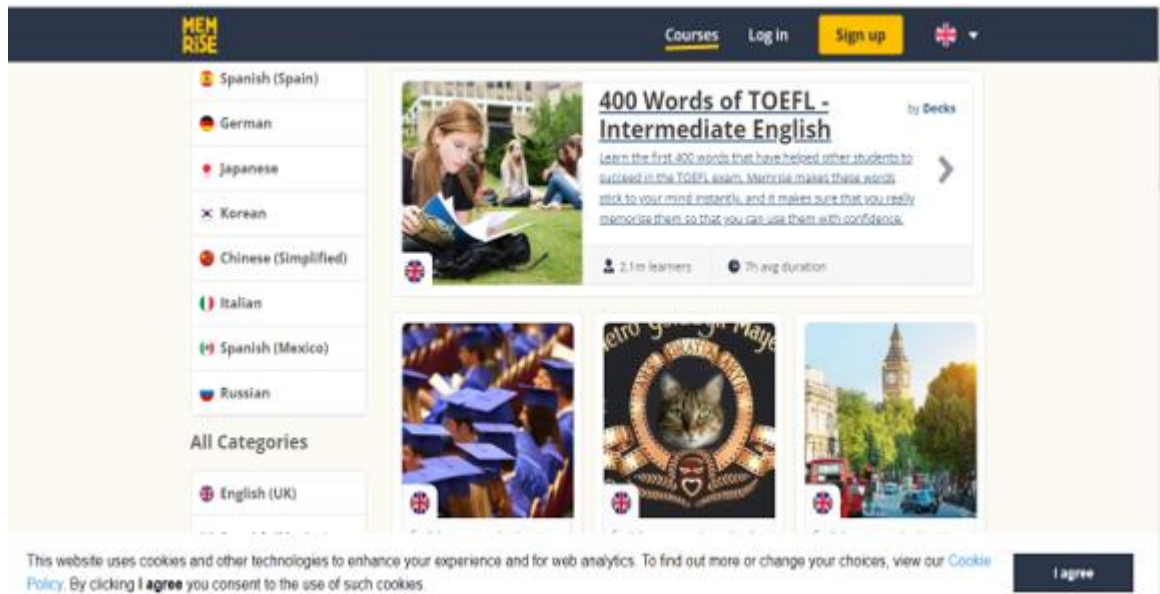


Рисунок 1.1. Сайт додатку Memrise

Додаток Babbel працює на підписній основі (рис. 1.2). У ньому існують курси для тих, хто вивчає англійську будь-якого рівня, і всі вони доступні на комп'ютері та мобільному пристрої. Уроки тривають 20 хвилин і містять поєднання розмовної практики (з автоматичним розпізнаванням голосу), практики граматики та запам'ятовування нових речень. Також є ефективний розділ карток, щоб запам'ятовувати речення та слова з інших уроків [10].

Babbel є корисним інструментом для тих, хто вивчає англійську мову, особливо для тих, кому потрібна додаткова практика вивчення правил граматики. Тут витримано гарний баланс між веселими, схожими на ігри програмами для вивчення англійської мови та суворими навчальними курсами. Однак найкраще це доповнювати живою розмовною практикою, щоб студенти могли разом використовувати свої нові навички.



Рисунок 1.2. Головна сторінка Babel для українських користувачів

Duolingo є найпопулярнішим додатком для вивчення мов [11]. Він надає користувачам 5-хвилинні ігри для вивчення англійських слів і трохи граматики. Здебільшого ігри передбачають переклад слова на англійську або з англійської мови або промовляння англійської фрази в мікрофон (рис. 1.3).

У Duolingo передбачена ціла система нагадувань про пропуск уроків, тому вона чудово підходить для формування постійної звички щоденного навчання. Але методологія програми в першу чергу базується на запам'ятовуванні фраз. Це означає, що учням, ймовірно, доведеться звернутися до інших методів, щоб зрозуміти структуру англійської мови.

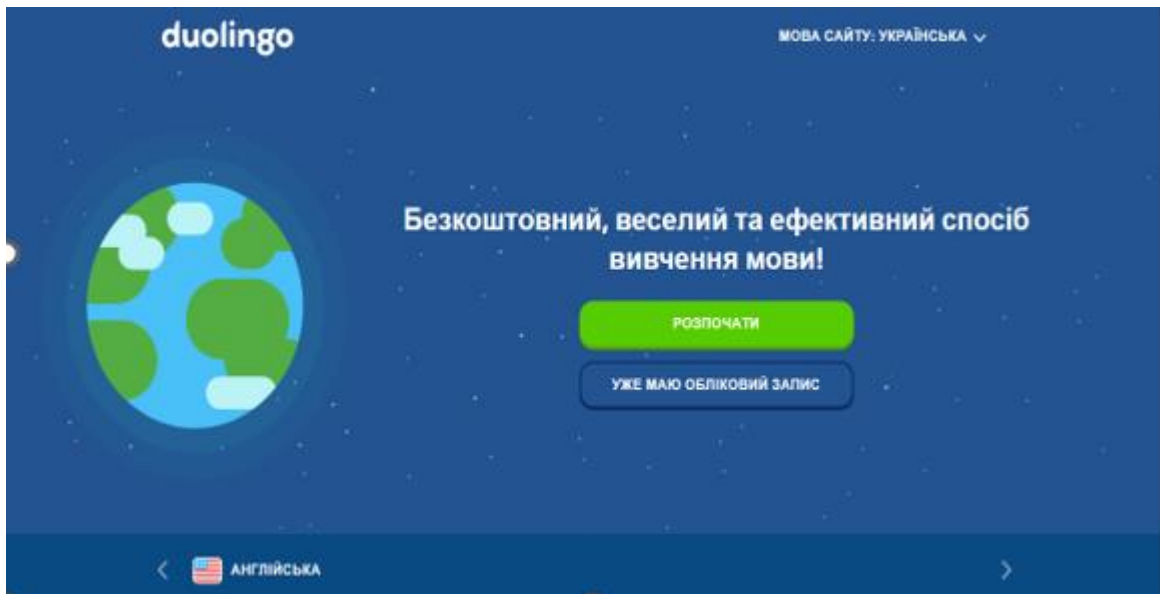


Рисунок 1.3. Головна сторінка Duolingo

Busuu має безкоштовну та платну версії. Безкоштовна версія Busuu є програмою, орієнтованою на словникові картки із записаними діалогами. Існує інструмент розпізнавання мовлення, за допомогою якого можна навчитися вимовляти англійські фрази. Уроки тривають десять хвилин [12].

Преміум-учасники отримують набагато більше переваг. Є уроки граматики для поглибленого рівня, а зміст адаптовано відповідно до власних причин вивчення англійської мови. Можна також скласти власний навчальний план і вибрати, коли додаток має нагадувати про навчання. Існує також форум для пошуку партнерів для мовного обміну. Busuu чудово підходить для того, щоб навчитися говорити базові англійські речення вголос (рис. 1.4).



Рисунок 1.4. Головна сторінка Busuu

Всі розглянуті програми є універсальними. Значна частина їх лексики абсолютно не пов'язана з ІТ-сферою. Приклади речень, які є в цих системах, у більшості торкаються тем загальної практики. У той же час у сфері ІТ існують усталені терміни, обороти та фрази, які необхідно знати, і, які потрібно знати фахівцю.

### 1.3. Постановка завдання

Проведений аналіз показав, що основним методом, який застосовується при вивченні англійської мови, є метод словникових карток. Даний метод є ефективним при вивченні основних термінів та фраз.

Програма, що розроблюється, повинна бути спрямована на розширення словникового запасу користувача та розвитку розуміння професійних текстів. Для цього пропонується використовувати метод словникових карток. У програмі користувачу показують нові слова з перекладом. Користувач може переглядати слова, додавати ті слова, які він зустрічає в професійних текстах. Колекції користувач може формувати



самостійно, додаючи нові слова. Таким чином, користувач отримує можливість вибудувати свою стратегію вивчення слів та фраз.

Для закріплення успіху користувач повинен мати можливість протестувати свої знання, виконуючи завдання. Оскільки програма не є контролюючим засобом, завдання повинні давати можливість користувачу отримати правильну відповідь.

У програмі потрібно передбачити два види завдань:

завдання першого типу – користувачу пропонується слово і декілька варіантів перекладу, серед яких він повинен вибрати правильний. Таким чином, відпрацьовується процес запам'ятовування слів. Словниковий запас користувача розширюється;

завдання другого типу – користувачу пропонуються фрази, серед яких він повинен обрати правильний переклад.

Фрази на розуміння готуються або користувачем, або викладачем, якщо він є. Такий підхід дає можливість користувачу зосередити зусилля на фразах та ідіомах, які він зустрічає в професійній літературі.

Всі додатки, що були розглянуті, є або веб-додатками, або мобільними додатками. Однак часто фахівцю потрібно працювати саме за комп'ютером, тому іноді потрібно використовувати саме десктопний додаток.

Отже, необхідно створити додаток, який повинен:

- надавати користувачу можливість вивчати і слова, і фрази з ІТ-сфери;
- бути гнучким, зручним для користувачів.

Додаток повинен бути десктопним, розробленим з максимальною простотою для використання користувачами.

При реалізації і використанні додатку повинні бути враховані вимоги до функціональних характеристик, параметрів технічних засобів, інформаційної та програмної сумісності.

*Загальна характеристика призначеного для користувача інтерфейсу.*

У роботі додатку повинно бути виділено основне завдання: вивчення основних термінів в області ІТ англійською мовою. Відповідно до цього в інтерфейсі повинна бути передбачена можливість для показу англійських слів. У програмі потрібно передбачити збереження прогресу користувача у вивчення слів та фраз.

## РОЗДІЛ 2

### ПРОЄКТУВАННЯ ДОДАТКУ З ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ

#### 2.1. Визначення функціональних та нефункціональних вимог до додатку

Функціональні вимоги визначають функціональність ПЗ, яку потрібно побудувати, щоб користувачі змогли виконати свої завдання. На основі проведеного дослідження та постановки задачі сформуємо вимоги до розроблювального проєкту:

- 1) додаток має на меті створювати та редагувати колекції слів;
- 2) додаток має створювати завдання для колекції слів;
- 3) додаток повинен перевіряти відповіді на завдання;
- 4) додаток повинен відслідковувати прогрес користувача у вивченні слів та фраз.

Нефункціональні вимоги описують властивості та обмеження, що накладаються на систему. Для реалізації програми були виявлені такі вимоги:

При розробці програми для підвищення її якості з точки зору юзабіліті потрібно користуватися правилом "Чим простіше, тим краще".

Дизайн програми визначається такими вимогами:

– простота і мінімалізм – інформація для користувача представляється в структурованому вигляді і тільки в тому обсязі, який необхідний на даному етапі;

– доступність – основна інформація і функції програми повинні бути доступними для користувача;

– актуальність – дизайн програми не повинен рясніти графічними ефектами, щоб не відволікати користувачів від вивчення слів.

Зручність навігації.

Простота навігації для будь-якої програми є важливим критерієм. Користувач повинен інтуїтивно і безперешкодно переміщатися по програмі, отримувати необхідну інформацію, мати можливість виконувати потрібні для нього дії.

## 2.2. Розробка проєкту додатку з вивчення англійської мови

Перейдемо до проєктування системи. Для цього пропонується використовувати об'єктно-орієнтований підхід.

Об'єктно-орієнтоване проєктування (ООП) – це методологія розробки програмного забезпечення, що базується на концепції об'єктів та їх взаємодії. Цей підхід спрощує процес розробки, робить код більш зрозумілим та зручним для обслуговування, а також забезпечує більшу гнучкість та повторне використання. Крім того, об'єктно-орієнтоване проєктування полегшує використання коду через можливість повторного використання класів та методів.

Під час аналізу предметної області був виділений один актор у системі – користувач програми. Він може:

- 1) управляти колекціями слів, створюючи нові, редагуючи існуючі;
- 2) вивчати слова та фрази;
- 3) виконувати завдання.

Використовуючи діаграму варіантів використання, визначимо основні способи використання додатка (рис. 2.1).

Опис основних варіантів використання наведений у табл. 2.1-2.5.

Таблиця 2.1

### Варіант використання "Керування колекцією"

Короткий опис	Варіант використання дозволяє керувати колекцією
Актори	Користувач
Передумова	Користувач запустив програму
Основний потік	Користувач натиснув кнопку "Управління колекцією" та

	перейшов на форму "Управління колекцією"
--	--

Таблиця 2.2

## Варіант використання "Створення колекції"

<b>Короткий опис</b>	<b>Варіант використання дозволяє створювати колекцією</b>
Актори	Користувач
Передумова	Користувач перейшов у вікно "Керування колекцією"
Основний потік	Користувач вводить ім'я нової колекції та натискає кнопку "Створити"
Альтернативний потік	Колекція не створюється, бо відсутній файл на диску

Таблиця 2.3

## Варіант використання "Редагування колекції"

<b>Короткий опис</b>	<b>Варіант використання дозволяє редагувати колекцією</b>
Актори	Користувач
Передумова	Користувач перейшов у вікно "Керування колекцією"
Основний потік	Користувач натискає кнопку "Редагувати", обирає колекцію, вводить слово, переклад, приклад та натискає кнопку "Створити"

Таблиця 2.4

## Варіант використання "Вивчення слів та фраз"

<b>Короткий опис</b>	<b>Варіант використання дозволяє вивчати слова</b>
Актори	Користувач
Передумова	Користувач перейшов у вікно "Вивчення слів"
Основний потік	Користувач натискає кнопку "Вчити", обирає колекцію, на екрані з'являються слова з поясненнями

Таблиця 2.5

## Варіант використання "Виконання завдання"

<b>Короткий опис</b>	<b>Варіант використання дозволяє вивчати слова</b>
Актори	Користувач
Передумова	Користувач перейшов у вікно "Тестування"
Основний потік	Користувач натискає кнопку "Тест"

	Користувач обирає колекцію Користувач обирає тип завдання
--	--

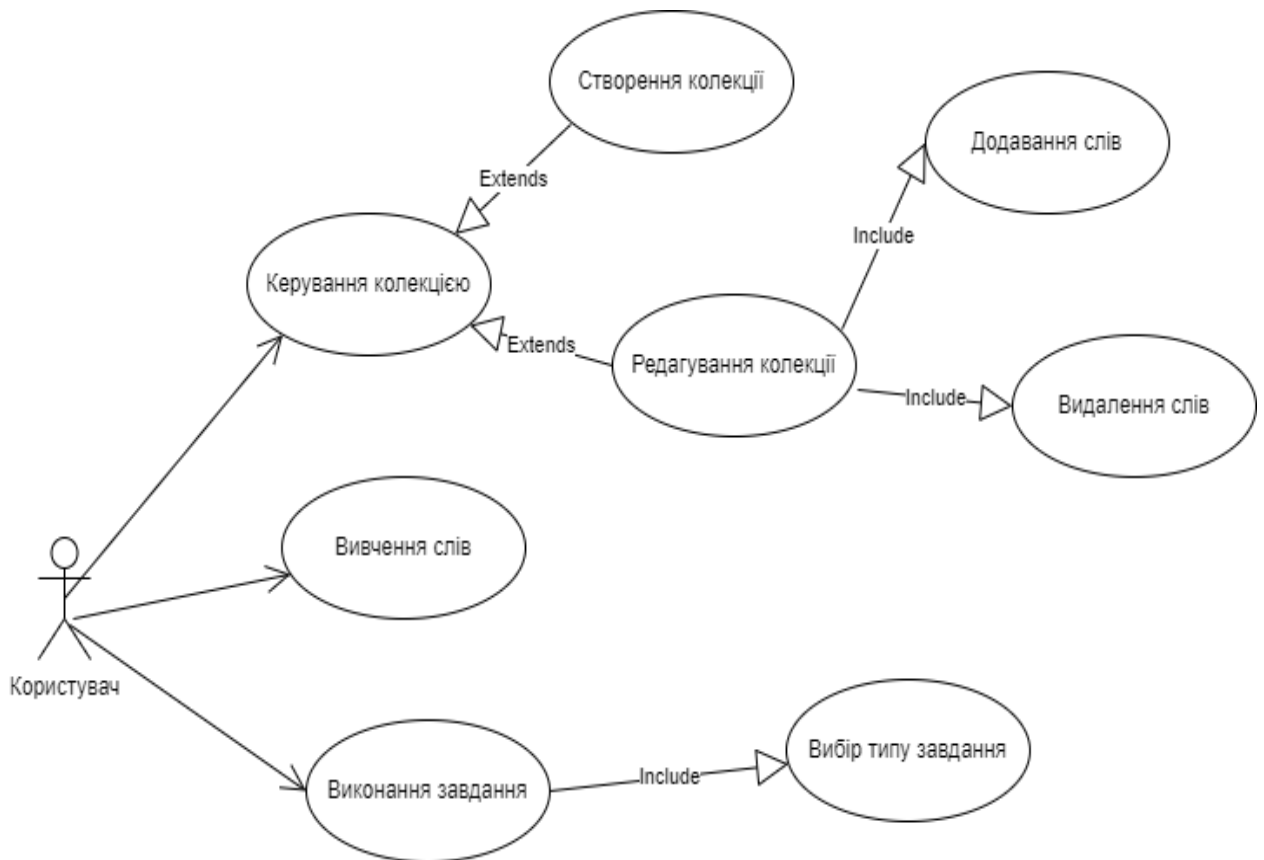


Рисунок 2.1. Діаграма варіантів використання розроблювального додатка

Послідовність виконання операцій у варіанті використання "Тестування" показана на діаграмі послідовностей (рис. 2.2).

Користувач обирає режим "Тестування".

Інтерфейсний модуль передає управління модулю тестування.

Модуль тестування надсилає запит до БД та з отриманих результатів запиту формує завдання.

Користувач отримує завдання, дає відповідь та надсилає на перевірку модулю тестування.

Модуль тестування оброблює результати та надсилає їх користувачу.

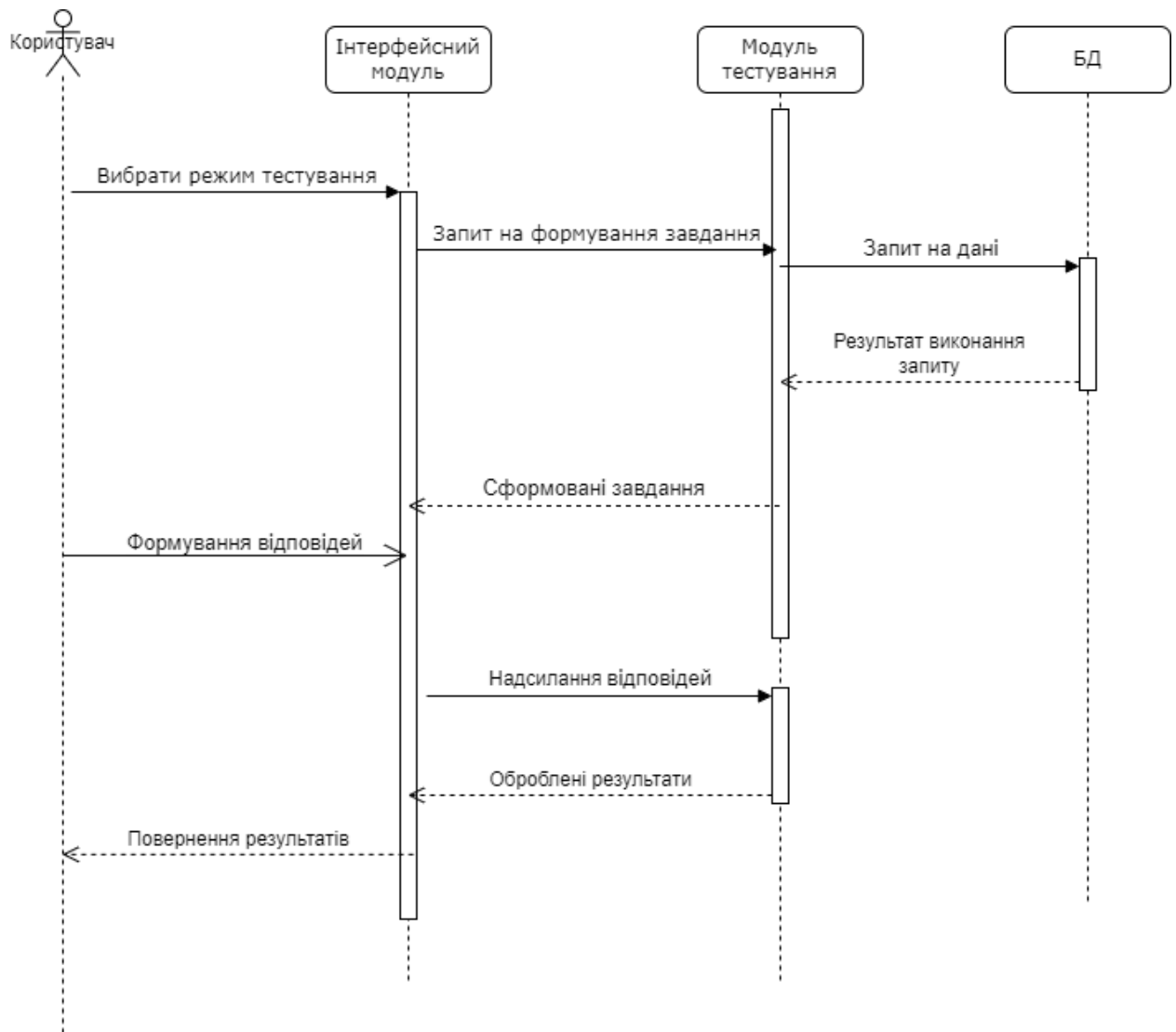


Рисунок 2.2. Діаграма послідовностей для варіанта використання "Тестування"

Для опису формування колекції слів використаємо діаграму активностей (рис. 2.3).

При управлінні колекцією слів користувач повинен мати можливість додавати та видаляти слова з колекції в залежності від своїх потреб.

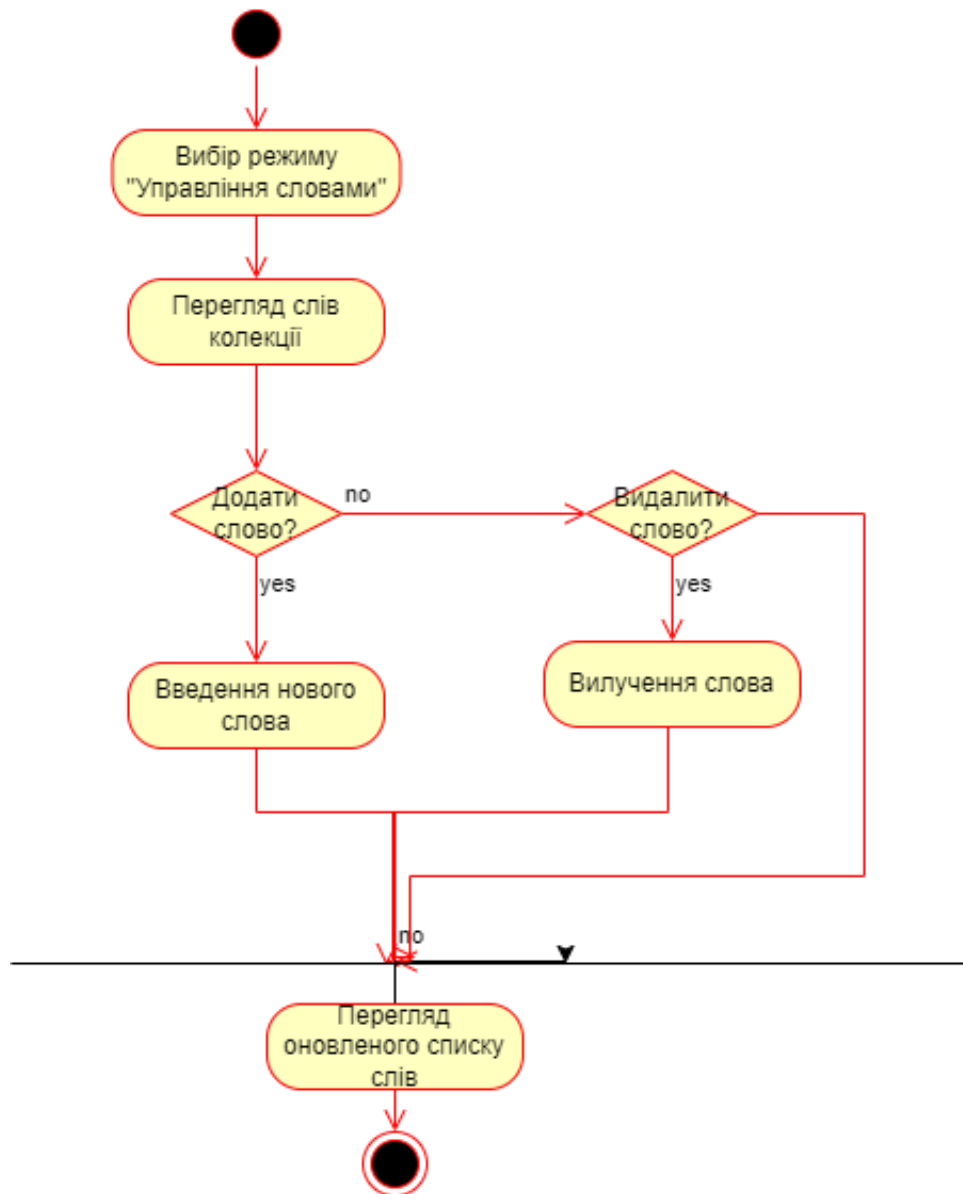


Рисунок 2.3. Діаграма активностей при управлінні колекцією слів

Програму вивчення англійської мови пропонується створювати у вигляді десктопного додатку, де створювана користувачем колекція слів буде зберігатися у БД. Щодо фраз, то пропонується використовувати текстовий файл, оскільки це забезпечить для користувача більш зручний спосіб його редагування. Загалом архітектура системи представлена на рис. 2.4.



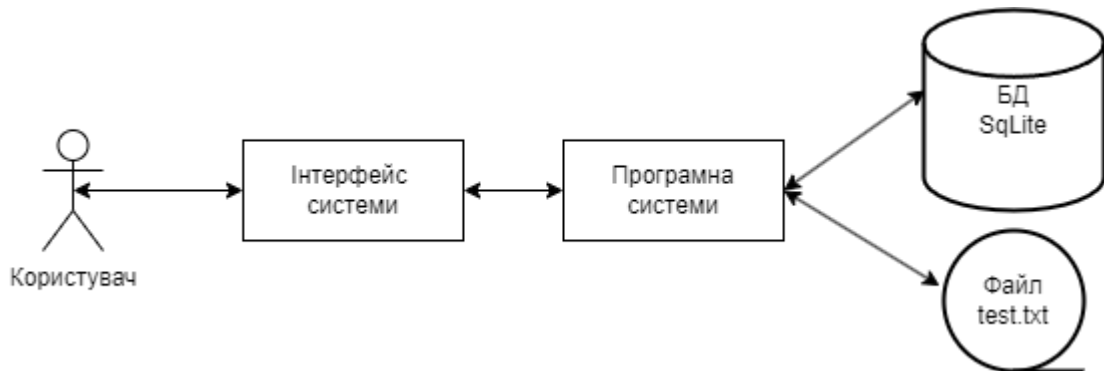


Рисунок 2.4. Узагальнена архітектура системи

Використання запропонованої архітектури дозволяє, з одного боку, забезпечити колекції слів користувача від ненавмисного втручання, а з іншого боку, дасть змогу досить ефективно визначити складні фрази професійної англійської мови для завдань.

### 2.3. Обґрунтування вибору засобів розробки

Для реалізації розробленої програми було вирішено використовувати мову C# і платформу NET [13-16]. Дана платформа орієнтована на розробку під Windows.

Платформа має потужний інструментарій, представлений у вигляді бібліотеки базових класів, що поставляється разом із середовищем, володіє достатнім функціоналом для вирішення завдань практично будь-якої складності [14].

Платформа підтримує мови високого рівня, що дає змогу зручно та швидко писати читабельний код, що дуже важливо задля подальшої підтримки програми.

Visual Studio – середовище розробки, що постачається разом із .NET, надає необхідний інструментарій для ефективного та швидкого створення програм з графічним інтерфейсом.

Мова програмування C# була створена фірмою Microsoft спеціально для .NET. Зв'язку C# і .NET активно застосовують для створення додатків,

які повинні встановлюватися на машині кінцевого користувача, де і виконуватиметься вся обробка даних. Розробку таких програм забезпечує бібліотека Windows Forms, що дозволяє проєктувати графічний інтерфейс.

Обґрунтування вибору мови програмування:

- мова програмування C# є дійсно об'єктно-орієнтованою мовою;
- мова програмування C# покликана реалізувати компонентно-орієнтований підхід до програмування, який сприяє меншій машинно-архітектурній залежності результуючого програмного коду, більшої гнучкості, переносимості та легкості повторного використання програм;
- початкова орієнтація на безпеку коду;
- розширена підтримка подієво-орієнтованого програмування;
- мова програмування C# є «рідною» для створення додатків у середовищі Microsoft .NET, оскільки найбільш тісно та ефективно інтегрована з нею.

Windows Forms – інтерфейс програмування додатків (API), який відповідає за графічний інтерфейс користувача і є частиною Microsoft .NET Framework. Цей інтерфейс спрощує доступ до елементів інтерфейсу Microsoft Windows. Перевагою є те, що керований код — класи реалізують API для Windows Forms, незалежно від мови розробки.

Однак у технології NET є деякі недоліки, наведемо їх перелік:

1. Об'єктно-реляційна підтримка, яка включається через Entity Framework. Вважається, що гнучкість Entity Framework обмежена, оскільки вона не підтримує деякі нові дизайни баз даних.

2. Ціноутворення. Так, ліцензія Visual Studio може коштувати від 799 до 5999 доларів США на рік, залежно від року підписки (перша завжди дорожче, ніж поновлення) і обраний вами пакет. При виборі Windows для розробки необхідно також сплатити ліцензійний збір.

3. Проблеми стабільності для нових випусків. Існують певні проблеми стабільної роботи нових випусків, що може утруднювати та сповільнювати роботу.

Однак у будь-якому разі, для десктопних розробок під Windows – це найкращий вибір.

## РОЗДІЛ 3

### РОЗРОБКА ДОДАТКУ З ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ

#### 3.1. Проектування інтерфейсу користувача

Дизайн інтерфейсу користувача (User Interface UI) – це процес створення візуальних та інтерактивних елементів програмного додатку, який бачать користувачі та з якими взаємодіють. Дизайн інтерфейсу користувача є ключовим аспектом користувацького досвіду (UX), тобто загального задоволення та простоти використання, які користувачі відчувають під час використання продукту.

У розробці інтерфейсу користувача для десктопної програми необхідно дотримуватися деяких загальних принципів дизайну інтерфейсу користувача, які допоможуть створити чіткий, послідовний та інтуїтивно зрозумілий інтерфейс.

Видимість є ключовою; необхідно переконатися, що найважливіші та часто використовувані функції є видимими та доступними для користувачів.

Крім того, дії користувача потрібно супроводжувати своєчасним і відповідним відгуком.

Візуальні підказки та індикатори повинні підказувати, як користувачі можуть взаємодіяти з елементами інтерфейсу, такими як кнопки, меню, піктограми та повзунки.

Вирівнювання елементів інтерфейсу має здійснюватись логічним і послідовним способом, який відповідає природним моделям читання та рухам очей користувачів.

Необхідно уникати непотрібної складності та безладу в інтерфейсі, використовувати чітку та стислу мову та мітки.

Під час розробки інтерфейсу користувача для десктопної програми необхідно вибрати правильні компоненти інтерфейсу користувача, які

відповідають функціональності та призначенню. Компоненти (кнопки, меню, діалогові вікна, вкладки, панелі інструментів і вікна) мають бути сумісні з операційною системою та платформою десктопної програми та відповідати стандартним угодам і рекомендаціям системи.

Для розробки програми для вивчення англійської мови пропонується тип діалогових вікон, що буде містити стандартні елементи управління Windows Forms. За основні варіанти використання будуть відповідати окремі вікна, що дасть змогу користувачу легше орієнтуватися в роботі додатка.

Ескіз головного вікна системи зображено на рис. 3.1.

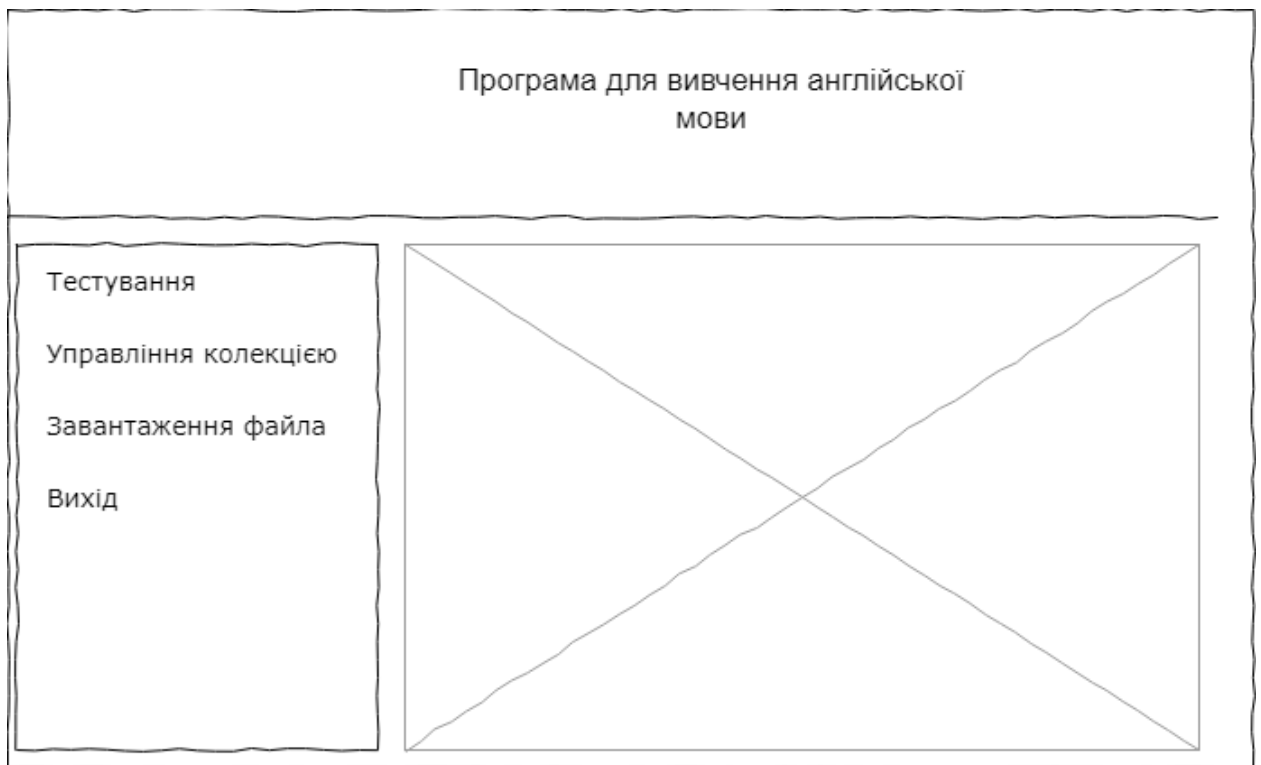


Рисунок 3.1. Головне вікно системи

Вікно "Тестування" містить у собі елементи, що пов'язані з проведенням тестування. За допомогою прапорців користувач може обрати варіанти завдання, переміщатися між окремими пунктами завдання (рис. 3.2).



Рисунок 3.2. Вікно тестування

Для управління колекціями було розроблено вікно, що дає змогу переглядати слова, шукати їх, додавати, редагувати та видаляти слова (рис. 3.3).

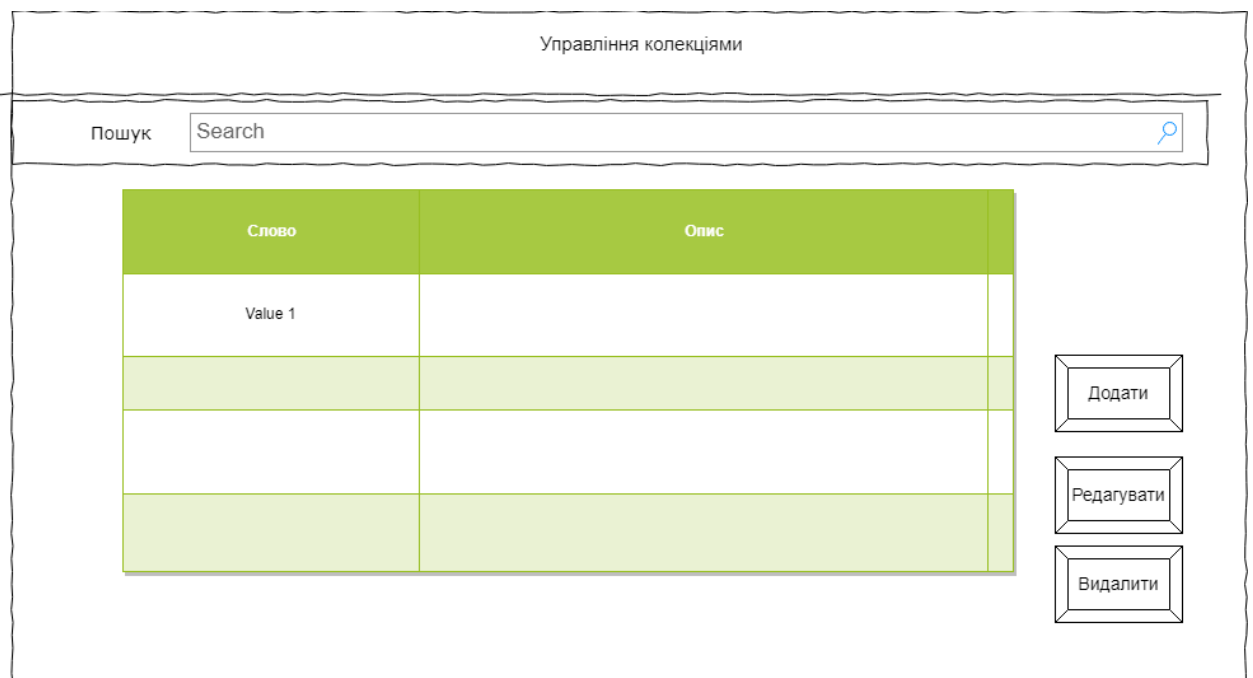


Рисунок 3.3. Вікно управління колекціями

Четверте вікно використовується для додавання слів до колекції (рис. 3.4). Воно дозволяє вводити слово та його опис (переклад).

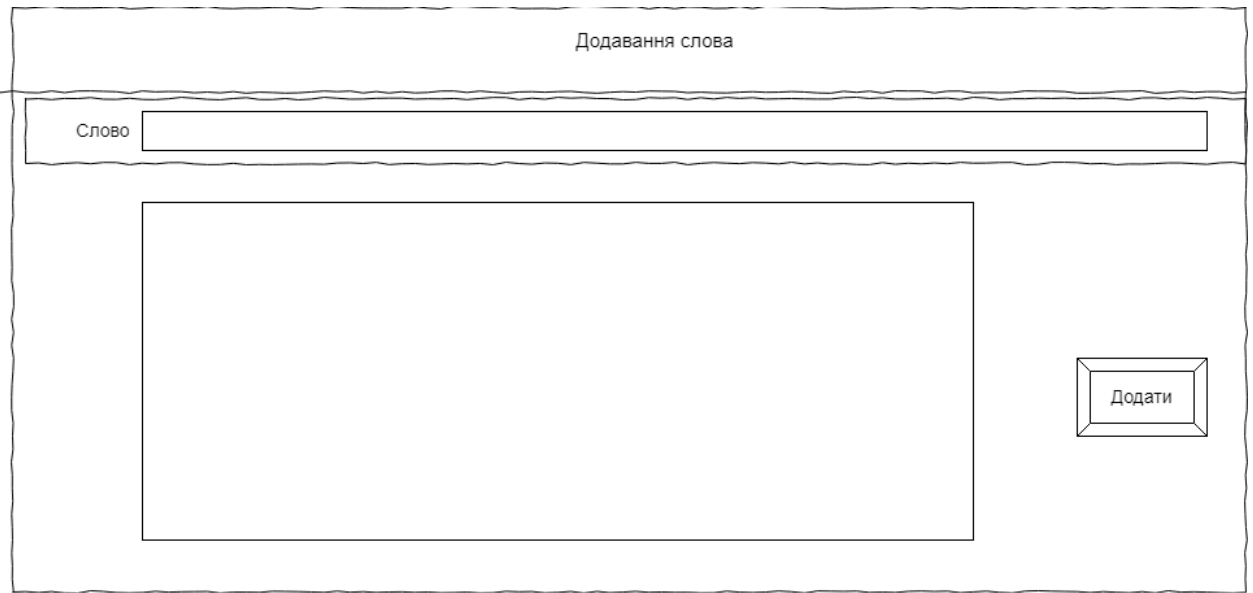


Рисунок 3.4. Вікно додавання слова

У програмі передбачається два режими тестування: тестування на знання слів та тестування на знання фраз. Для переходу до режиму тестування пропонується використовувати проміжне вікно, у якому користувач обирає потрібний режим (рис. 3.5)

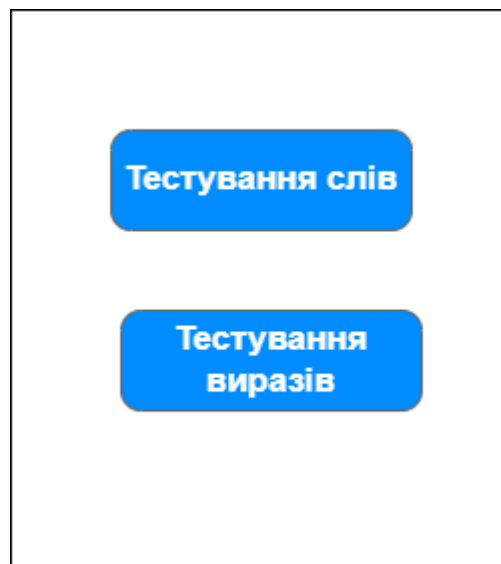


Рисунок 3.5. Вибір режиму тестування

У кожному з режимів тестування передбачена можливість за допомогою кнопок здійснювати навігацію по завданнях, та відображення прогресу користувача (кількості правильних відповідей).

Вікно режиму тестування слів наведено на рис. 3.6.

Тестування слів

Line 1

Опис слова

Прогрес

Відповідь

Назад

Вперед

Рисунок 3.6. Структура вікна для тестування на знання слів

Вікно для тестування на правильний переклад фраз наведено на рис. 3.7.

Тестування фраз

Фраза

Відповідь 1

Відповідь 2

Відповідь 3

Відповідь 4

Прогрес

Відповідь

Назад

Вперед

Рисунок 3.7. Структура вікна для тестування на знання фраз



Загалом ієрархія вікон зображена на рис. 3.8.

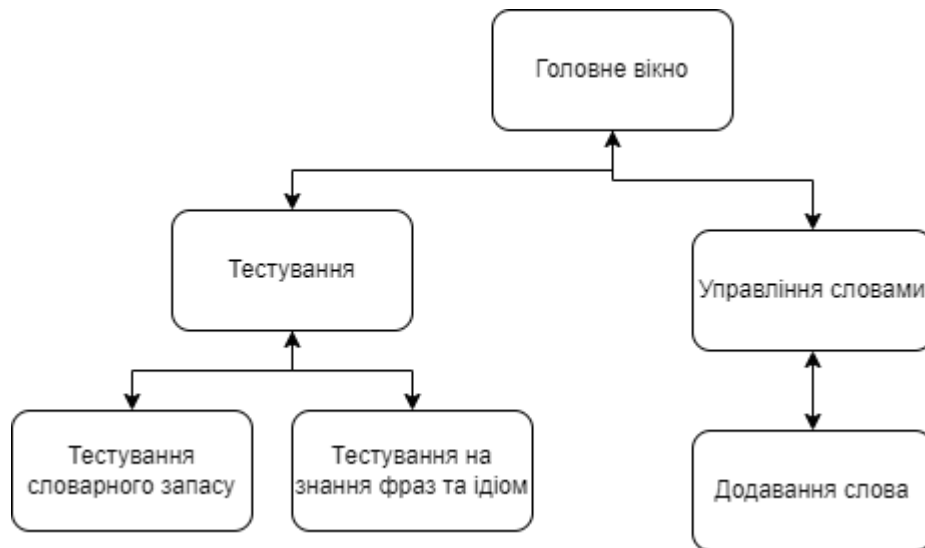


Рисунок 3.8. Ієрархія вікон програми

Розроблений інтерфейс є досить гнучким, інтуїтивно простим. У такому вигляді програма не буде потребувати додаткових пояснень і буде зручною в користуванні.

### 3.2. Створення системи з вивчення англійської мови

Перед створенням системи опишемо концептуальну ієрархію класів. Основним класом програми є клас WordData, який інкапсулює роботу з БД. Клас містить інформацію щодо підключення до БД та методи по роботі з нею: створення таблиці (якщо вона не існує), додавання, вилучення та редагування записів у БД. Також запропоновано використовувати ще два класи для організації тестових завдань. Загалом, оскільки кількість завдань може варіюватися в залежності від прогресу конкретного користувача, завдання зберігають у вигляді двох однозв'язного списків.

Загалом структура класів наведена на рис. 3.9.

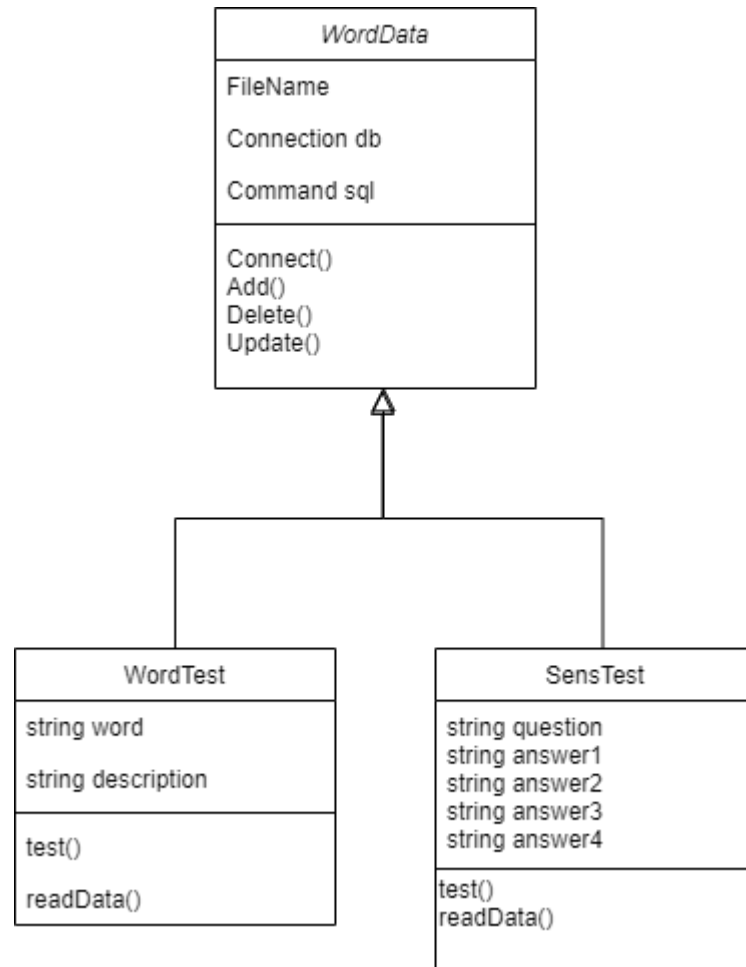


Рисунок 3.9. Узагальнена схема класів

На основі проведеного проектування був розроблений алгоритм роботи програми. Його узагальнена схема наведена на рис. 3.10. Оскільки ОС Windows є подієво-орієнтованою операційною системою, то розроблене програмне забезпечення є нелінійним відносно дій користувача і реагує на події, які викликаються його діями. При цьому програма представляє собою сукупність описів можливих сценаріїв обробки даних. Такий підхід має ряд переваг:

- інтуїтивна зрозумілість;
- близькість до предметної області;
- високий ступінь абстракції;
- можливість повторного використання коду.

Однак, у той же час, програма є досить складною для верифікації та тестування. А результати її роботи можуть бути несподіваними.

Використання бібліотеки Windows Forms дає можливість реалізувати цей підхід.

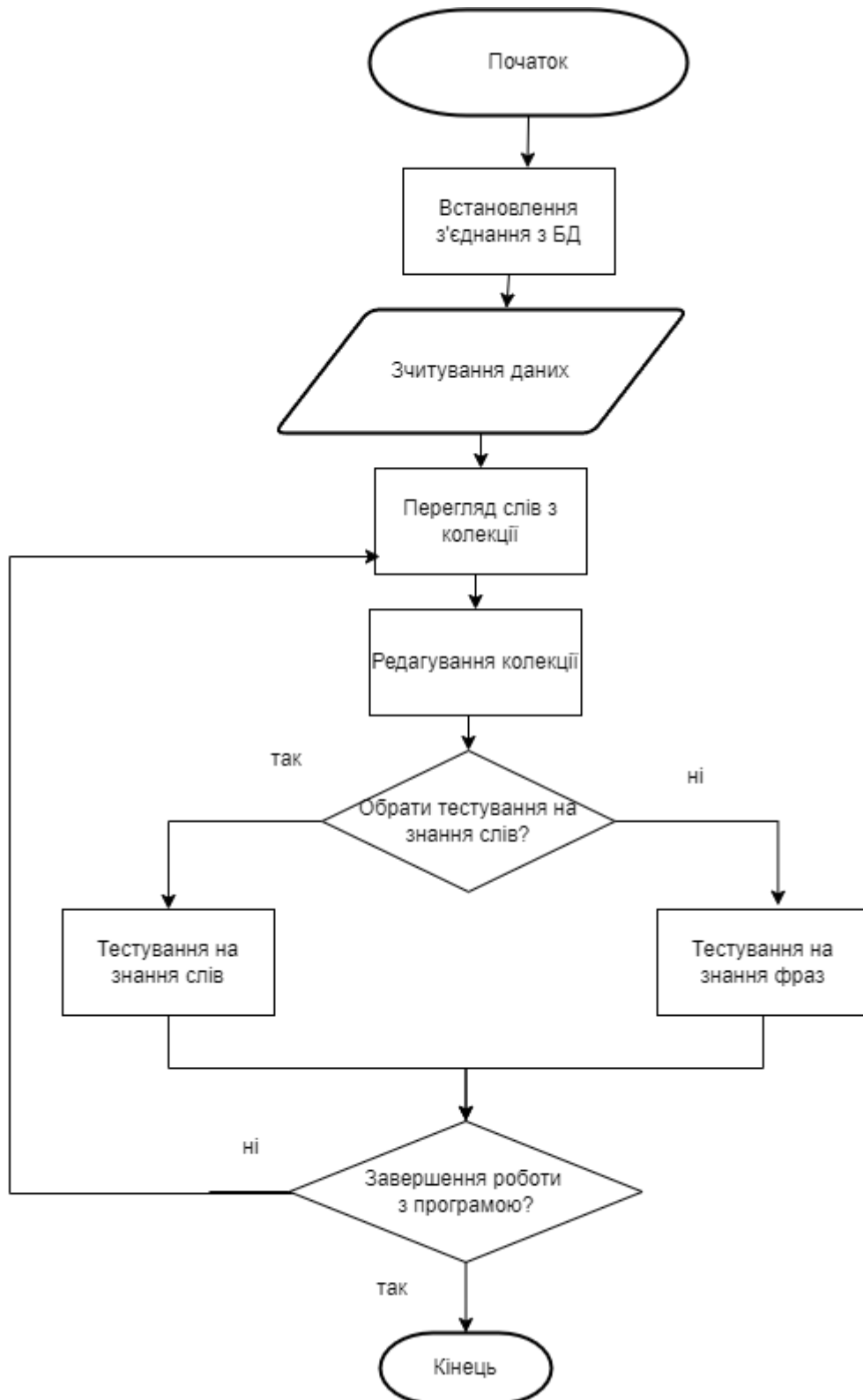


Рисунок 3.10. Узагальнена схема роботи з програмою

Опишемо процес створення програми з вивчення англійської мови.

Створення додатку здійснювалось у Visual Studio. Для цього створювався проєкт з такими параметрами: мова – C#, ОС – Windows, тип проєкту – Windows Forms).

Будь-який проєкт Windows Forms складається з таких компонентів:

Dependencies – вузол, який містить збірки dll, які додані до проєкту за замовчуванням. Збірки містять класи бібліотеки .NET, які використовуватиме C#.

*Form.Designer.cs*: файли, які містять визначення компонентів відповідної форми, доданих на форму у графічному дизайнері.

Файли форми *Form.cs*.

У нашому випадку в проєкті будуть присутні 6 форм (кожна форма міститься у своєму файлі).

Для кожної форми згідно розробленого проєкту інтерфейсу проводилось візуальне програмування. На форму поміщалися відповідні елементи управління, для яких встановлювались відповідні властивості. Використання Windows Forms у даному випадку дає можливість досить швидко та гнучко створювати програми з графічним інтерфейсом.

У Visual Studio передбачена можливість використання великої кількості різноманітних елементів управління. Для додавання елементів управління на форму потрібно на панелі елементів обрати необхідний елемент та перетягнути його на форму (рис. 3.11).

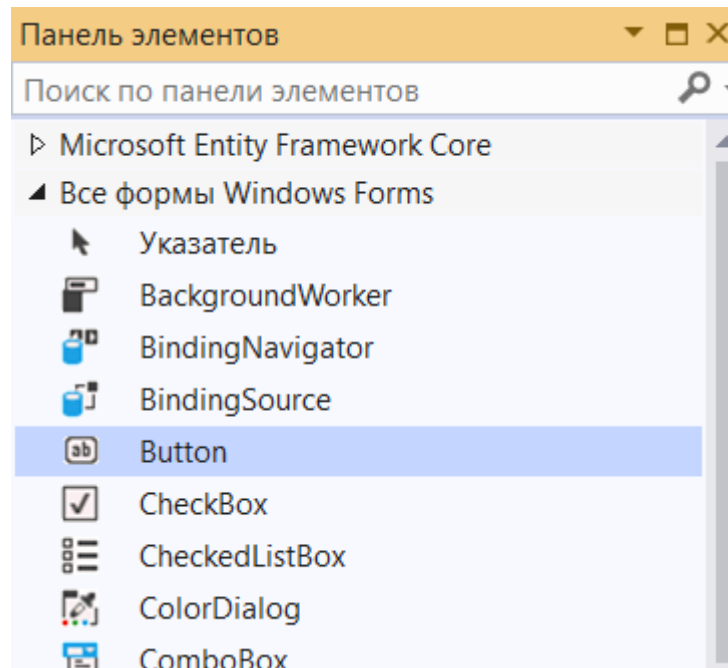


Рисунок 3.11. Панель елементів

Для кожного елемента управління у вікні властивостей встановлюється його зовнішній вигляд та поведінка (рис. 3.12).

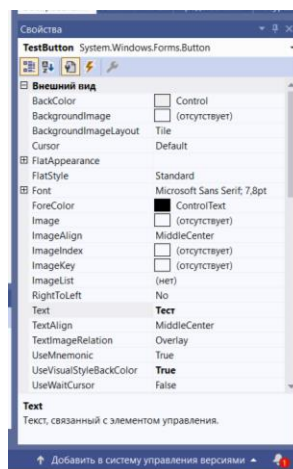


Рисунок 3.12 Вікно властивостей кнопки «Тест»

Після встановлення всіх основних елементів зовнішній вигляд головної форми має вигляд, наведений на рис. 3.13.

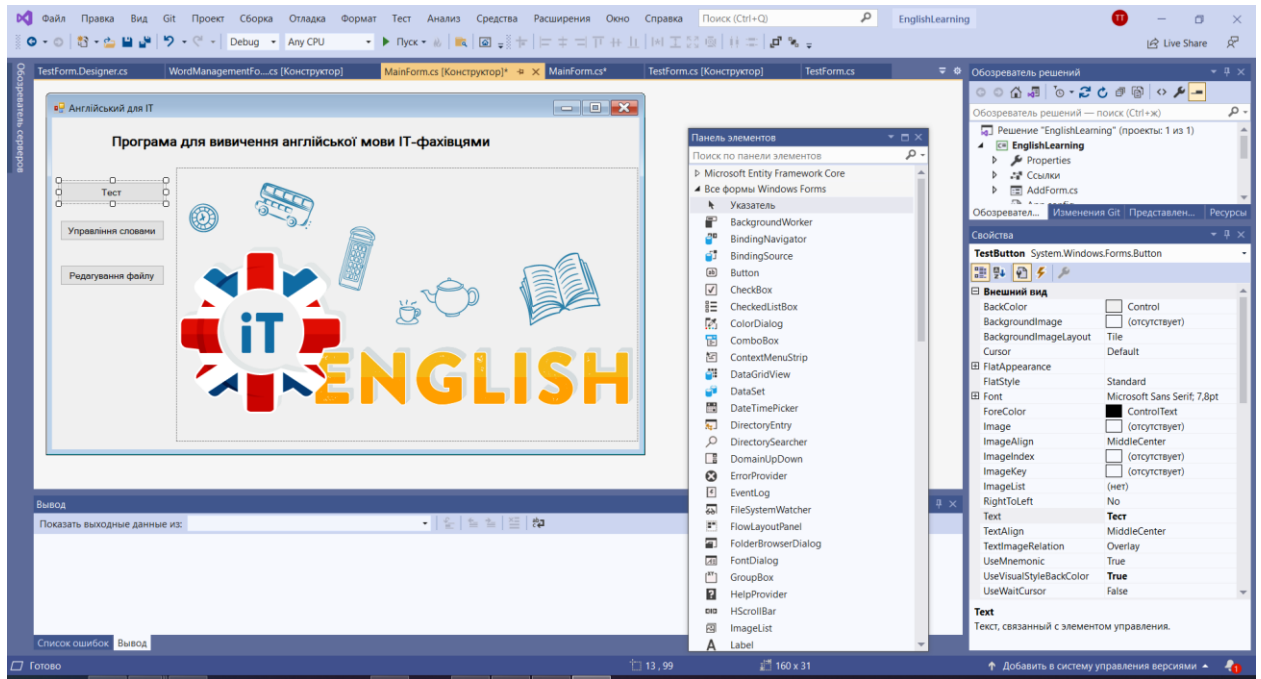


Рисунок 3.13. Головна форма у режимі конструктора

Для додавання обробників подій потрібно двічі клацнути мишею по відповідному елементу управління. Тоді середовище розробки переключить до файлу з кодом відповідного елемента, де й вводиться відповідний код (рис. 3.14).

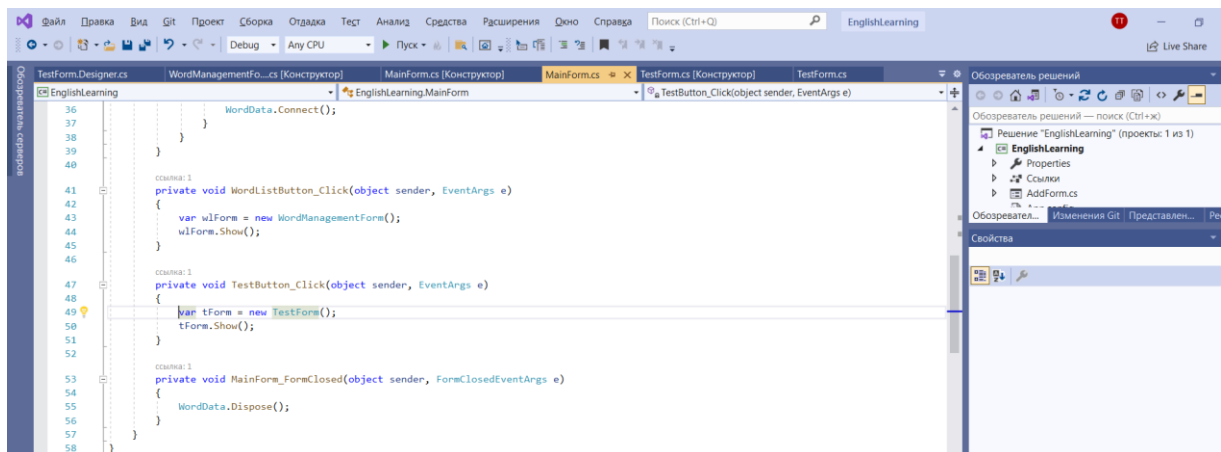
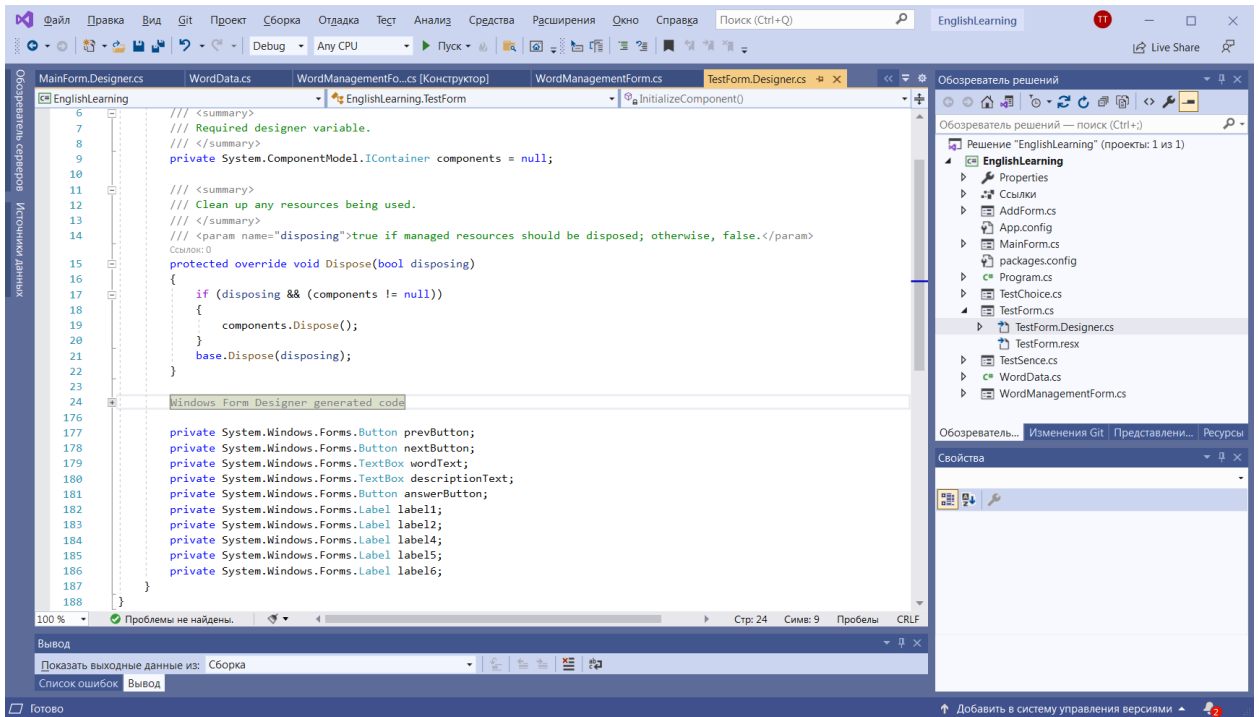


Рисунок 3.14. Код з функцією обробки події натискання кнопки «Тест»

Загалом лістинг файлів відповідних форм наведені в додатку.

Вікно для проведення тестування наведено на рис. 3.15.



Лістинг файлів WordManagmentForm.cs, WordManagmentForm.Designer.cs наведені в Додатку.

Розроблена форма використовується для додавання слів наведена на рис. 3.17.

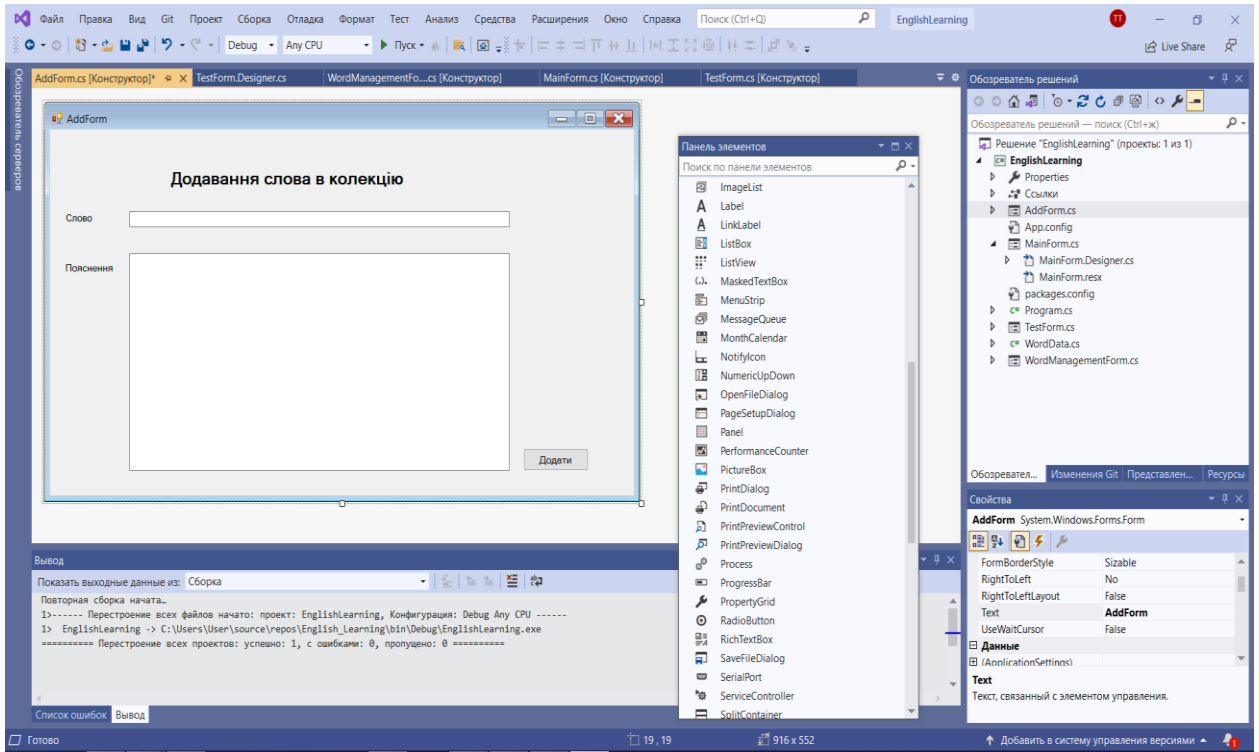


Рисунок 3.17. Макет форми для додавання слова

Лістинг файлів AddForm.cs, AddForm.Designer.cs наведені в додатку.

Форма для проведення тестування наведена на рис. 3.18.



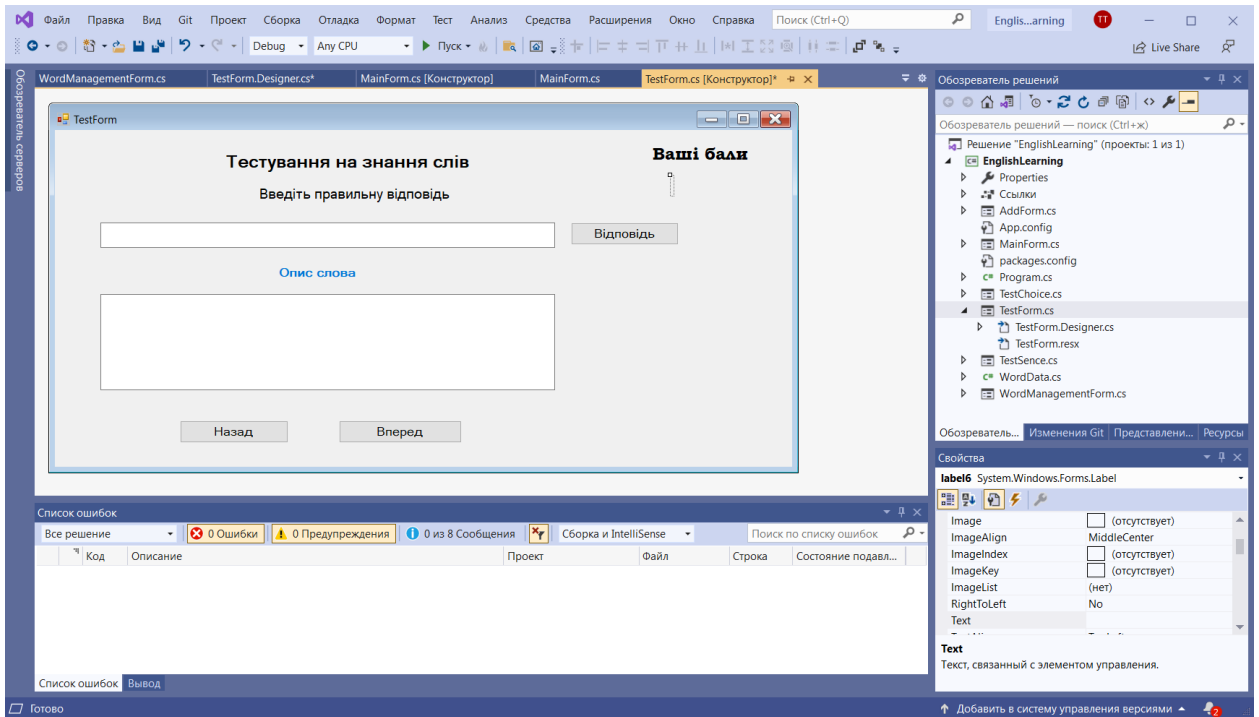


Рисунок 3.18. Форма для проведення тестування на знання слів

Лістинг коду TestForm.cs для проведення тестування наведений у додатку.

Форма для проведення тестування на знання фраз та ідіом наведена на рис. 3.19.

Розроблені візуальні форми містять усі необхідні компоненти, які визначались на етапі проєктування інтерфейсу.

Розроблена система діалогових вікон відповідає спроектованій моделі програмної системи з вивчення англійської мови.

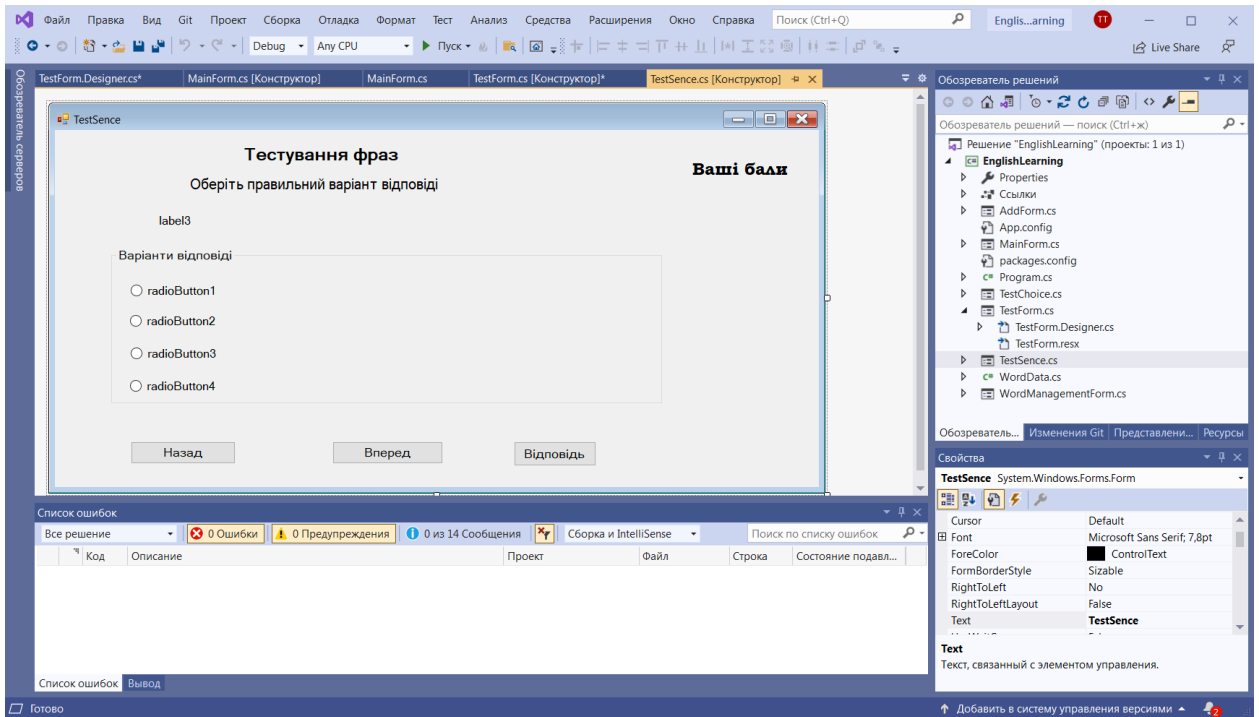


Рисунок 3.19. Форма для тестування на знання фраз

При реалізації тестування для підвищення ефективності навчання було вирішено виводити тестові завдання користувачу випадковим чином. Загальна структура розробленої системи наведена на рис. 3.20.

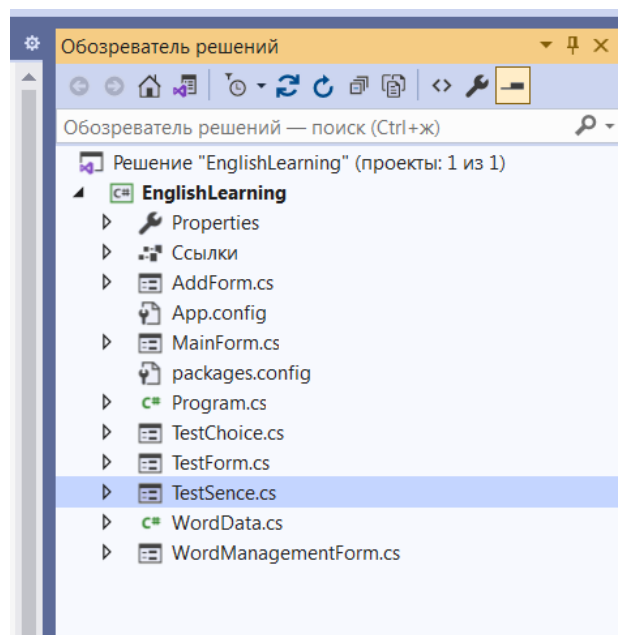


Рисунок 3.20. Структура розробленої системи

### 3.3. Тестування роботи програми

Результатом проведеного проектування та розробки став програмний продукт з вивчення англійської мови ІТ-фахівцями. Програма реалізована як десктопний додаток, що працює під управлінням ОС Windows.

Для запуску програми необхідно 1ГБ оперативної пам'яті (при запуску на віртуальній машині необхідно 1,5Гб пам'яті). На жорсткому диску для розгортання програми потрібно 500 Мб. Роздільна здатність монітора повинна становити 1024x768. Все це забезпечить для користувача комфортбельну роботу з програмою.

Програма використовує БД SQLite. У разі її відсутності програма автоматично створює її при запуску. БД містить слова та їх опис, які вивчає користувач. БД складається з трьох полів: id – ключове поле, word містить слово, description містить опис даного слова. БД знаходиться у файлі default.db. Користувач може підключити інший файл БД.

Для роботи з фразами програма потребує файл test.txt. Файл може створюватись у будь-якому текстовому редакторі, але мати визначений вигляд. Для кожного завдання відводиться 5 рядків: перший рядок – це питання (текст англійською або українською мовами, що містить якість стандартизовані лексеми або ідіоми), другий, третій, четвертий, п'ятий рядки варіанти відповідей (наприклад, переклади чи пояснення тексту завдання). Правильна відповідь повинна знаходитися у наступному після завдання рядку. Кількість завдань необмежена. Користувач може заповнювати вміст файла даними самостійно. Перевірка правильності фраз, записаних користувачем не здійснюється. Приклад файла з фразами наведений на рис. 3.21.

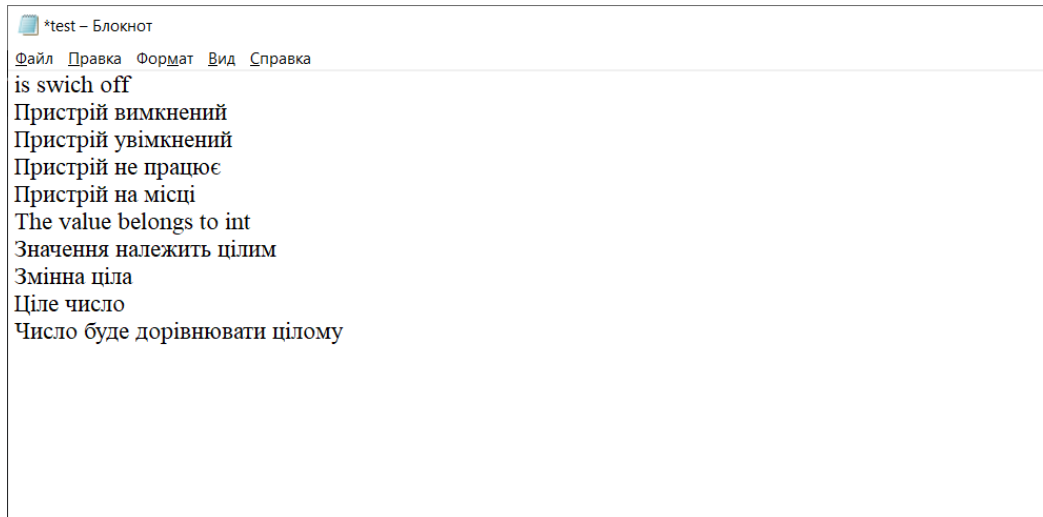


Рисунок 3.21. Приклад вмісту файлу test.txt

Для запуску програми використовується файл EnglishLearning.exe.  
Після запуску на екрані з'являється вікно програми (рис. 3.22).



Рисунок 3.22. Вікно роботи з програмою

Користувач може працювати з програмою в трьох режимах:

- управляти словами, які він вивчає,
- пройти тестування,

– змінити файл БД.

Для управління колекцією слів користувачу потрібно натиснути кнопку Управління словами. На екрані з'явиться вікно Управління словами, яке містить усі додані користувачем слова (рис. 3.23). Кожному слову відводиться один рядок таблиці: у лівій колонці знаходиться слово, у правій колонці опис слова.

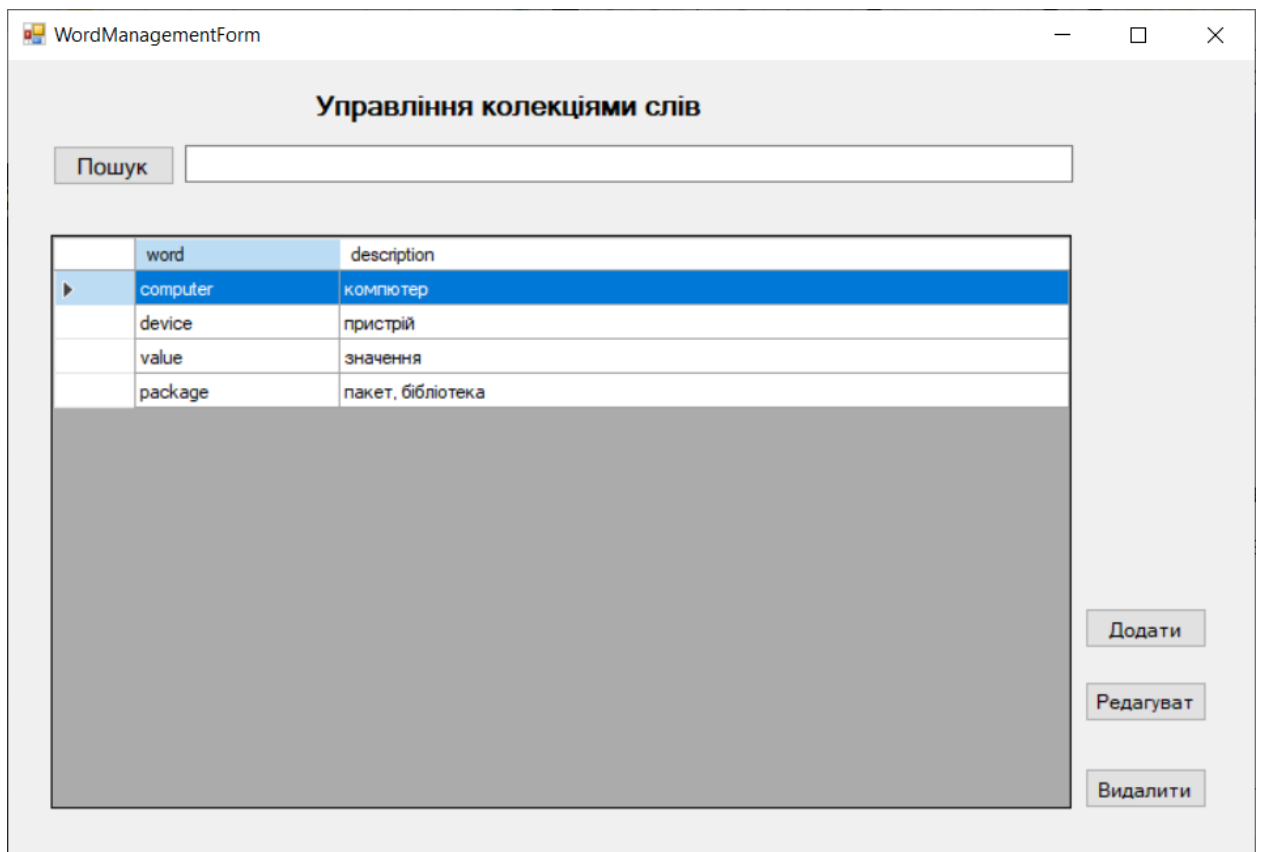
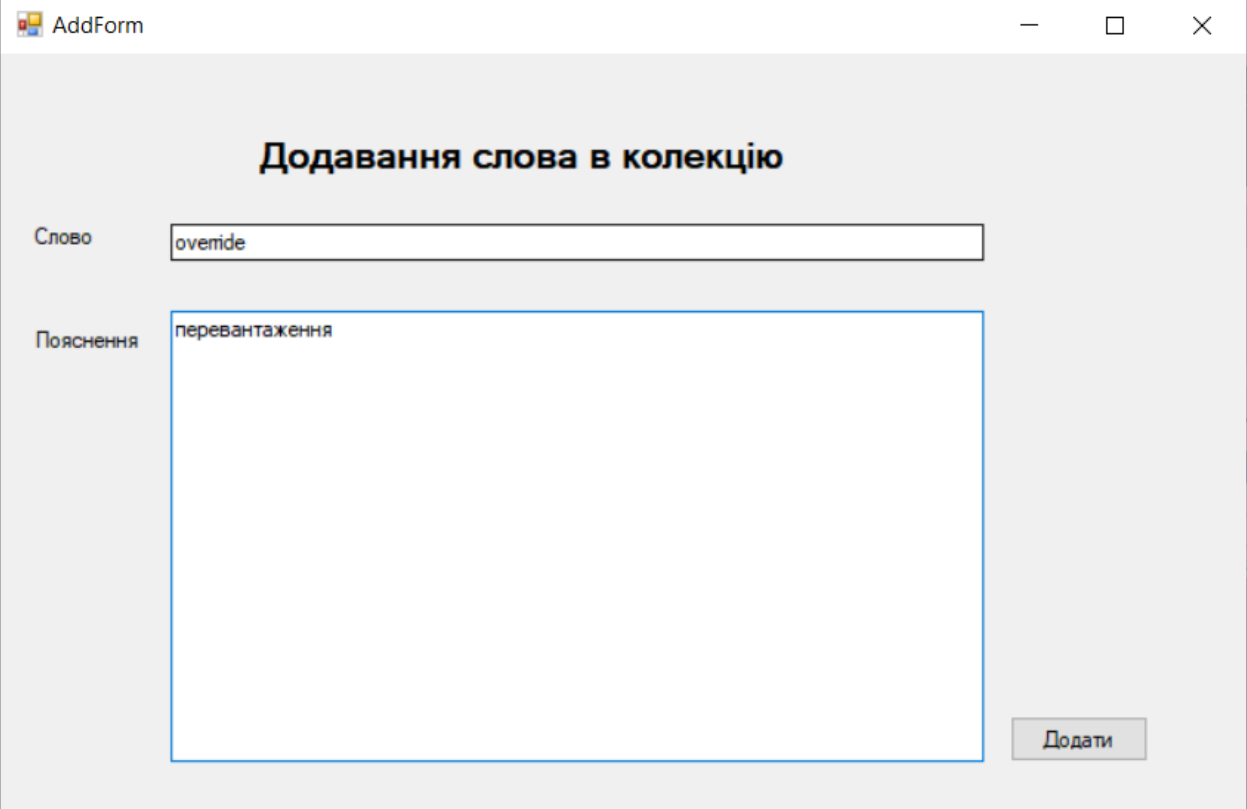


Рисунок 3.23. Управління словами

Користувач може додати слово до словника. Для цього йому потрібно натиснути кнопку Додати. У вікні додавання слова (рис. 3.24) користувач може ввести слово та його переклад (або пояснення).



The screenshot shows a window titled "AddForm" with a title bar containing standard window controls (minimize, maximize, close). The main content area has a heading "Додавання слова в колекцію". Below the heading, there are two input fields: "Слово" (Word) containing the text "overide" and "Пояснення" (Explanation) containing the text "перевантаження". A "Додати" (Add) button is located in the bottom right corner of the form area.

Рисунок 3.24. Додавання слова

Після натискання кнопки Додати слово додається до списку, а форма очищується. Після повернення до вікна Управління словами користувач може побачити додані нові слова (рис. 3.25).

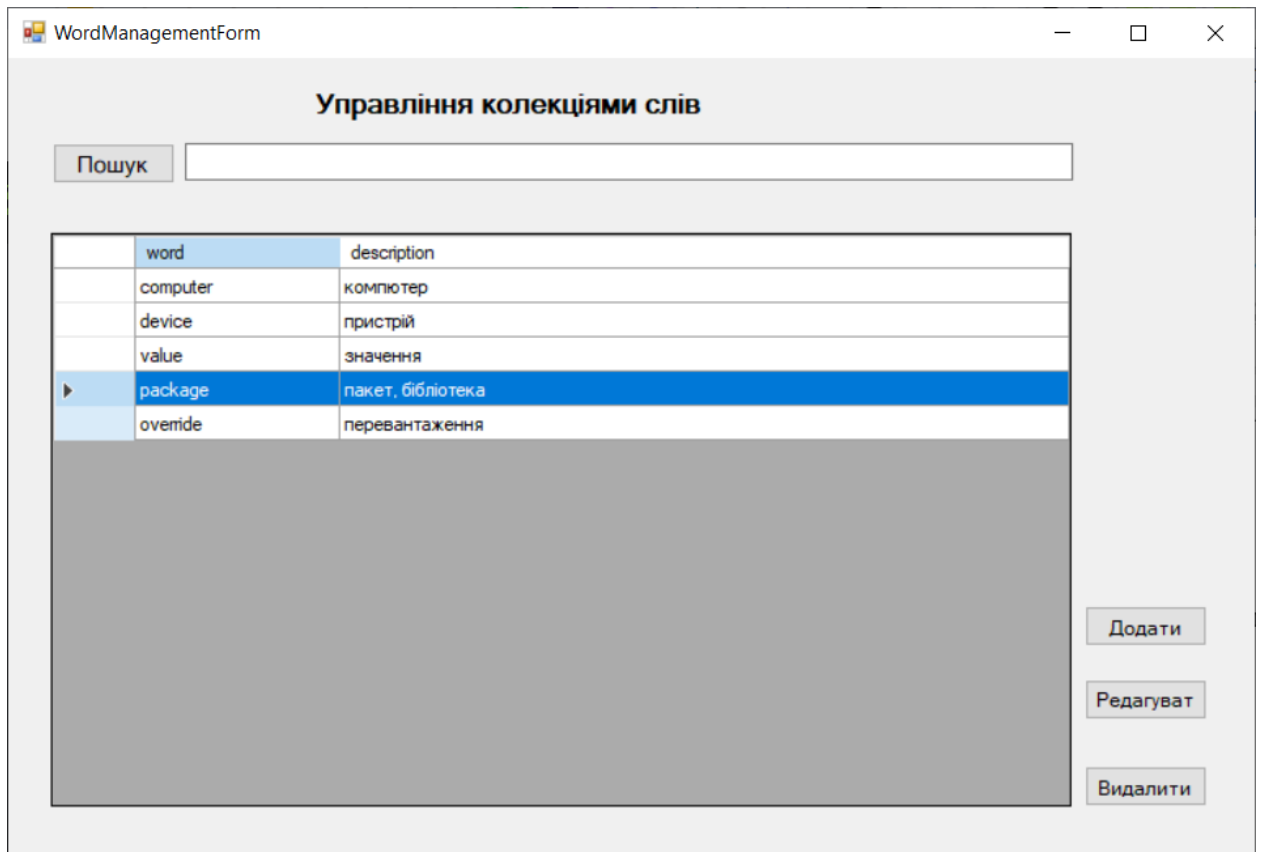


Рисунок 3.25. Результат додавання слова

Слово також можна видалити. Для цього потрібно встановити курсор на нього та натиснути кнопку Видалити. Слово буде видалено з БД.

Користувач може здійснювати пошук слів. Для цього достатньо у вікні пошуку ввести потрібне слово та натиснути кнопку Пошук (рис. 3.26).

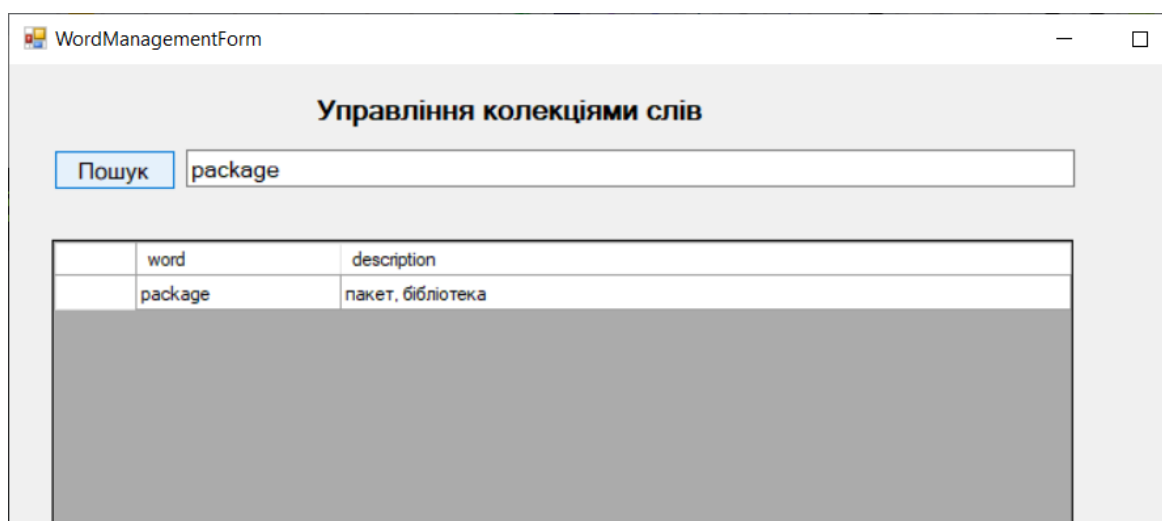


Рисунок 3.26. Пошук слова в колекції.

Для тестування та навчання користувачу потрібно в головному вікні натиснути кнопку Тест. У наступному вікні йому запропонують два режими тестування (рис. 3.27).

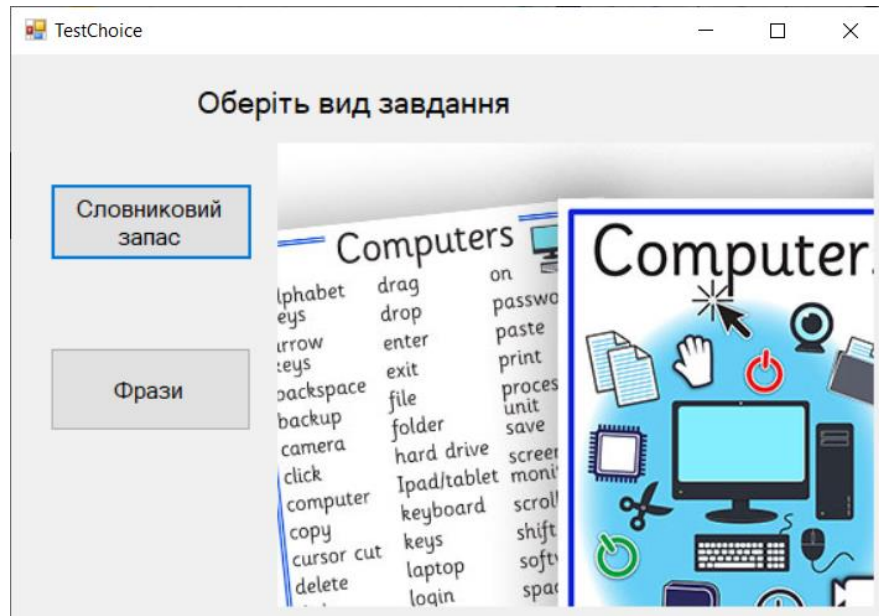


Рисунок 3.27. Вибір режиму тестування

При виборі Словниковий запас користувач вивчає та одночасно тестує себе на знання слів з колекції (рис. 3.28). Йому пропонується згідно опису слова дати його англійську назву.

 The screenshot shows a window titled "TestForm" with the heading "Тестування на знання слів" (Word knowledge testing). In the top right corner, it displays "Ваші бали" (Your score) as "0 / 6". Below the heading, the instruction "Введіть правильну відповідь" (Enter the correct answer) is shown. There is an empty text input field and a "Відповідь" (Answer) button. Below this, the heading "Опис слова" (Word description) is shown in blue. The description text is "перевантаження" (overload). At the bottom, there are two buttons: "Назад" (Back) and "Вперед" (Next).



### Рисунок 3.28. Тестування по колекції слів

Користувач вводить у рядок слово і натискає відповідь. Якщо відповідь правильна, він отримує повідомлення (рис. 3.29).

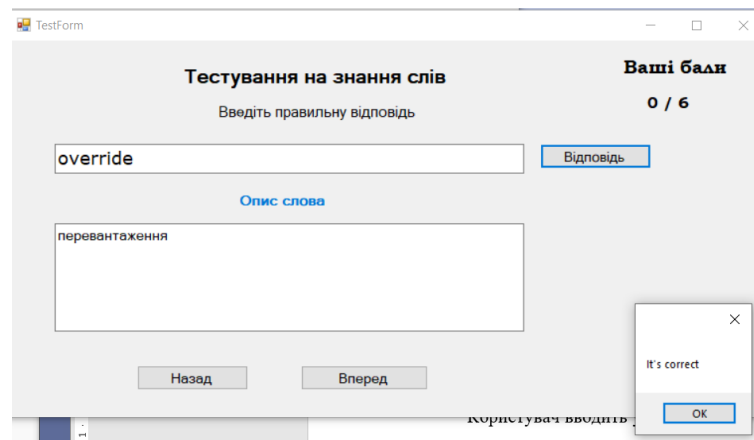


Рисунок 3.29. Правильна відповідь

У випадку неправильної відповіді користувач отримує повідомлення про помилку (рис. 3.30). Після натискання ОК рядок введення очищається і користувач може повторити спробу.

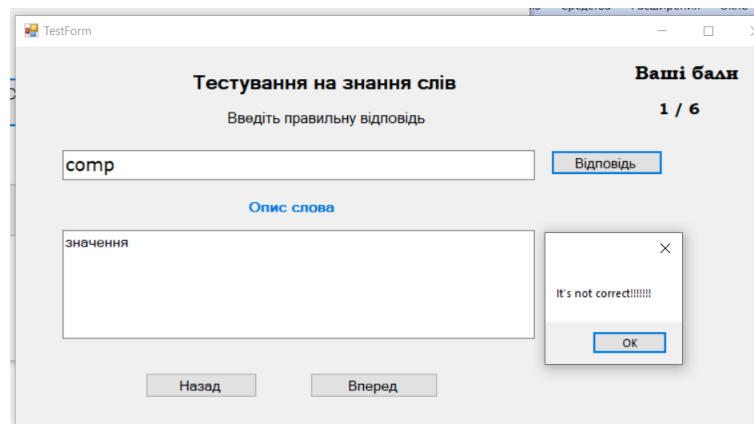


Рисунок 3.30. Неправильна відповідь

Якщо користувач не знає даного слова він може рухатися вперед або назад по словнику, відповідаючи на інші запитання. Також користувач може натиснути кнопку Відповідь та дізнатися правильне слово.

Вдалі відповіді користувача рахуються і праворуч він може побачити свій прогрес у вивченні слів.

При тестуванні на знання фраз перед користувачем відкривається вікно з порожнім завданням. Після натискання кнопки Вперед на екрані з'являється завдання (рис. 3.31), і користувач повинен обрати правильну відповідь.

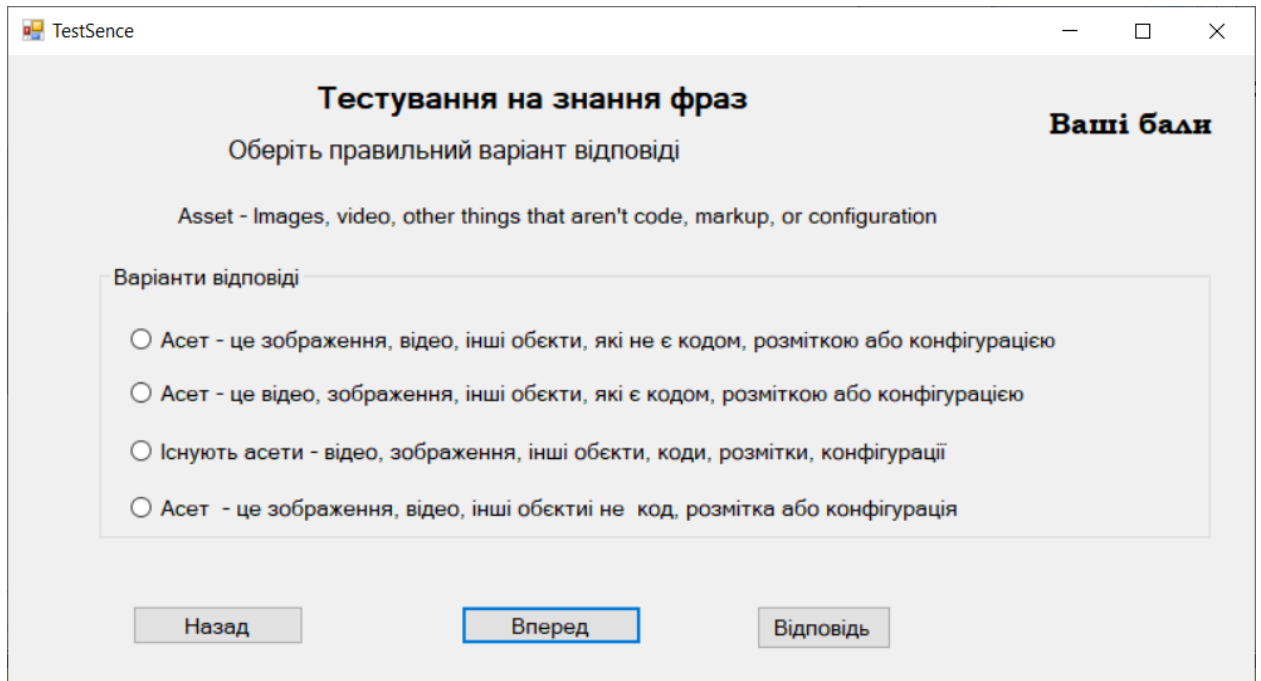


Рисунок 3.31. Завдання на фрази

Після вибору варіанта користувач натискає кнопку Відповідь і отримує відповідне повідомлення про правильність або неправильність своєї відповіді (рис. 3.32-3.33).

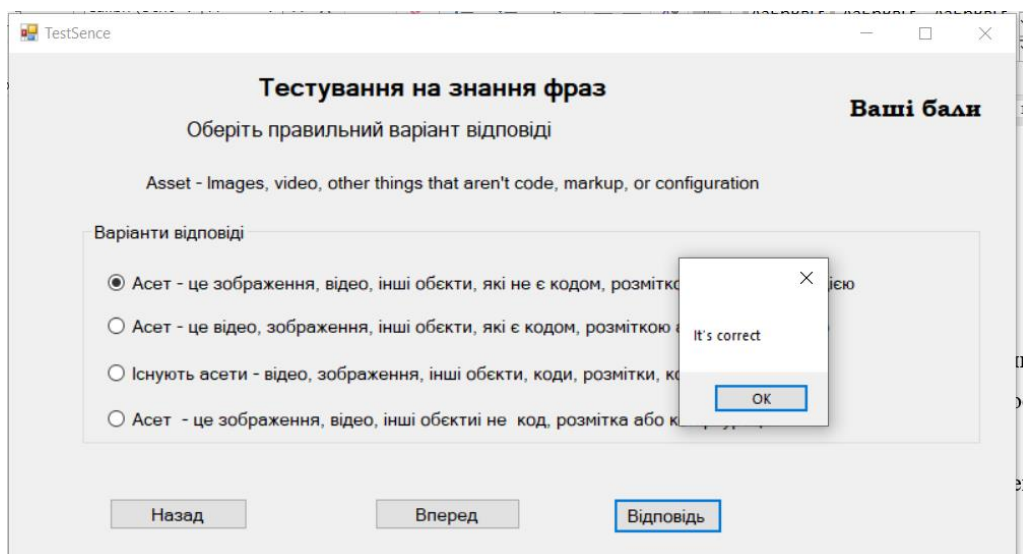


Рисунок 3.32. Повідомлення про правильність відповіді

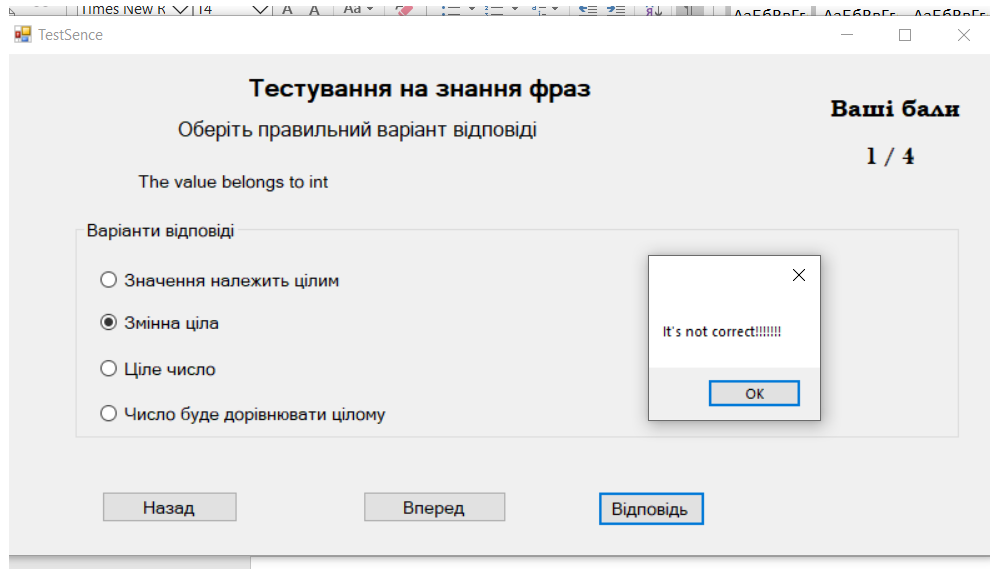


Рисунок 3.33. Повідомлення про неправильну відповідь

Користувач також може слідкувати за своїм прогресом, який висвічується праворуч вгорі.

Такий режим дає користувачу можливість краще вивчити англійську мову.

## ВИСНОВКИ

Відповідно до поставлених завдань та мети дослідження зробимо такі висновки.

У сучасному світі не можна недооцінювати зростаючий вплив інформаційних технологій на повсякденне життя та робоче середовище, де знання іноземних мов необхідне для повноцінної та грамотної роботи. Знання іноземної мови дозволяє спілкуватися в багатонаціональній команді фахівців, читати професійну літературу та документацію без перекладу.

У той же час для багатьох людей доволі важко примусити себе вивчати мову. Наявність різних програм, що дозволяють використовувати комп'ютер у процесі навчання, суттєво підвищує мотивацію.

Під час виконання кваліфікаційної роботи були розглянуті різні методики вивчення іноземних мов. Серед них виділяється комунікативна методика, яка повинна доповнюватись методикою словникових карток. Також були розглянуті різні програми для вивчення англійської мови. Виявлено, що ці програми в основному направлені на універсальну лексику та не враховують специфіку IT-галузі. До того ж часто вони є платними, причому на принципі абонентської плати. Для програмістів це не завжди виправдано, оскільки вони мають ненормований робочий день.

Далі було проведено проектування системи вивчення англійської мови на основі об'єктно-орієнтованого підходу. Були розроблені діаграми варіантів використання, послідовності та активності. Також була визначена архітектура системи. Це все дало змогу обґрунтовано визначити технології розробки, а саме Microsoft Visual Studio, мова C#, Windows Forms.

Після етапу проектування системи, був розроблений її інтерфейс, який забезпечив зручність роботи з програмою.

Результатом цих етапів стала розробка програмної системи з вивчення англійської мови. Програма дозволяє користувачу створювати

власну колекцію слів та управляти нею. Навчання в системі організовано за допомогою тестування. Користувач за описом повинен визначити англійське слово та ввести його як відповідь. Таким чином, задіюються зорова та механічна пам'ять людини, що пришвидшує вивчення слів. Також користувач може потренуватися у правильності перекладу та вживання тих слів, що він вивчає.

Отже, розроблена програма є недорогим зручним інструментом для вивчення англійської мови ІТ-фахівцями.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гарбарчук І. С., Бабич С. М. Розробка навчальної комп'ютерної програми «Англійська для програмістів». *Матеріали XXIV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів «Стан, досягнення та перспективи інформаційних систем і технологій»*. м. Одеса, 18-19 квітня 2024 р. / Видавництво ОНТУ, 2024. С. 156-157.
2. Рівні англійської мови: все по порядку. URL: <https://smile-school.kiev.ua/rivni-anglijskoyi-movi-vse-po-poryadku/> (дата звернення: 17.05.2024)
3. Кравець О. В. Актуальні методи викладання іноземної мови. *Науковий вісник Ужгородського національного університету. Серія «Педагогіка, соціальна робота»*. Випуск 35. С.92-95.
4. Філіппова Л. В., Підоренко В. В. Сучасні методи викладання іноземних мов. *Наукові записки*. 2013. Вип.13, С.46-48.
5. Шеверун Н. В. Форми, методи, педагогічні технології навчання іноземних мов студентів технічних університетів: досвід Польщі. *Вісник Житомирського державного університету. Педагогічні науки*. 2015. Випуск 1 (79). С.143-149.
6. Бунтурі Ю. В., Полянська Т. В., Ніконоров С. І. Сучасні методики викладання англійської та німецької мов. *Вчені записки Таврійського національного університету ім. В. І. Вернадського. Сер. : Філологія. Соціальні комунікації*. 2018. Т. 29 (68), № 1. С. 1-4.
7. Laufer B. The development of passive and active vocabulary in a second language: Same Or different? *AppliedLinguistics*. 1998. N 12. P. 255–271.
8. Nation I. S. P. Fluency and learning. *The English Teacher*. 1991. N 20. P.1–8.
9. We push you to speak another language. URL: <https://www.memrise.com> (дата звернення: 25.05.2024).

10. Безкоштовні мовні курси для українців. URL: <https://ua.babbel.com> (дата звернення: 25.05.2024).
11. Безкоштовний, веселий та ефективний спосіб вивчення мови! URL: <https://uk.duolingo.com> (дата звернення: 25.05.2024).
12. New language, new opportunities, new you. URL: <https://www.busuu.com/en> (дата звернення: 25.05.2024).
13. Brown E. Windows Forms Programming with C#. O'Reily, 2002. 768 p.
14. Perkind B., Reid J. D. Beginning C# and NET. Wiley's ebook EULA. 2021. 826 p.
15. C# Windows Forms Application Tutorial with Example/ URL: <https://www.guru99.com/c-sharp-windows-forms-application.html> (дата звернення: 02.11.2023).
16. Create a Windows Forms app in Visual Studio with C#. URL: <https://learn.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022> (дата звернення: 02.11.2023)

## ДОДАТКИ

### Додаток А

#### Лістинг програми

##### MainForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using Microsoft.Data.Sqlite;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace EnglishLearning
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();

            WordData.FileName =
Environment.GetFolderPath(Environment.SpecialFolder.Personal) + @"\default.db";
            WordData.Connect();
        }

        private void FileSelectButton_Click(object sender, EventArgs e)
        {
            var op = new OpenFileDialog();
            op.FileName = WordData.FileName;
            if (op.ShowDialog() == DialogResult.OK)
            {
                var ofd = new OpenFileDialog();
                ofd.FileName = WordData.FileName;
                if (ofd.ShowDialog() == DialogResult.OK)
                {
                    WordData.FileName = op.FileName;
                    WordData.FileName = ofd.FileName;
                    WordData.Connect();
                }
            }
        }

        private void WordListButton_Click(object sender, EventArgs e)
        {
            var wlForm = new WordManagementForm();
            wlForm.Show();
        }

        private void TestButton_Click(object sender, EventArgs e)
        {
            var tForm = new TestForm();
            tForm.Show();
        }

        private void MainForm_FormClosed(object sender, FormClosedEventArgs e)
        {
            WordData.Dispose();
        }
    }
}
```



```

    }
}
}

```

## MainForm.Design.cs

```

namespace EnglishLearning
{
    partial class MainForm
    {
        /// <summary>
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// </summary>
        /// <param name="disposing">. </param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows

        /// <summary>
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(MainForm));
            this.TestButton = new System.Windows.Forms.Button();
            this.WordListButton = new System.Windows.Forms.Button();
            this.FileSelectButton = new System.Windows.Forms.Button();
            this.label1 = new System.Windows.Forms.Label();
            this.pictureBox1 = new System.Windows.Forms.PictureBox();

            ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
            this.SuspendLayout();
            //
            // TestButton
            //
            this.TestButton.Location = new System.Drawing.Point(13, 154);
            this.TestButton.Margin = new System.Windows.Forms.Padding(4);
            this.TestButton.Name = "TestButton";
            this.TestButton.Size = new System.Drawing.Size(160, 31);
            this.TestButton.TabIndex = 0;
            this.TestButton.Text = "Тест";
            this.TestButton.UseVisualStyleBackColor = true;
            this.TestButton.Click += new System.EventHandler(this.TestButton_Click);
            //
            // WordListButton
            //
            this.WordListButton.Location = new System.Drawing.Point(13, 213);
            this.WordListButton.Margin = new System.Windows.Forms.Padding(4);
            this.WordListButton.Name = "WordListButton";
            this.WordListButton.Size = new System.Drawing.Size(160, 31);

```

```

        this.WordListButton.TabIndex = 1;
        this.WordListButton.Text = "Управління словами";
        this.WordListButton.UseVisualStyleBackColor = true;
        this.WordListButton.Click += new
System.EventHandler(this.WordListButton_Click);
        //
        // FileSelectButton
        //
        this.FileSelectButton.Location = new System.Drawing.Point(13, 282);
        this.FileSelectButton.Margin = new System.Windows.Forms.Padding(4);
        this.FileSelectButton.Name = "FileSelectButton";
        this.FileSelectButton.Size = new System.Drawing.Size(160, 31);
        this.FileSelectButton.TabIndex = 2;
        this.FileSelectButton.Text = "Редагування файлу";
        this.FileSelectButton.UseVisualStyleBackColor = true;
        this.FileSelectButton.Click += new
System.EventHandler(this.FileSelectButton_Click);
        //
        // label1
        //
        this.label1.AutoSize = true;
        this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label1.Location = new System.Drawing.Point(87, 24);
        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(593, 25);
        this.label1.TabIndex = 3;
        this.label1.Text = "Програма для вивчення англійської мови ІТ-фахівцями";
        //
        // pictureBox1
        //
        this.pictureBox1.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("pictureBox1.BackgroundImage")));
        this.pictureBox1.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Center;
        this.pictureBox1.ErrorImage = null;
        this.pictureBox1.InitialImage = null;
        this.pictureBox1.Location = new System.Drawing.Point(191, 77);
        this.pictureBox1.Name = "pictureBox1";
        this.pictureBox1.Size = new System.Drawing.Size(934, 469);
        this.pictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.CenterImage;
        this.pictureBox1.TabIndex = 4;
        this.pictureBox1.TabStop = false;
        //
        // MainForm
        //
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(1137, 558);
        this.Controls.Add(this.pictureBox1);
        this.Controls.Add(this.label1);
        this.Controls.Add(this.FileSelectButton);
        this.Controls.Add(this.WordListButton);
        this.Controls.Add(this.TestButton);
        this.Margin = new System.Windows.Forms.Padding(4);
        this.Name = "MainForm";
        this.Text = "Англійський для ІТ";
        this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.MainForm_FormClosed);
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
        this.ResumeLayout(false);
        this.PerformLayout();

```

```

    }

    #endregion

    private System.Windows.Forms.Button TestButton;
    private System.Windows.Forms.Button WordListButton;
    private System.Windows.Forms.Button FileSelectButton;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.PictureBox pictureBox1;
}
}

```

## WordManagementForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using Microsoft.Data.Sqlite;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace EnglishLearning
{
    public partial class WordManagementForm : Form
    {
        public WordManagementForm()
        {
            InitializeComponent();

            dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
            WordData.sql.CommandText = "SELECT * FROM MAIN";
            WordData.sql.ExecuteNonQuery();
            var reader = WordData.sql.ExecuteReader();
            while (reader.Read())
            {
                dataGridView1.Rows.Add(reader["WORD"].ToString(),
reader["DESCRIPTION"].ToString());
            }
            reader.Dispose();
        }

        private void deleteButton_Click(object sender, EventArgs e)
        {
            /*
            * foreach(var r in dataGridView1.SelectedRows)
            */
            foreach(DataGridViewRow r in dataGridView1.SelectedRows)
            {
                dataGridView1.Rows.Remove(r);
                WordData.Remove(r.Cells[0].ToString(), r.Cells[1].ToString());
            }
        }

        private void editButton_Click(object sender, EventArgs e)
        {
            /*
            *
            */

```



```

private void InitializeComponent()
{
    this.dataGridView1 = new System.Windows.Forms.DataGridView();
    this.Column1 = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.Column2 = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.deleteButton = new System.Windows.Forms.Button();
    this.addButton = new System.Windows.Forms.Button();
    this.editButton = new System.Windows.Forms.Button();
    this.searchButton = new System.Windows.Forms.Button();
    this.searchtb = new System.Windows.Forms.TextBox();
    this.label1 = new System.Windows.Forms.Label();

    ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).BeginInit();
    this.SuspendLayout();
    //
    // dataGridView1
    //
    this.dataGridView1.AllowUserToAddRows = false;
    this.dataGridView1.AllowUserToDeleteRows = false;
    this.dataGridView1.AllowUserToOrderColumns = true;
    this.dataGridView1.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
    this.dataGridView1.Columns.AddRange(new
System.Windows.Forms.DataGridViewColumn[] {
    this.Column1,
    this.Column2});
    this.dataGridView1.Location = new System.Drawing.Point(36, 155);
    this.dataGridView1.Margin = new System.Windows.Forms.Padding(4);
    this.dataGridView1.Name = "dataGridView1";
    this.dataGridView1.ReadOnly = true;
    this.dataGridView1.RowHeadersWidth = 51;
    this.dataGridView1.RowTemplate.Height = 21;
    this.dataGridView1.Size = new System.Drawing.Size(832, 432);
    this.dataGridView1.TabIndex = 0;
    //
    // Column1
    //
    this.Column1.HeaderText = "word";
    this.Column1.MinimumWidth = 6;
    this.Column1.Name = "Column1";
    this.Column1.ReadOnly = true;
    this.Column1.Width = 125;
    //
    // Column2
    //
    this.Column2.AutoSizeMode =
System.Windows.Forms.DataGridViewAutoSizeColumnMode.Fill;
    this.Column2.HeaderText = "description";
    this.Column2.MinimumWidth = 6;
    this.Column2.Name = "Column2";
    this.Column2.ReadOnly = true;
    //
    // deleteButton
    //
    this.deleteButton.Location = new System.Drawing.Point(877, 553);
    this.deleteButton.Margin = new System.Windows.Forms.Padding(4);
    this.deleteButton.Name = "deleteButton";
    this.deleteButton.Size = new System.Drawing.Size(100, 31);
    this.deleteButton.TabIndex = 1;
    this.deleteButton.Text = "Видалити";
    this.deleteButton.UseVisualStyleBackColor = true;
    this.deleteButton.Click += new
System.EventHandler(this.deleteButton_Click);
    //

```

```

// addButton
//
this.addButton.Location = new System.Drawing.Point(877, 432);
this.addButton.Margin = new System.Windows.Forms.Padding(4);
this.addButton.Name = "addButton";
this.addButton.Size = new System.Drawing.Size(100, 31);
this.addButton.TabIndex = 2;
this.addButton.Text = "Додати";
this.addButton.UseVisualStyleBackColor = true;
this.addButton.Click += new System.EventHandler(this.addButton_Click);
//
// editButton
//
this.editButton.Location = new System.Drawing.Point(877, 492);
this.editButton.Margin = new System.Windows.Forms.Padding(4);
this.editButton.Name = "editButton";
this.editButton.Size = new System.Drawing.Size(100, 31);
this.editButton.TabIndex = 3;
this.editButton.Text = "Редагувати";
this.editButton.UseVisualStyleBackColor = true;
this.editButton.Click += new System.EventHandler(this.editButton_Click);
//
// searchButton
//
this.searchButton.Location = new System.Drawing.Point(37, 105);
this.searchButton.Margin = new System.Windows.Forms.Padding(4);
this.searchButton.Name = "searchButton";
this.searchButton.Size = new System.Drawing.Size(100, 31);
this.searchButton.TabIndex = 4;
this.searchButton.Text = "Пошук";
this.searchButton.UseVisualStyleBackColor = true;
this.searchButton.Click += new
System.EventHandler(this.searchButton_Click);
//
// searchtb
//
this.searchtb.Location = new System.Drawing.Point(145, 105);
this.searchtb.Margin = new System.Windows.Forms.Padding(4);
this.searchtb.Name = "searchtb";
this.searchtb.Size = new System.Drawing.Size(723, 22);
this.searchtb.TabIndex = 5;
//
// label1
//
this.label1.AutoSize = true;
this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.label1.Location = new System.Drawing.Point(243, 48);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(296, 25);
this.label1.TabIndex = 6;
this.label1.Text = "Управління колекціями слів";
//
// WordManagementForm
//
this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(1016, 600);
this.Controls.Add(this.label1);
this.Controls.Add(this.searchtb);
this.Controls.Add(this.searchButton);
this.Controls.Add(this.editButton);
this.Controls.Add(this.addButton);
this.Controls.Add(this.deleteButton);

```

```

        this.Controls.Add(this.dataGridView1);
        this.Margin = new System.Windows.Forms.Padding(4);
        this.Name = "WordManagementForm";
        this.Text = "WordManagementForm";

        ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).EndInit();
        this.ResumeLayout(false);
        this.PerformLayout();

    }

    #endregion

    private System.Windows.Forms.DataGridView dataGridView1;
    private System.Windows.Forms.DataGridViewTextBoxColumn Column1;
    private System.Windows.Forms.DataGridViewTextBoxColumn Column2;
    private System.Windows.Forms.Button deleteButton;
    private System.Windows.Forms.Button addButton;
    private System.Windows.Forms.Button editButton;
    private System.Windows.Forms.Button searchButton;
    private System.Windows.Forms.TextBox searchtb;
    private System.Windows.Forms.Label label1;
    }
}

```

## AddForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace EnglishLearning
{
    public partial class AddForm : Form
    {
        DataGridView d;
        public AddForm(ref DataGridView d)
        {
            InitializeComponent();
            this.d = d;
        }

        private void OKButton_Click(object sender, EventArgs e)
        {
            WordData.Add(WordText.Text, DescriptionText.Text);
            d.Rows.Add(WordText.Text, DescriptionText.Text);
            WordText.Clear();
            DescriptionText.Clear();
        }
    }
}

```

## AddForm.Designer.cs

```

namespace EnglishLearning
{
    partial class AddForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
        otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.WordText = new System.Windows.Forms.TextBox();
            this.DescriptionText = new System.Windows.Forms.TextBox();
            this.OKButton = new System.Windows.Forms.Button();
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.SuspendLayout();
            //
            // WordText
            //
            this.WordText.Location = new System.Drawing.Point(122, 113);
            this.WordText.Margin = new System.Windows.Forms.Padding(4, 4, 4, 4);
            this.WordText.Name = "WordText";
            this.WordText.Size = new System.Drawing.Size(585, 22);
            this.WordText.TabIndex = 0;
            //
            // DescriptionText
            //
            this.DescriptionText.Location = new System.Drawing.Point(122, 171);
            this.DescriptionText.Margin = new System.Windows.Forms.Padding(4, 4, 4,
4);

            this.DescriptionText.Multiline = true;
            this.DescriptionText.Name = "DescriptionText";
            this.DescriptionText.Size = new System.Drawing.Size(585, 300);
            this.DescriptionText.TabIndex = 1;
            //
            // OKButton
            //
            this.OKButton.Location = new System.Drawing.Point(728, 440);
            this.OKButton.Margin = new System.Windows.Forms.Padding(4, 4, 4, 4);
            this.OKButton.Name = "OKButton";
            this.OKButton.Size = new System.Drawing.Size(100, 31);
            this.OKButton.TabIndex = 2;

```



```

        this.OKButton.Text = "Додати";
        this.OKButton.UseVisualStyleBackColor = true;
        this.OKButton.Click += new System.EventHandler(this.OKButton_Click);
        //
        // label1
        //
        this.label1.AutoSize = true;
        this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 13.8F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label1.Location = new System.Drawing.Point(181, 53);
        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(370, 29);
        this.label1.TabIndex = 3;
        this.label1.Text = "Додавання слова в колекцію";
        //
        // label2
        //
        this.label2.AutoSize = true;
        this.label2.Location = new System.Drawing.Point(22, 113);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(48, 17);
        this.label2.TabIndex = 4;
        this.label2.Text = "Слово";
        //
        // label3
        //
        this.label3.AutoSize = true;
        this.label3.Location = new System.Drawing.Point(21, 182);
        this.label3.Name = "label3";
        this.label3.Size = new System.Drawing.Size(81, 17);
        this.label3.TabIndex = 5;
        this.label3.Text = "Пояснення";
        //
        // AddForm
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(898, 505);
        this.Controls.Add(this.label3);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.label1);
        this.Controls.Add(this.OKButton);
        this.Controls.Add(this.DescriptionText);
        this.Controls.Add(this.WordText);
        this.Margin = new System.Windows.Forms.Padding(4, 4, 4, 4);
        this.Name = "AddForm";
        this.Text = "AddForm";
        this.ResumeLayout(false);
        this.PerformLayout();

    }

#endregion

private System.Windows.Forms.TextBox WordText;
private System.Windows.Forms.TextBox DescriptionText;
private System.Windows.Forms.Button OKButton;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
}
}

```

## TestForm.Designer.cs

```

namespace EnglishLearning
{
    partial class TestForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
        otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.prevButton = new System.Windows.Forms.Button();
            this.nextButton = new System.Windows.Forms.Button();
            this.wordText = new System.Windows.Forms.TextBox();
            this.descriptionText = new System.Windows.Forms.TextBox();
            this.answerButton = new System.Windows.Forms.Button();
            this.checkBox1 = new System.Windows.Forms.CheckBox();
            this.checkBox2 = new System.Windows.Forms.CheckBox();
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.SuspendLayout();
            //
            // prevButton
            //
            this.prevButton.Location = new System.Drawing.Point(54, 472);
            this.prevButton.Margin = new System.Windows.Forms.Padding(4);
            this.prevButton.Name = "prevButton";
            this.prevButton.Size = new System.Drawing.Size(150, 31);
            this.prevButton.TabIndex = 0;
            this.prevButton.Text = "Назад";
            this.prevButton.UseVisualStyleBackColor = true;
            //
            // nextButton
            //
            this.nextButton.Location = new System.Drawing.Point(495, 472);
            this.nextButton.Margin = new System.Windows.Forms.Padding(4);
            this.nextButton.Name = "nextButton";
            this.nextButton.Size = new System.Drawing.Size(170, 31);
            this.nextButton.TabIndex = 1;
            this.nextButton.Text = "Вперед";
        }
    }
}

```

```

this.nextButton.UseVisualStyleBackColor = true;
this.nextButton.Click += new System.EventHandler(this.nextButton_Click);
//
// wordText
//
this.wordText.Font = new System.Drawing.Font("MS UI Gothic", 20.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(128)));
this.wordText.Location = new System.Drawing.Point(150, 130);
this.wordText.Margin = new System.Windows.Forms.Padding(4);
this.wordText.Name = "wordText";
this.wordText.Size = new System.Drawing.Size(721, 41);
this.wordText.TabIndex = 2;
//
// descriptionText
//
this.descriptionText.Location = new System.Drawing.Point(63, 194);
this.descriptionText.Margin = new System.Windows.Forms.Padding(4);
this.descriptionText.Multiline = true;
this.descriptionText.Name = "descriptionText";
this.descriptionText.Size = new System.Drawing.Size(629, 207);
this.descriptionText.TabIndex = 3;
this.descriptionText.UseWaitCursor = true;
//
// answerButton
//
this.answerButton.Location = new System.Drawing.Point(256, 472);
this.answerButton.Margin = new System.Windows.Forms.Padding(4);
this.answerButton.Name = "answerButton";
this.answerButton.Size = new System.Drawing.Size(149, 31);
this.answerButton.TabIndex = 4;
this.answerButton.Text = "Відповідь";
this.answerButton.UseVisualStyleBackColor = true;
this.answerButton.Click += new
System.EventHandler(this.answerButton_Click);
//
// checkBox1
//
this.checkBox1.AutoSize = true;
this.checkBox1.Location = new System.Drawing.Point(731, 350);
this.checkBox1.Margin = new System.Windows.Forms.Padding(4);
this.checkBox1.Name = "checkBox1";
this.checkBox1.Size = new System.Drawing.Size(140, 21);
this.checkBox1.TabIndex = 5;
this.checkBox1.Text = "Завдання типу 1";
this.checkBox1.UseVisualStyleBackColor = true;
//
// checkBox2
//
this.checkBox2.AutoSize = true;
this.checkBox2.Location = new System.Drawing.Point(731, 380);
this.checkBox2.Margin = new System.Windows.Forms.Padding(4);
this.checkBox2.Name = "checkBox2";
this.checkBox2.Size = new System.Drawing.Size(140, 21);
this.checkBox2.TabIndex = 6;
this.checkBox2.Text = "Завдання типу 2";
this.checkBox2.UseVisualStyleBackColor = true;
//
// label1
//
this.label1.AutoSize = true;
this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 13.8F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.label1.Location = new System.Drawing.Point(429, 23);
this.label1.Name = "label1";

```

```

        this.label1.Size = new System.Drawing.Size(158, 29);
        this.label1.TabIndex = 7;
        this.label1.Text = "Тестування";
        //
        // label2
        //
        this.label2.AutoSize = true;
        this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label2.Location = new System.Drawing.Point(354, 74);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(299, 25);
        this.label2.TabIndex = 8;
        this.label2.Text = "Введіть правильну відповідь";
        //
        // label3
        //
        this.label3.AutoSize = true;
        this.label3.Location = new System.Drawing.Point(732, 294);
        this.label3.Name = "label3";
        this.label3.Size = new System.Drawing.Size(99, 17);
        this.label3.TabIndex = 9;
        this.label3.Text = "Тип завдання";
        //
        // label4
        //
        this.label4.AutoSize = true;
        this.label4.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold |
System.Drawing.FontStyle.Italic))), System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label4.Location = new System.Drawing.Point(60, 139);
        this.label4.Name = "label4";
        this.label4.Size = new System.Drawing.Size(83, 18);
        this.label4.TabIndex = 10;
        this.label4.Text = "Завдання";
        //
        // TestForm
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(942, 600);
        this.Controls.Add(this.label4);
        this.Controls.Add(this.label3);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.label1);
        this.Controls.Add(this.checkBox2);
        this.Controls.Add(this.checkBox1);
        this.Controls.Add(this.answerButton);
        this.Controls.Add(this.descriptionText);
        this.Controls.Add(this.wordText);
        this.Controls.Add(this.nextButton);
        this.Controls.Add(this.prevButton);
        this.Margin = new System.Windows.Forms.Padding(4);
        this.Name = "TestForm";
        this.Text = "TestForm";
        this.Load += new System.EventHandler(this.TestForm_Load);
        this.ResumeLayout(false);
        this.PerformLayout();

    }

#endregion

private System.Windows.Forms.Button prevButton;

```

```

private System.Windows.Forms.Button nextButton;
private System.Windows.Forms.TextBox wordText;
private System.Windows.Forms.TextBox descriptionText;
private System.Windows.Forms.Button answerButton;
private System.Windows.Forms.CheckBox checkBox1;
private System.Windows.Forms.CheckBox checkBox2;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Label label4;
    }
}

```

## SenseTest.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace EnglishLearning
{
    public partial class TestSence : Form
    {
        List<int> TestOrder = new List<int>();
        int Order = 0;
        int score = 0;

        struct sentTest
        {
            public string question;
            public string ans1, ans2, ans3, ans4;
        };
        List<sentTest> arrSent = new List<sentTest>();

        public TestSence()
        {
            InitializeComponent();

            sentTest temp;
            string[] lines = File.ReadAllLines("test.txt");
            if (lines.Length == 0)
            {
                MessageBox.Show("It's not correct!!!!!!");
                return;
            }
            for (int i=0; i<lines.Length;i+=5)
            {
                temp.question = lines[i];
                temp.ans1 = lines[i + 1];
                temp.ans2 = lines[i + 2];
                temp.ans3 = lines[i + 3];
                temp.ans4 = lines[i + 4];
                arrSent.Add(temp);
            }
        }
    }
}

```

```

    for (int i = 0; i < arrSent.Count(); i++)
    {
        TestOrder.Add(i);
    }
    Random rnd = new Random();
    for (int i = TestOrder.Count() - 1; i >= 1; i--)
    {
        int j = rnd.Next(i + 1);
        // обменять значения data[j] и data[i]
        var temp1 = TestOrder[j];
        TestOrder[j] = TestOrder[i];
        TestOrder[i] = temp1;
    }
}

private void TestSence_Load(object sender, EventArgs e)
{
    radioButton1.Checked = radioButton2.Checked = radioButton3.Checked =
radioButton4.Checked = false;
    /*LabelSentence.Text = arrSent[TestOrder[Order]].question;
    radioButton1.Text = arrSent[TestOrder[Order]].ans1;
    radioButton2.Text = arrSent[TestOrder[Order]].ans2;
    radioButton3.Text = arrSent[TestOrder[Order]].ans3;
    radioButton4.Text = arrSent[TestOrder[Order]].ans4;*/
}

private void button2_Click(object sender, EventArgs e)
{
    if (Order >= TestOrder.Count() - 1)
    {
        Order = 0;
        return;
    }
    else
    {
        LabelSentence.Text = arrSent[TestOrder[Order]].question;
        radioButton1.Text = arrSent[TestOrder[Order]].ans1;
        radioButton2.Text = arrSent[TestOrder[Order]].ans2;
        radioButton3.Text = arrSent[TestOrder[Order]].ans3;
        radioButton4.Text = arrSent[TestOrder[Order]].ans4;
        radioButton1.Checked = radioButton2.Checked = radioButton3.Checked =
radioButton4.Checked = false;
        Order++;
    }
}

private void button1_Click(object sender, EventArgs e)
{
    if (Order == 0)
    {
        Order = TestOrder.Count() - 1;
        //return;
    }
    else
    {
        Order--;
        LabelSentence.Text = arrSent[TestOrder[Order]].question;
        radioButton1.Text = arrSent[TestOrder[Order]].ans1;
        radioButton2.Text = arrSent[TestOrder[Order]].ans2;
        radioButton3.Text = arrSent[TestOrder[Order]].ans3;
        radioButton4.Text = arrSent[TestOrder[Order]].ans4;
    }
}

```

```

        radioButton1.Checked = radioButton2.Checked = radioButton3.Checked =
radioButton4.Checked = false;
    }
}

private void button3_Click(object sender, EventArgs e)
{
    if (radioButton1.Checked == true)
    {
        MessageBox.Show("It's correct");
        score++;
        string mes1, mes2;
        mes1 = score.ToString();
        mes2 = TestOrder.Count().ToString();
        labelScore.Text = mes1 + " / " + mes2;
    }
    else
        MessageBox.Show("It's not correct!!!!!!");
}
}
}
}

```

## TestForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace EnglishLearning
{
    public partial class TestForm : Form
    {
        List<int> TestOrder = new List<int>();
        int Order = 0;
        int score = 0;

        struct wordTest {
            public string word;
            public string description;
        };

        List<wordTest> arrWord = new List<wordTest>();

        public TestForm()
        {
            InitializeComponent();

            WordData.sql.CommandText = "SELECT * FROM MAIN";
            WordData.sql.ExecuteNonQuery();
            var reader = WordData.sql.ExecuteReader();
            wordTest temp;
            int countWord = 0;
            while (reader.Read())
            {
                temp.word = reader["WORD"].ToString();
                temp.description = reader["DESCRIPTION"].ToString();
            }
        }
    }
}

```

```

        arrWord.Add(temp);
        countWord++;
    }
    reader.Dispose();

    for (int i= 0; i < countWord;i++)
    {
        TestOrder.Add(i);
    }
    Random rnd = new Random();
    for (int i = TestOrder.Count() - 1; i >= 1; i--)
    {
        int j = rnd.Next(i + 1);
        // обменять значения data[j] и data[i]
        var temp1 = TestOrder[j];
        TestOrder[j] = TestOrder[i];
        TestOrder[i] = temp1;
    }
}

private void TestForm_Load(object sender, EventArgs e)
{
    descriptionText.Text = arrWord[TestOrder[Order]].description;
    string mes1, mes2;
    mes1 = score.ToString();
    mes2 = TestOrder.Count().ToString();
    label6.Text = mes1 + " / " + mes2;
}

private void answerButton_Click(object sender, EventArgs e)
{
    if (wordText.TextLength == 0)
    {
        wordText.Text = arrWord[TestOrder[Order]].word;
    }
    else
    {
        if (wordText.Text == arrWord[TestOrder[Order]].word)
        {
            MessageBox.Show("It's correct");
            score++;
            string mes1, mes2;
            mes1 = score.ToString();
            mes2 = TestOrder.Count().ToString();
            label6.Text = mes1 + " / " + mes2;
        }
        else
        {
            MessageBox.Show("It's not correct!!!!!!");
            wordText.Clear();
        }
    }
}

private void nextButton_Click(object sender, EventArgs e)
{
    if (Order >= TestOrder.Count() - 1)
    {
        Order = -1;
        //return;
    }
}

```



```
    }  
    else  
    {  
        Order++;  
        wordText.Clear();  
        descriptionText.Text = arrWord[TestOrder[Order]].description;  
    }  
}  
  
private void prevButton_Click(object sender, EventArgs e)  
{  
    if (Order == 0)  
    {  
        Order = TestOrder.Count()-1;  
        //return;  
    }  
    else  
    {  
        Order--;  
        wordText.Clear();  
        descriptionText.Text = arrWord[TestOrder[Order]].description;  
    }  
}  
}
```