

Міністерство освіти і науки України
Кам'янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп'ютерних наук

Кваліфікаційна робота магістра
з теми: «**Методи і засоби інформаційної підтримки процесу вивчення
іноземної мови**»

Виконав: здобувач вищої освіти групи KN1-M23,
спеціальності 122 комп'ютерні науки

Жолтовський Олексій Олексійович

(прізвище та ім'я і по батькові здобувача вищої освіти)

Керівник: Федорчук В. А., д-р техн. наук, професор

(прізвище та ініціали, науковий ступінь, учене звання)

Рецензент: Ковальська І. Б., канд. ф.-м. наук, доцент

(прізвище та ініціали, науковий ступінь, учене звання)

м. Кам'янець-Подільський – 2024 р.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	3
ВСТУП.....	4
РОЗДІЛ 1. Особливості методів запам'ятовування інформації у вивченні іноземної мови	7
1.1 Асоціативна пам'ять та іноземна мова.....	7
1.2. Флеш-картки та інтервальне повторення	7
1.3. Значення збирання інформації в процесі вивчення іноземної мови.....	9
1.4. Існуючі програмні засоби для вивчення іноземних мов.....	10
1.5 Відео-матеріали у вивченні мови.....	11
1.6 Штучний інтелект у вивченні мови	12
РОЗДІЛ 2. Проєктування та дизайн програмного продукту	13
2.1 Проблематика та завдання застосунку	13
2.2 Цільова аудиторія програмного засобу	14
2.3 Користувацький досвід та збирання інформації	15
2.4 Пріоритезація зібраного контенту	17
2.5 Тестування та оцінка знань користувача	19
2.6 Зв'язки між контентом	20
2.7 Інтегровані пошукові та генеративні сервіси	23
2.8 Прослуховування вимови	24
РОЗДІЛ 3. Реалізація прототипу програмного продукту	26
3.1. Фреймворк Flutter Dart та кросплатформність	26
3.2 Визначення СУБД	27
3.3 Планування та реалізація бази даних	28
3.4 Впровадження пошукових та генеративних сервісів.....	31
3.5 Оверлей та особливості його реалізації.....	33
3.6 Деталі впровадження TTS.....	34
3.7 Демонстрація роботи прототипу	35
ВИСНОВКИ	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	42

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API — інтерфейс для програмування застосунків, що надається певною платформою або фреймворком.

BLOB — аббревіатура від binary large object. Тип даних, який дозволяє зберігати двійкові дані часто невизначеної довжини.

ID — будь-який ідентифікатор певної сутності.

OS — операційна система.

TTS — аббревіатура від text-to-speech, тобто переведення тексту в мовлення.

БД — база даних.

ПК — персональний комп'ютер.

ШІ — штучний інтелект.

ВСТУП

Актуальність. Володіння багатьма іноземними мовами завжди було вимогою для багатьох людей, які при здобуванні знань користуються іншомовними джерелами або спілкуються з колегами в іноземних компаніях. Більше того, у сучасному світі потреба у вивченні іноземних мов, особливо англійської, стає все більш актуальною в умовах теперішньої глобалізації.

На сьогоднішній день існує багато засобів для вивчення іноземних мов, починаючи від словників та закінчуючи програмами зі штучним інтелектом, а також спеціалізованими онлайн ресурсами. Однак, більшість з них фокусуються лише на одній мові та є обмеженими щодо навчального контенту. З одного боку використання таких засобів є ефективним для початківців, а саме людей, які не стикалися з вибраною мовою та вивчають її вперше. З іншого боку такий підхід втрачає свою ефективність, як тільки користувач засвоїть матеріали навчального курсу, оскільки для подальшого вдосконалення йому потрібно витратити час на проходження нових курсів.

Відповідним рішенням для усунення цих недоліків є використання застосунків, що не прив'язуються до певного набору навчальних матеріалів, а надають користувачу можливість формувати з високим рівнем автоматизації навчальний контент, який базується на власному досвіді і власних потребах, тобто процес навчання продовжується в ході професійної діяльності користувача.

Отже, акцент робиться на пошуку ефективного рішення для швидкого занурення людини в необхідну іноземну лексику, більше орієнтуючись на її професійну або іншу діяльність, що дозволить поєднати вивчення іноземної мови з роботою або іншою діяльністю, оптимізуючи процес засвоєння нових знань, при умові якщо людина вимушена або готова взаємодіяти з інформацією на іноземній мові, такою як відео, статті, документація тощо. Планується що такий підхід допоможе краще запам'ятовувати лексику, завдяки феномену пасивної практики та прямої асоціації іноземної лексики з

діяльністю людини, що може заощадити час на активне свідоме вивчення навчальних матеріалів, дозволяючи фокусуватися на професійній або повсякденній діяльності, де нові лексичні одиниці засвоюються природно через контекст.

Об'єктом дослідження є процес вивчення іноземної мови із залученням допоміжних спеціалізованих комп'ютерних засобів.

Предметом дослідження є комп'ютеризовані засоби інформаційної підтримки процесу вивчення іноземної мови.

Мета і завдання дослідження. Метою дослідження є розробка методів і засобів для підвищення ефективності процесу запам'ятовування іноземних слів та фраз на основі створення і використання спеціалізованого програмного засобу, який дозволить користувачеві формувати асоціативні зв'язки між словами, фразами, зображеннями та іншими фрагментами інформації, із залученням пасивної практики.

Для досягнення поставленої мети необхідно виконати такі завдання:

1. Проаналізувати сучасні методи активізації процесу запам'ятовування інформації, зокрема іноземних слів;
2. Розробити проєкт та дизайн програмного рішення;
3. Розробити базовий функціонал програмного інструменту для професійного розвитку, збирання та засвоєння матеріалу на іноземних мовах.

Методи дослідження. Методологія дослідження зосереджена на вивченні та аналізі звичної й рутинної поведінки користувачів, коли вони стикаються з незнайомими матеріалами на іноземній мові, що дозволяє виявити потенційні області для спрощення та автоматизації процедур, пов'язаних із збиранням лексики, формуванням асоціативних зв'язків між ними та іншими об'єктами для забезпечення кращого досвіду запам'ятовування та засвоєння інформації та лексики.

Практичне значення одержаних результатів. Вважається, що це дослідження зробить внесок як в архітектуру, так і в реалізацію

спеціалізованого програмного забезпечення для вивчення іноземних мов. Вивчаючи та аналізуючи звичні та рутинні дії користувачів, коли вони стикаються з невідомими матеріалами на іноземних мовах, ця робота має на меті визначити ефективні стратегії спрощення, автоматизації та покращення процесу вивчення лексики та інших мовних структур з акцентом на індивідуальні особливості користувача та об'єднання процесу вивчення мови з користувацькою діяльністю. Практичні результати цієї роботи можуть бути застосовані до розробки нових інтуїтивно зрозумілих і зручних інструментів, додатків для вивчення мов, що фокусуються на допомозі та користувачу у вивченні матеріалів, з якими він стикається в процесі будь-якої діяльності, ніж тих які фокусуються на чітко визначених навчальних матеріалах, що не представляють особливого інтересу для користувача.

Наукова новизна дослідження полягає у подальшому розвитку методів активізації процесу оволодіння іноземною мовою на основі розробки спеціалізованого програмного засобу, який відрізняється від існуючих високим рівнем автоматизації процесу збору навчального контенту та можливістю роботи в фоновому режимі, що дає змогу користувачеві формувати власну траєкторію навчання, а також навчатися в ході своєї професійної діяльності.

Апробація результатів. Результати дослідження доповідались на 10-й Міжнародній науковій конференції «Сучасні проблеми математичного моделювання, прогнозування та оптимізації» (Optima-2024) пам'яті почесного професора Кам'янець-Подільського національного університету, д. т. н., професора, чл.-кор. НАПНУ Анатолія Федоровича Верланя, яка відбулась 28-29 червня 2024 р. в Кам'янець-Подільському національному університеті імені Івана Огієнка, а також на двох Звітних студентських конференціях. За матеріалами дослідження здійснено 4 публікації [3 - 6].

Структура роботи. Робота складається зі вступу, трьох розділів, висновків та списку використаних джерел, що налічує 44 джерела.

РОЗДІЛ 1. ОСОБЛИВОСТІ МЕТОДІВ ЗАПАМ'ЯТОВУВАННЯ ІНФОРМАЦІЇ У ВИВЧЕННІ ІНОЗЕМНОЇ МОВИ

1.1 Асоціативна пам'ять та іноземна мова

Ще з раннього віку людина починає формувати асоціативні зв'язки між речами, що вона бачить, та звуками, в процесі вивчення рідної мови. Ця стратегія асоціативного навчання може бути ефективно застосована для вивчення іноземної мови, де не потрібна пряма асоціація між словом та його перекладом. Натомість, поєднання слова з відповідним зображенням може сприяти негайному згадуванню образів при зустрічі з іноземним словом або реченням [40].

Такий підхід дозволяє ментально відновлювати образи при зустрічі зі словом або реченням на іноземній мові.

Образ — це не обов'язково якийсь візуальний об'єкт. Це може бути подія або інший досвід, з яким користувач асоціює певне слово або речення.

При цьому відповідне слово або речення не обов'язково має напряду асоціюватися з об'єктом, що воно означає, для прикладу, німецьке слово «Fenster» (укр. Вікно) дуже схоже по звучанню на англійське слово «Fence» (укр. Паркан). До того ж паркан може містити щось на подібні вікна, що вже викликає деякі асоціації між словами «Fenster» та «Fence».

Отже, для запам'ятовування іноземних слів не обов'язково використовувати прямі візуальні або інші образи. Достатньо використовувати ті, які в повній мірі відновлюють слово в пам'яті.

1.2. Флеш-картки та інтервальне повторення

Флеш-картки — це концепція для організації матеріалу на іноземній мові у вигляді карток, де слово напряду асоціюється з образом без залучення рідної мови для перекладу. Суть методу полягає у тому, що на одній стороні картки наноситься фото, а на іншій — слово іноземною мовою [17, с. 52].

Відповідно, людина може зібрати набір карток зі словами або фразами, які їй необхідно вивчити.

Але з накопиченням матеріалу виникає питання, які саме картки пріоритетніше вивчати, а які слід відкласти на потім. Це вирішується методом інтервального повторення. Він був винайдений у 1880-х роках німецьким вченим Германом Еббінгаузом, який також є автором кривої забування, емпіричного графіку, що описує процес забування вивченого протягом часу, на основі якого базується цей метод [42].

Суть методу полягає в експлуатації ефекту інтервалів, який ілюструє, що навчання є більш ефективним, коли навчальні сесії розподілені на певні проміжки часу. Іншими словами нові й складні картки показуються частіше, тоді як старіші та легші картки показуються рідше [28, с. 72]. Метод набув широкої популярності в програмних застосунках, відповідно існує багато його реалізацій, включаючи ті, що підлаштовуються під користувача за допомогою нейронних мереж. Незважаючи на це, однією з найпопулярніших реалізацій цього методу є система Лейтнера, через свою простоту та ефективність [28, с. 72].

У методі Лейтнера картки розподіляються на кілька груп, при цьому їхню кількість визначає людина. Кожна група визначає наскільки добре людина знає вміст карток, що у ній знаходиться. Користувач намагається запам'ятати інформацію на кожній картці. Якщо відповідь на питання у картці вірна, то картка переходить до наступної групи. В іншому випадку картка повертається до першої групи для подальшого вивчення. Процес вивчення вмісту карток розподіляється на сесії. Часовий інтервал між сесіями встановлюється користувачем, як правило, день. У першій сесії користувач розглядає картки тільки з першої групи, на другій — з першої та другої, на третій — з першої по третю і так далі, доки не будуть пройдені усі групи. Далі цикл повторюється [38, с. 41].

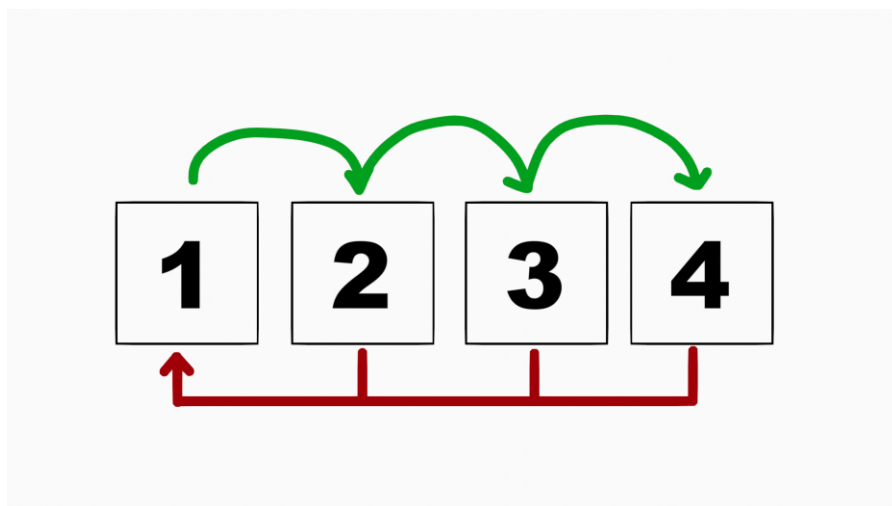


Рисунок 1.1 – Візулізація методу Лейтнера.

Метод Лейтнера мав великий вплив на програмне забезпечення для запам'ятовування інформації, включаючи популярні мобільні додатки, такі як Quizlet, Anki та інші.

1.3. Значення збирання інформації в процесі вивчення іноземної мови

Збирання інформації відіграє не менш важливу роль, ніж просто знаходження слів у словнику. Наприклад, людина може точно визначити, коли та як вживати певні фрази або слова, якщо вона застала ситуацію або контекст, у яких визначена лексика використовується [33]. До того ж, людина може зібрати набір флеш-карток на потрібну їй тему та згодом скористатися, скажімо, методом Лейтнера, щоб засвоїти зібраний матеріал. Це дозволяє фокусуватися на матеріалах, що дійсно є цікавими та необхідними для людини та відповідно позитивно впливає на мотивацію та процес навчання. Особливо такий підхід ефективний для людей,

Таким чином програмний засіб має забезпечувати швидкий доступ до функцій фото або додавання контенту, щоб дозволити зберігати матеріал, який викликає найбільше асоціацій.

1.4. Існуючі програмні засоби для вивчення іноземних мов

На сьогоднішній день існує велике різноманіття різних програмних рішень для вивчення іноземних мов, кожен з яких реалізує свій підхід та методи навчання:

Duolingo — один з найпопулярніших програмних засобів, який пропонує гейміфіковане вивчення мови. Увесь матеріал, що надає додаток, ділиться на рівні від простого до складного. Чудово підходить для початківців, але втрачає ефективність з досвідом користувача [18, с. 105].

Rosetta Stone — популярний програмний засіб, схожий до Duolingo, але робить акцент на використанні лексики без перекладу на мову користувача. Також надає навчальні матеріали, такі як подкасти, розмовники тощо. Підходить для людей, які надають перевагу візуальному навчанню та інтерактивному підходу. Важливою особливістю є також фокус на вимові та аудіюванні, що допомагає розвивати комунікативні навички [18, с. 107].

WordUp — програмний засіб для вивчення англійської мови зі штучним інтелектом, що фокусується на вивченні необхідної лексики на основі вподобань користувача. Додаток аналізує контент, з яким працює користувач, і пропонує слова та фрази, що використовуються в повсякденному житті, завдяки чому навчання стає більш актуальним і ефективним. Він також містить функції повторення для закріплення матеріалу.

Busuu — надає можливість спілкування з носіями мови через інтерактивні уроки. Користувачі можуть отримувати фідбек на свої вправи від інших учнів та викладачів, що сприяє покращенню навичок спілкування.

Memrise — програма, що фокусується на запам'ятовуванні лексики та фраз через візуальні асоціації та ігрові елементи. Memrise також пропонує відео з носіями мови, що допомагає звикнути до акцентів та живого мовлення.

Quizlet — інструмент для онлайн-навчання, який дозволяє користувачам створювати та вивчати флеш-картки, ігри та вікторини. Він

широко використовується для вивчення різних предметів, що робить його популярним серед студентів. Користувачі можуть створювати власні навчальні набори або отримати доступ до мільйонів наборів, створених іншими [38, с. 17]. На відміну від додатків вище, підходить для усіх категорій користувачів, але фокусується виключно на флеш-картках.

Таким чином, вибір програми залежить від особистих уподобань, цілей навчання та стилю навчання кожного користувача, але питання розширення фразового й словникового запасу завжди залишиться актуальним, навіть після проходження курсів, що надаються додатками.

1.5 Відео-матеріали у вивченні мови

Відео-матеріали грають важливу роль в процесі вивчення іноземної мови, оскільки вони дозволяють слухати живу мову в контексті, що допомагає у розвитку навичок аудіювання та підвищує рівень розуміння.

Перегляд відео, таких як фільми, серіали, новини чи навчальні ролики, дозволяє студентам збагачувати словниковий запас, оскільки вони стикаються з розмовною лексикою і різними акцентами. Це також сприяє кращому засвоєнню граматики та інтонації, завдяки таким феноменам як пасивна практика та ефект хамелеона, тобто коли людина або група людей адаптується до поведінки, манер, або мовлення іншої особи, копіюючи жести, міміку, позу або рухи іншої людини без свідомого наміру, підсвідомо підлаштовуючись під її стиль спілкування [38, с. 93].

Проте для новачків перегляд відео може бути викликом, оскільки фонетичні особливості мови, до яких людина ще не звикла, можуть ускладнити сприйняття на слух, відповідно деякі мобільні додатки та платформи впроваджують можливості вивчення мови по фільмах з субтитрами. Зазвичай такі платформи пропонують обмежений каталог фільмів, що може не задовольняти вимоги або смаки всіх користувачів. Крім того, не кожен готовий витратити свій час на перегляд фільмів, які не відповідають його вподобанням або очікуванням.

1.6 Штучний інтелект у вивченні мови

Одним зі значних сучасних технологічних трендів є безумовно штучний інтелект, який не лише інтегрується в існуючі програмні рішення, але й відкриває нові горизонти для розвитку інноваційних сервісів.

На момент написання цієї роботи спостерігається стрімкий розвиток різноманітних програмних рішень, які активно впроваджують та використовують штучний інтелект, включаючи мобільні додатки для вивчення мов. Для прикладу, WordUp використовує ШІ для визначення лексики відповідно до вподобань користувача. Також інколи ШІ впроваджується у деяких реалізаціях алгоритмів інтервального повторення (див. 1.2), згаданому вище, для визначення пріоритетів карток [43].

Ці задачі можна вирішити простими лінійними нейронними мережами, але в контексті нашого дослідження найбільш перспективною архітектурою нейронних мереж можна вважати трансформер. Трансформери відзначаються своєю здатністю ефективно обробляти послідовності даних завдяки механізму уваги, що дозволяє моделі фокусуватися на різних частинах входу, що робить трансформери особливо потужними для вирішення задач, пов'язаних із обробкою природних мов, комп'ютерним зором, обробкою аудіо тощо [26, с. 291]. У контексті вивчення іноземних мов трансформери використовуються для багатьох цілей, починаючи від перекладу тексту, закінчуючи імітацією реальної розмови з іноземцем, через що такі сервіси-трансформери як ChatGPT, Gemini та інші користуються популярністю серед людей для практики. Відповідно, спостерігається тенденція до впровадження цих сервісів у функціонал програмних рішень для вивчення мов.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА ДИЗАЙН ПРОГРАМНОГО ПРОДУКТУ

2.1 Проблематика та завдання застосунку

Як було зазначено у вступі до цієї роботи, головною проблемою, на якій робиться акцент дослідження — це пошук ефективних рішень для швидкого занурення людини в необхідну іноземну лексику, більше орієнтуючись на її професійну або іншу діяльність. Звідси виникає питання створення такого інструменту, який би не лише слугував засобом запам'ятовування матеріалу, але й активно сприяв би розумінню структури та механізмів функціонування мови, навіть для початківців, які не мають достатнього словникового запасу, щоб у повній мірі зрозуміти написане або почуте. Тому кінцеве програмне рішення має стати своєрідним допоміжним засобом, який компенсуватиме недостачу словникового запасу на початкових етапах вивчення за допомогою вбудованих функцій, поглиблюватиме його на пізніх, забезпечуватиме користувача інструментами збирання, аналізу лексики, тексту або зображення, у випадку, якщо людині потрібно отримати назву об'єкта, який вона спостерігає.

Водночас необхідно забезпечити зручність і простоту використання інструменту, щоб людина сильно не відволікалася від своєї діяльності, при зборі матеріалу, а також мала змогу налаштувати програму під свої індивідуальні потреби. Це може включати налаштування частоти повторення слів, набору слів для вивчення для вивчення або категорії матеріалів, які відповідають конкретній сфері діяльності користувача. За допомогою персоналізації можна створити оптимальні умови для активного засвоєння зібраної лексики.

2.2 Цільова аудиторія програмного засобу

З огляду на те, що додаток розробляється як засіб для збирання, організації та засвоєння фраз і слів на іноземних мовах, важливо підкреслити, що для досягнення максимальної ефективності його використання користувачу необхідно активно взаємодіяти з матеріалами, представленими на іноземних мовах. Це може включати читання статей, перегляд відео, прослуховування аудіозаписів, участь у мовних іграх тощо, що сприятиме глибшому зануренню в мовне середовище. Чим більше людина контактує з різноманітними джерелами інформації, тим швидше вона зможе розширити свій словниковий запас та покращити своє розуміння мови [38, с. 40]. Таким чином, регулярна практика і активна участь у мовних процесах є ключовими факторами для ефективного збирання та засвоєння матеріалу.

Якщо проаналізувати досвід представників різних професій, стає очевидним, що максимальний ефект від використання додатка буде досягатися, зокрема, у представників ІТ-сфери. Це зумовлено їхньою постійною необхідністю читання та вивчення великої кількості документації, а також спілкуванням з іноземними колегами та партнерами. Відповідно збирання лексики та поповнення словникового запасу є одним з ключових питань.

Також ефект помітний і в освітній сфері, особливо для студентів та освітніх працівників. У процесі навчання або написання статей вони часто стикаються з необхідністю вивчення матеріалів різних іноземних авторів, що є критично важливим для написання наукових статей та виконання дослідницьких проєктів.

Отже, можна стверджувати, що додаток підійде для ІТ-спеціалістів, студентів, освітніх працівників та інших людей, яким необхідно вивчати та запам'ятовувати лексику у контексті, адже вони значно спрощують процес навчання та підвищують ефективність професійної діяльності.

2.3 Користувацький досвід та збирання інформації

Для кращого та подальшого розуміння структури моделі потенційної реалізації програмного засобу, спробуємо визначитися з тим, як буде відбуватися процес взаємодії користувача та самим програмним засобом.

Перший базовий аспект, який слід розглянути, — це процес збирання лексики, іншої інформації та як він буде впроваджуватися. У контексті цієї роботи під терміном «збирання» мається на увазі наповнення бази даних програмного засобу інформацією з різних джерел, такою як слова, фрази, фото та пов'язання їх з атрибутами, до яких можуть відноситися інші синоніми, скорочення, фото, об'єкти на фото тощо. Під терміном «джерело», в свою чергу, розуміється будь-яка компонента комп'ютерної або мобільної програми, з якої можна отримати текстову або візуальну інформацію. Джерелами можуть виступати сторінки в браузері, текстові документи, фото тощо.

Відповідно, є потреба у побудові зручного інтерфейсу для додавання нової інформації. Крім того, слід також врахувати один з основних елементів мети цієї праці — поєднання вивчення іноземної мови з роботою або іншою діяльністю. З цього виходить, що процес збирання інформації не має сильно відволікати користувача від його повсякденної діяльності та надавати простий набір інструментів, що можуть бути викликані за потреби для швидкого зчитування та занесення інформації у базу даних.

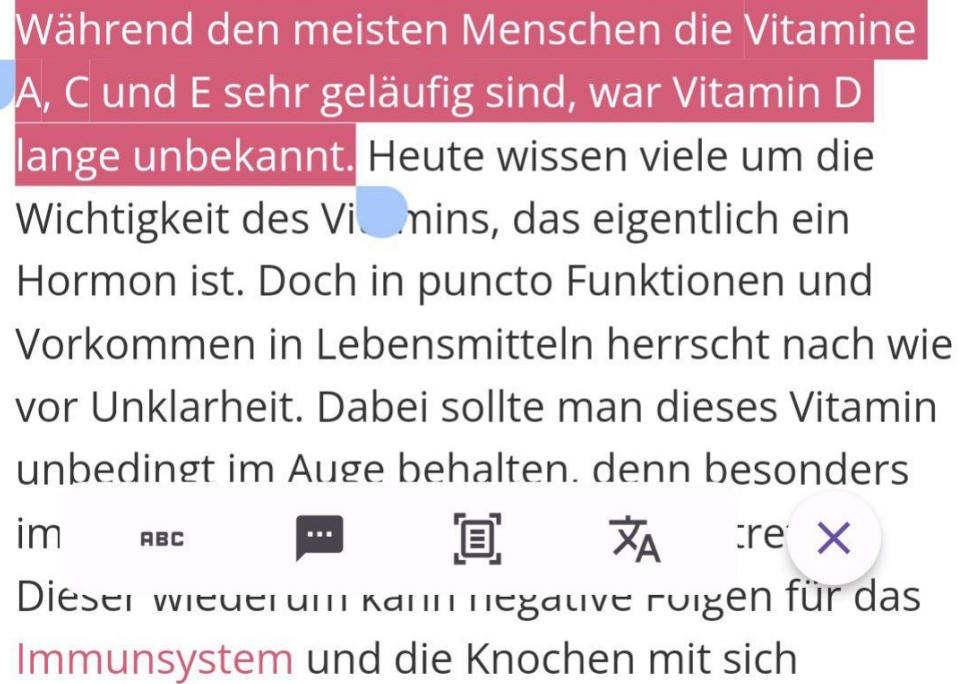
Як було сказано раніше, програмний засіб також повинен компенсувати малий словниковий запас й допомагати користувачеві у процесі адаптації до мови (див. 2.1), тому також є потреба у впровадженні інструментів для перекладу тексту й для вивчення принципів функціонування мови. З розвитком сучасних технологій та сфери ШІ це завдання можна вирішити із залученням сучасних сервісів-трансформерів (див. 1.7), які можуть бути використані для якісного перекладу текстів, пояснення граматики, синтаксису тощо. Тому, користувач за потреби може

викликати інструменти пояснення, якщо він стикнеться із незрозумілими аспектами іноземної мови.

Для повноти розуміння процесу збирання контенту, розглянемо ситуацію, де користувач вперше стикається з невідомою лексикою, читаючи технічну або іншу літературу. Через специфіку термінології та складність матеріалу, користувач може відчувати труднощі у розумінні тексту, тому він може скористатися вбудованим перекладачем, щоб зрозуміти зміст деяких речень. Після цього він може зберегти цю лексику, вказавши додаткову інформацію про неї, за потреби, таку як фото, нотатки, й опрацювати цю інформацію пізніше за допомогою інструментів програмного засобу. При цьому, завдяки вбудованому ІІІ, він може отримати інформацію про граматику певного речення. Водночас інструменти повинні бути доступні користувачеві в будь-який час, щоб він міг зберігати та аналізувати невідому лексику, не відкриваючи саме програмне забезпечення, оскільки його відкриття може суттєво відволікати увагу, особливо на таких пристроях, як смартфони та планшети.

Таким чином, користувач не просто перекладає та заносить нову лексику до бази даних, а зустрічає її в контексті, асоціюючи цю лексику з поточною діяльністю та джерелом, де ця лексика зустрілася.

На рисунку нижче можна побачити приклад використання програмного засобу для збирання й перекладу змісту на іноземній мові. У нижній частині рисунка можна побачити оверлей з інструментами для перекладу, додавання фрази, слова, створення скріншота для прив'язування до слів, дозволяючи користувачу орієнтуватися в тексті при мінімальних знаннях мови та паралельно збирати незрозумілу йому лексику для подальшого вивчення, компенсуючи малий словниковий запас.



Während den meisten Menschen die Vitamine A, C und E sehr geläufig sind, war Vitamin D lange unbekannt. Heute wissen viele um die Wichtigkeit des Vitamins, das eigentlich ein Hormon ist. Doch in puncto Funktionen und Vorkommen in Lebensmitteln herrscht nach wie vor Unklarheit. Dabei sollte man dieses Vitamin unbedingt im Auge behalten, denn besonders im Winter wiederholt kann negative Folgen für das Immunsystem und die Knochen mit sich

Рисунок 2.1 – Використання інструментів на прикладі виділеної фрази у браузері Google Chrome на мобільному пристрої.

2.4 Пріоритезація зібраного контенту

У процесі активного користування програмним засобом не виключається ймовірність того, що з часом людині може стати важко орієнтуватися в зібраних матеріалах через зростання їхньої кількості. До того ж, слід також враховувати рівень знань користувача, щоб забезпечити пошук та визначення потенційних матеріалів для вивчення й робити менший фокус на вже засвоєному контенті.

Часткове вирішення цієї проблеми — це використання метода Лейтнера (див. 1.2). Принцип роботи методу Лейтнера полягає в систематичному повторенні інформації з використанням флеш-карток, які розподіляються по групах залежно від рівня засвоєння матеріалу. Картки, які вивчаються найменше або викликають труднощі, повторюються частіше, тоді як ті, що засвоєні добре, вивчаються рідше, що можна використати для пріоритезації матеріалу на основі знань користувача [40].

Знання користувача можна оцінити за допомогою тестів, відповіді на питання яких буде впливати на пріоритет контенту або лексики, що зустрічалися у тесті. Детальніше про тест та його формування — у наступному підрозділі.

Окрім оцінки знань, слід також враховувати частоту зустрічі лексики або іншого контенту, що також робить внесок в оцінку загального пріоритету певного матеріалу.

На початковому етапі вивчення мови оцінка частоти зустрічі лексики є вкрай важливим аспектом, оскільки дозволяє визначити базову лексику, вивчення якої дозволить орієнтуватися й розуміти основний зміст у реченнях на іноземній мові.

Для визначення частоти зустрічі лексики доцільно зв'язати цей процес із процесом її збирання. Для прикладу, коли користувач повторно додає певне слово — його частота зустрічі збільшується, що дозволяє прив'язати оцінку частоти до уподобань та преференцій користувача.

Отже, маємо два критерії, що впливають на пріоритет — знання користувача та частота зустрічі контенту. Окреме використання цих критеріїв не може повністю задовільнити розрахунок пріоритету найбільш перспективної лексики у повному обсязі, оскільки є висока ймовірність того, що один критерій виявиться важливішим за інший з часом. Тому доцільно внести ще один критерій, який поєднуватиме особливості двох попередніх.

Третій критерій можна визначити як лінійну інтерполяцію між нормалізованими значеннями двох критеріїв. Нормалізація робиться з метою усунення домінування числових значень одного критерію над іншим при інтерполяції. Слід також зазначити, що у формулі нормалізований критерій знань має бути інвертованим, щоб акцент робився саме на тій лексиці, що користувач знає найменше. Формулу для розрахунку числового значення третього критерію наведено на рисунку нижче:

$$(1 - \alpha) \cdot u(a) + \alpha \cdot [1 - v(b)]$$

Рисунок 2.2 – Формула розрахунку загального пріоритету.

Змінні a , b описують відповідні критерії частоти зустрічі та знань користувача Функції u та v – функції нормалізації для даних критеріїв.

Наявність у формулі коефіцієнту інтерполяції дозволяє задавати пріоритет одного критерію над іншим, тому користувач може визначитися, якому критерію він буде надавати перевагу. Для прикладу, нульове значення коефіцієнту враховує тільки частоту слів, в свою чергу, при одиниці враховуються тільки знання користувача. За замовчування коефіцієнту доцільно присвоїти значення 0.5. Таким чином, ми можемо визначати найбільш перспективну лексику для вивчення, зберігаючи певний компроміс між знаннями користувача та частотою зустрічі відповідної лексики.

2.5 Тестування та оцінка знань користувача

У попередньому підрозділі ми частково розглянули питання оцінювання знань й згадали про використання методу Лейтнера для отримання числового значення критерію, однак не вдавалися у подробиці, як саме відбувається цей процес. Незважаючи на те, що метод Лейтнера розрахований на роботу з флеш-картками (див. 1.2), його можна успішно використовувати для тестових завдань, розглядаючи їх як більш широке представлення флеш-карток.

Сам метод Лейтнера ґрунтується на принципі повторення матеріалу з певною періодичністю, що залежить від того, наскільки успішно було засвоєна лексика у флеш-картці. Ключовим аспектом методу є те, що лексика з флеш-картки, на яку користувач відповів правильно, переміщується в більш віддалену групу, що означає, що воно повторюється рідше. Натомість, завдання, на яке відповідь була неправильною, потрапляє в групу, де його слід повторювати частіше.

Саме групу можна трактувати, як критерій знань — чим більший номер групи, тим краще користувач знає цю лексику. Групу можна представити беззнаковим цілим числом, що безпосередньо впливає на загальний пріоритет цієї лексики (див. 2.4). Відповідно, це число можна модифікувати у процесі тестування. Наприклад, правильні відповіді збільшують це число, переміщуючи цю лексику до наступної групи, тоді як неправильні — присвоюють йому нульове значення. Водночас, відповіді можуть бути частковими, що може впливати на групу по-іншому, для прикладу — переносити відповідну лексику на кілька кроків назад, в залежності від правильності відповіді на тестове запитання.

Окрім власне оцінювання знань, виникає низка інших питань, які потребують уваги: як проводити тести, як визначати їхнє наповнення, які типи запитань включати.

Визначившись із критеріями оцінювання пріоритету вище, ми можемо скористатися цим підходом для отримання вибірки певної лексики, що буде зустрічатися під час тесту. Сама вибірка може бути використана для створення тестових запитань, але слід також уникати дублювання запитань, які вже були згенеровані й визначити оптимальний розподіл різних видів питань, що будуть зустрічатися під час тесту.

Щодо розподілу питань, то кожному типу питання можна присвоїти певну вагу, тобто ймовірність, з якою воно буде з'являтися у тесті.

Після проходження тесту критерії знань лексики оновлюються залежно від відповідей на питання тесту.

2.6 Зв'язки між контентом

Незважаючи на те, що основною лексичною одиницею будь-якої мови є слово, їхнє окреме вивчення не завжди є хорошою практикою, оскільки значення слів можуть набувати різних форм та функцій в залежності від контексту. Слова можуть комбінуватися у фрази, речення та в складні конструкції, утворюючи більш великі мовні одиниці, такі як речення. Крім

того, речення можуть містити різні граматичні та інші конструкції, такі як пасивний стан, відмінки тощо, що може вимагати створення нотаток для пояснення цих явищ.

Відповідно у програмному засобі слід передбачити можливість формування певних зв'язків між різними видами контенту, що дозволить людині отримати не тільки інформацію про слово, але й про контекст, де воно використовується, об'єкт, що воно описує, синоніми, форми слова тощо.

Такий підхід дозволяє представити весь сегмент контенту бази даних програмного засобу у вигляді своєрідного графу, який користувач може редагувати, збираючи новий контент або доповнюючи його додатковими асоціативними зв'язками, а також використовувати ці зв'язки для надання одиницям контенту додаткових атрибутів, які можуть бути використані інструментами програмного засобу в їхній роботі.

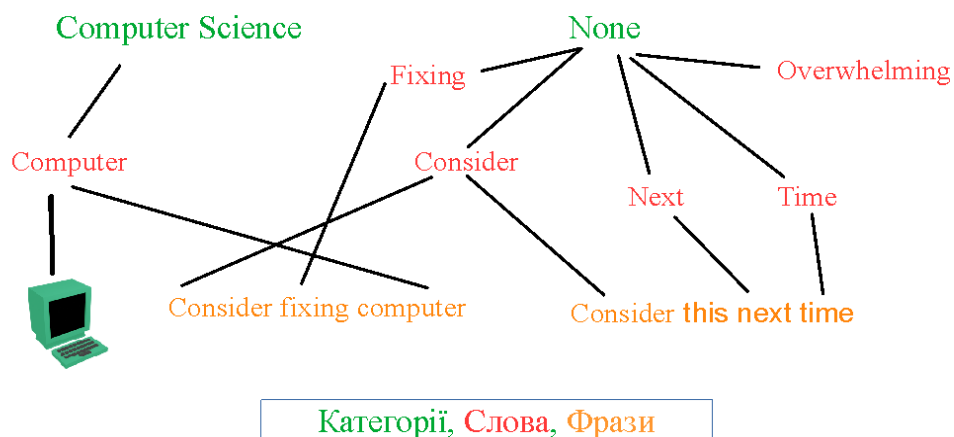


Рисунок 2.3 – Приклад формування асоціативних зв'язків між словами, об'єктами та категоріями.

Як видно з рисунку вище, зв'язки також можуть встановлюватися до категорій, об'єднуючи та групуєючи різні типи контенту за темою, що дозволяє здійснювати зручніший пошук потрібної зібраної лексики та формувати більш структуровані тести, що спираються не тільки на пріоритет слів (див. 2.5), але й на тему, до якої відноситься контент. Також контент може не відноситися до жодної категорії. У такому випадку всім матеріалам

присвоюються категорія «None» для легкого знаходження некласифікованого контенту.

Визначивши теоретичний базис щодо зв'язків, для подальшого дослідження стає важливим окреслити методологію їхнього формування. Більша частина зв'язків визначається через взаємодію між користувачем й інструментами програмного засобу, в той час як деякі зв'язки формуються автоматично. До автоматичних видів зв'язків можна віднести зв'язок типу «Слово-Речення», що створюється при занесенні нового слово або речення БД, в свою чергу, користувацьких відносяться зв'язки типу «Об'єкт-Слово», «Об'єкт-Фото», що створюються користувачем при редагуванні об'єктів на фото, приклад чого можна побачити на рисунку 2.4.

Окрім того, що зв'язки допомагають у навігації контентом, вони також корисні для генерації тестів, а отже, і для генерації вибірки лексики (див. 2.5), що також включає розширену інформацію про її вміст, що дозволяє створювати більш складні запитання. Ці питання можуть включати вибір об'єкта, зображеного на фото, завдяки зв'язкам типу «Об'єкт-фото», пошук антонімів (зв'язок «Слово-антонім») тощо.

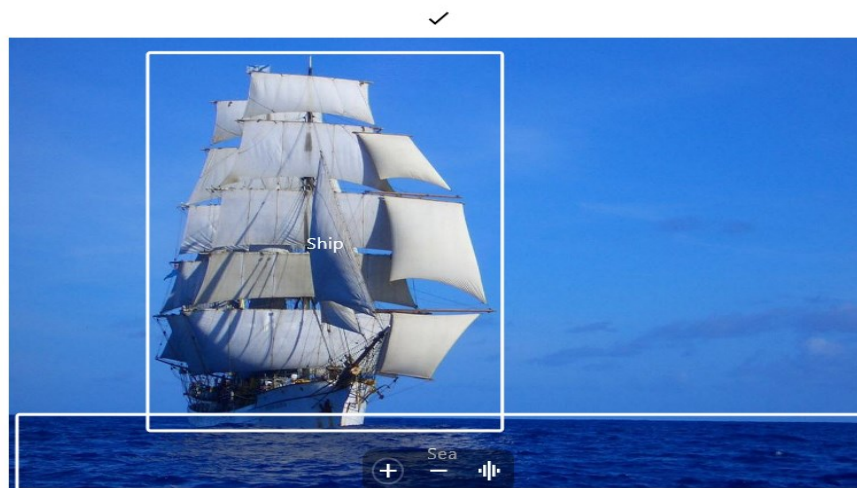


Рисунок 2.4 – Вікно перегляду фото після встановлення зв'язків між об'єктами, словами та самим фото.

Отже, такий підхід до організації контенту підхід дозволить краще сприймати та запам'ятовувати зібрані матеріали, впроваджуючи механізм формування різних видів зв'язків між різними типами контенту та надаючи додаткову інформацію для програмного засобу про зібраний контент.

2.7 Інтегровані пошукові та генеративні сервіси

Беручи до уваги природу програмного засобу, очікується, що більшість зібраного контенту буде представлено у вигляді текстових даних без будь-яких додаткових атрибутів та зв'язків. Користувач, як правило, не стане шукати в Інтернеті фото або опис для великого об'єму зібраної лексики, оскільки це займе багато його часу.

Тому програмний засіб має передбачати та впроваджувати засоби для автоматизації пошуку та генерації різних типів контенту, таких як, фотографії. Завдяки автоматизованому пошуку та створенню контенту програмне забезпечення може значно підвищити ефективність та зручність користування ним.

Самі сервіси можуть базуватися на різних пошукових системах або ШІ [25, с. 13], що дозволяє виконувати такі завдання, як пошук або генерація візуального представлення слова, генерація нотаток для пояснення граматичних аспектів речення тощо. Впровадження таких сервісів дозволить користувачам легко пов'язуватиме відповідні зображення з матеріалами БД.

Додатково такі сервіси можуть також допомогти користувачеві з отриманням лексики з фото, у випадках, коли йому необхідно взнати інформацію про об'єкт, що він спостерігає, що актуально для мобільних пристроїв через їхню портативність та наявність у них камери.

Таким чином, автоматизовані сервіси дозволяють значно заощадити час користувачів на пошук візуальних та навчальних матеріалів в Інтернеті та формувань відповідних зв'язків між лексикою та завантаженими фото, дозволяючи їм більше зосередитися на вивченні лексики, тим самим підвищуючи свою загальну продуктивність.

2.8 Прослуховування вимови

У кожній мові існують специфічні фонетичні та інтонаційні одиниці, які відрізняються не лише на рівні індивідуальних звуків, але й на рівні більш складних акустичних явищ, таких як інтонація, ритм, акцентуація, темп мовлення та паузи. Ці особливості мови можуть суттєво варіюватися від мови до мови, що значною мірою ускладнює процес розуміння та правильного відтворення мовних структур для носіїв інших мов. Наприклад, деякі звуки, що є типовими для однієї мови, можуть бути відсутніми або вимовлятися зовсім інакше в інших мовах, що створює труднощі у навчанні правильній вимові та сприйнятті на слух.

У цьому контексті, впровадження технологічних сервісів для прослуховування вимови є важливим елементом пріоритетом, оскільки дозволить адаптувати слух користувача пам'ять до специфічних звуків та вимови іноземної мови. Такими сервісами виступають так звані Text-To-Speech (TTS), які можуть бути вбудованими в програмний засіб.

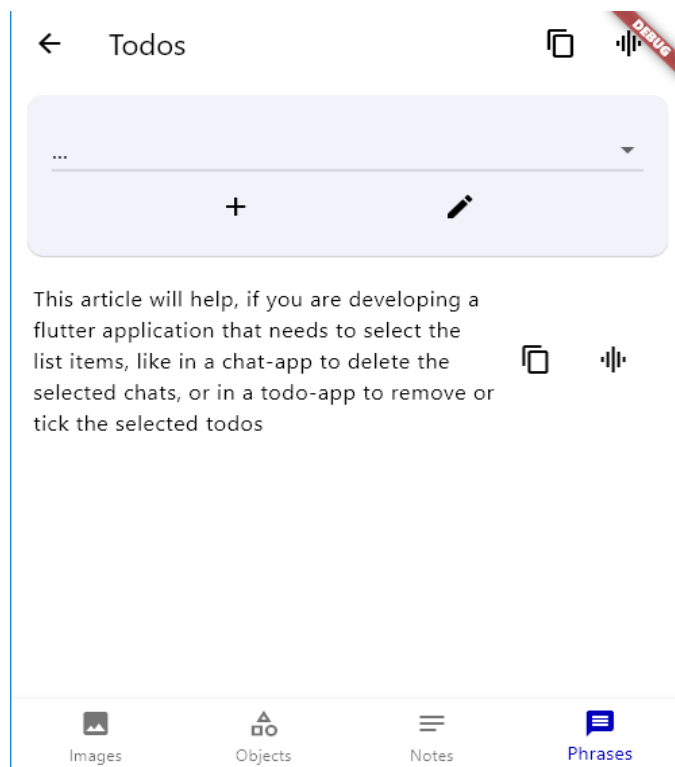


Рисунок 2.5 – Сторінка фраз зі словом "Todos" з кнопками виклику TTS у верхньому правій частині сторінки та біля фрази.

З розвитком III технологія TTS значно просунулася вперед, забезпечуючи більш природну та точну вимову, що також може допомогти з вивченням певних діалектів різних мов, що особливо актуально для німецької та китайської мов через відсутність єдиного стандарту для вимови деяких звуків та інших інтонаційних особливостей.

Відповідні кнопки виклику цих сервісів можна розміщувати безпосередньо коло міток зі словами або фразами для зручного прослуховування вимови, як показано на рисунку вище. Додатково таку можливість слід передбачити й в інтерфейсі оверлею програмного засобу (рисунок 2.1) для можливості прослуховуванні невідомих фраз при читанні будь-яких текстових матеріалів.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОТОТИПУ ПРОГРАМНОГО ПРОДУКТУ

3.1. Фреймворк Flutter Dart та кросплатформність

Для кодової бази проєкту було обрано фреймворк Flutter Dart. Це рішення було зумовлене кількома ключовими факторами, такими як швидкість створення інтерфейсів користувача та мультиплатформність, тобто можливість створювати додатки, які працюють на різних платформах, таких як iOS, Android Mac, Windows, з єдиної кодової бази, що значно спрощує та прискорює розробку. Іншим фактором є функція «Hot reload», яка дозволяє розробникам вносити зміни в код під час відлагодження й спостерігати зміни у самому додатку [39, с. 21].

Крім того, Flutter пропонує багатий набір віджетів, плагінів й надає просту у використанні систему збирання програмного забезпечення, що автоматизує процес ручного встановлення пакетів та загалом процесу компіляції в нативний код [23, с. 71].

Ще одним важливим аспектом є простота мови Dart, яка дозволяє розробникам швидко освоїти її і почати ефективно працювати. Dart підтримує об'єктно-орієнтоване програмування, що сприяє організації коду та його підтримці в майбутньому.

Але слід також зазначити й недоліки. Через те, що фреймворк Flutter прагне забезпечити єдину кодову базу для всіх платформ, повністю уникнути використання платформозалежних API неможливо. Це пояснюється тим, що деякі розширення та плагіни для Flutter залежні від виклику сторонніх сервісів або функцій, які надаються безпосередньо самою платформою. Наприклад, для реалізації специфічних функцій, таких як доступ до сервісів TTS, геолокації, апаратних ресурсів, системних сповіщень або камери Flutter може використовувати API, доступні тільки в рамках певної операційної системи (iOS, Android тощо) [39, с. 29].

Отже, у процесі реалізації прототипу програмного засобу буде використовуватися саме фреймворк Flutter Dart для зменшення складності

портування на інші платформи та ОС. Таким чином єдиною складністю буде реалізація деяких функцій, які не можуть бути однаково реалізовані для усіх платформ. Іншими словами, такі компоненти, як оверлей будуть реалізовані по-різному для користувачів ПК та для користувачів мобільних пристроїв через різницю у роботі з віконною системою та іншими системними компонентами, що відповідають за взаємодію людини з ОС.

3.2 Визначення СУБД

Однією з головних компонент програмного засобу — це безумовно база даних, де зберігається весь зібраний контент.

Враховуючи той факт, що програмний засіб передбачає збирання інформації, він також має передбачати її пошук за певними атрибутами й зв'язками. У таких випадках доцільно використовувати бази даних SQL [12, с. 60].

За замовчуванням програмний засіб використовує локальну реляційну база даних SQLite, що є оптимальною для додатків, які потребують швидкого доступу до даних без необхідності налаштування серверної інфраструктури. SQLite була обрана саме через свою простоту у використанні та високу продуктивність для невеликих і середніх обсягів даних. Вона не вимагає додаткового серверного програмного забезпечення, що спрощує налаштування та запуск програми. База даних SQLite зберігається в одному файлі, що робить її зручною для використання в середовищах, де немає потреби в розподіленому зберіганні даних та спрощує процедуру відлагодження й редагування таблиць й самої структури БД, що дозволить легко адаптувати конфігурацію SQLite до інших БД. Попри це, SQLite підтримує всі основні можливості SQL, включаючи транзакції, індекси та складні запити, що забезпечує достатню гнучкість для більшості типових задач з обробки даних. Це робить її ідеальним вибором для програм, які повинні працювати на різних платформах, зберігати дані локально та мати високу продуктивність при мінімальних витратах на ресурси. [12, с. 71]

Для комунікації із базою даних програмний засіб передбачає загальний інтерфейс, який дозволяє працювати не лише з SQLite, а й з іншими популярними SQL-базами даних, такими як MySQL, PostgreSQL чи Microsoft SQL Server, що дозволить легко змінити базу даних у подальшій розробці, якщо у цьому виникне необхідність.

Таким чином, SQLite обрана як оптимальне рішення для більшості випадків завдяки своїй простоті та ефективності, але в той же час програмний засіб забезпечує гнучкість в плані налаштувань, підтримуючи інтеграцію з іншими базами даних для більш складних і масштабованих сценаріїв. Це дозволяє адаптувати програму до різних інфраструктур і робочих процесів.

3.3 Планування та реалізація бази даних

Визначивши основні особливості та функції програмного засобу в другому розділі, ми переходимо до реалізації бази даних. Враховуючи той факт, що програмний інструмент працює з різними типами контенту, важливо спроектувати базу даних таким чином, щоб вона була гнучкою і масштабованою. Це означає використання модульного дизайну, який підтримує додавання нових типів даних і зв'язків між ними (див. 2.6) без необхідності значної реструктуризації. Реляційна система управління базами даних (СКБД) часто підходить для цієї мети завдяки своїй здатності обробляти складні запити і зв'язки за допомогою чітко визначених таблиць і зовнішніх ключів.

Перше, з чого можна почати — це визначення таблиць, що будуть описувати сутності, тобто типи контенту. Посилаючись на попередній розділ, ми можемо виокремити наступні сутності:

1. Слово;	5. Фото;
2. Фраза;	6. Об'єкт на фото;
3. Нотатка;	7. Зв'язок;
4. Категорія;	8. Користувач.

Відповідно, у базі даних має налічуватися не менше 8 таблиць, що описують кожну з цих сутностей. Тепер спробуємо визначитися, які поля мають містити таблиці цих сутностей.

Розглянемо сутність «Слово». Перше поле, яке спадає на думку — це сам вміст слова, тобто текстове поле. Друге поле — це ID, як правило, числового типу, що буде виступати головним ключем до запису в таблиці.

Хоча, на перший погляд, може здаватися, що текстове поле може бути первинним ключем — це не так, оскільки може бути декілька користувачів, кожен з яких вивчає різні мови. Ці мови, в свою чергу, можуть включати однакові слова, що може викликати колізії, що перешкоджатимуть вставці нових слів у БД.

Третє та четверті поля можна виділити під критерії частоти та знань, тобто два числових типи, на основі яких можна розрахувати загальний пріоритет цього слово (див. 2.4).

Наступне поле — це дата створення та дата останньої зустрічі слова, що дозволяє користувачу визначити лексику, яка довго не повторювалася.

Останнє поле — ID профілю користувача, якому належить це слово.

Наступна сутність — фраза. Через схожість цієї зі словом, доцільно включити поля слова у таблицю фраз. Відмінністю від таблиці слова буде те, що фраза виключає у себе наявність прапорців, що ідентифікують, чи фраза є фразеологізмом, словосполученням тощо та поля з даними у форматі JSON для збереження додаткової інформації про частини мови тощо.

Наступна сутність — це нотатка. Має схожу структуру до фрази або слова, але не включає поля пріоритетів, дати останньої зустрічі. Має поля теми та видозмінене поле текстової інформації з підтримкою HTML для формування зв'язків з іншими нотатками, фразами тощо.

Розглянемо наступну сутність — категорію. Представляє собою просту таблицю з трьома полями — ID, назва та профіль.

Таблиця, що описує фото. Складається з ID, імені фото, та стиснутих даних самого фото у форматі PNG, що зберігається у полі з типом даних

BLOB. Має відсутнє поле профілю користувача, що дозволяє повторно використовувати фото з інших користувацьких профілів.

Об'єкти на фото, в свою чергу, описуються прямокутниками (рисунок 2.4) із заданими позицією та розміром, полем з ID фото, до якого вони прив'язуються, та самого ID цього прямокутника.

Таблицю зв'язків можна представити чотирма полями — ID профілю користувача, тип зв'язку, ID першої сутності, ID другої сутності. Такий підхід дозволяє зберігати зв'язки між будь-якими сутностями та об'єктами без необхідності створення додаткових таблиць.

Сутність профілю користувача можна описати трьома полями — полем ID, назвою профілю та текстовим полем у форматі JSON для зберігання налаштувань профілю, статистики тощо.

Відповідно, структура БД матиме вигляд, який представлено на рисунку 3.1.

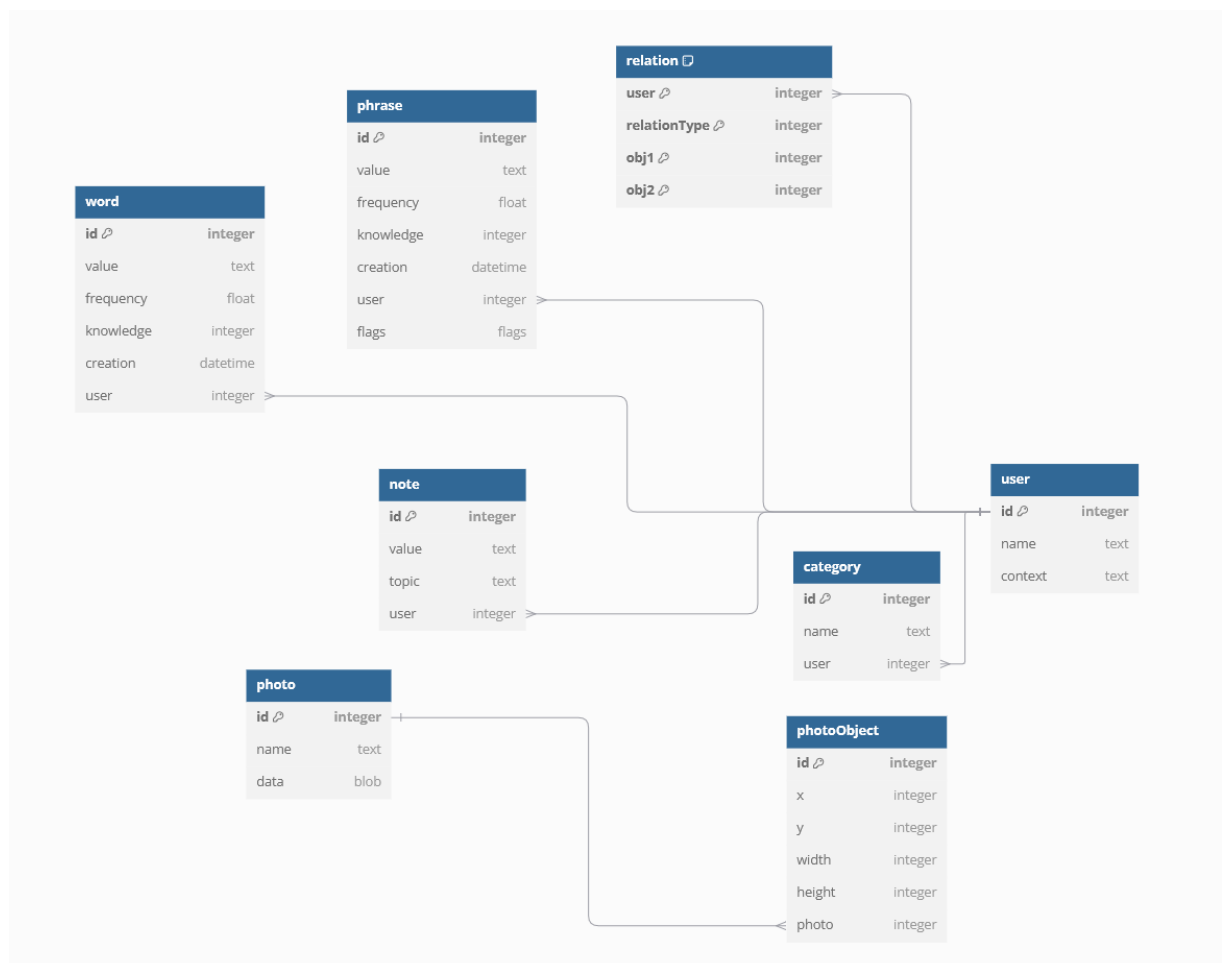


Рисунок 3.1 – Структура БД програмного засобу.

3.4 Впровадження пошукових та генеративних сервісів

У другому розділі ми поверхнево розглядали можливість впровадження сервісів пошуку та генерації додаткового контенту, але не вдавалися у деталі їхньої реалізації.

Усього існує два способи впровадження таких сервісів. Перший полягає у використанні спеціалізованих API, що надаються різними пошуковими системами, для прикладу, Google надає великий спектр різних API через платформу Google Cloud для різних цілей, починаючи від пошукових сервісів до розпізнавання образів. З переваг — простота використання та інтеграції в проєкт. Недоліком таких API, в свою чергу, є те, що вони, як правило, платні і не кожен може дозволити їхнє використання в довгостроковій перспективі.

Другий спосіб — це вебскрейпінг. Спосіб полягає в імітації поведінки людини, при доступі до певних Інтернет-ресурсів з метою їх зчитування та перетворення у форму, сприятливу для кінцевої програми. Недоліком такого підходу є те, що перед розробником стоїть проблема реалізації парсера сторінки та емуляція користувацьких дій. У більшості випадків, в програмному засобі використовується другий спосіб, хоча деякі сервіси реалізуються через API.

Для прикладу, один з сервісів програмного засобу використовує результати пошукової системи DuckDuckGo. Це досягається за допомогою відправлення GET-запиту із заголовками, характерними для певних браузерів на адресу пошукової системи для отримання HTTP-cookie та вмісту сторінки. У реалізації програмного засобу з заголовки GET-запиту виглядають наступним чином.

Після цього на адресу пошукового сервісу відправляється POST-запит, що імітує завантаження деякого набору зображень. Результати цього запиту згодом переводяться у формат, зрозумілий самій програмі. У нашому випадку — це масив об'єктів класу «NetImageInfo», структуру якого наведено на рисунку нижче. У першому полі зверху зберігається посилання

```

var initReqUri = Uri.https(
  'duckduckgo.com', '/', {'q': value, 't': 'ffab', 'ia': 'web'});

var headers = {
  'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8',
  'Accept-Encoding': 'gzip',
  'Accept-Language': 'en-US,en;q=0.5',
  'DNT': '1',
  'Referer': 'duckduckgo.com',
  'Sec-Fetch-Dest': 'empty',
  'User-Agent':
    'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0'
};

var initReq = await http.get(initReqUri, headers: headers);

```

Рисунок 3.2 – Заголовки та процедура відправлення GET-запиту в реалізації програмного засобу.

на мініатюру фотографії, в другому — посилання на саме фото, в третьому зберігається посилання на сайт, у полі «title» зберігається назва фотографії.

```

class NetImageInfo {
  final String thumbnail;
  final String image;
  final String url;
  final String title;

  NetImageInfo(this.title, this.image, this.thumbnail, this.url);
}

```

Рисунок 3.3 – Клас «NetImageInfo».

Об’єкти даного класу згодом передаються відповідним віджетам фреймворка Flutter, що потім відображають знайдені фотографії.

Генеративні сервіси, в свою чергу, реалізуються на основі API.

Усього в програмному засобі налічується два генеративних сервіси — генерація фото на основі тексту та аналізатор граматички на основі ChatGPT.

Перший сервіс впроваджується на основі вебскрейпінгу з сайту нейронної мережі DeepAI за схожою процедурою до отримання фотографій з пошукових систем. Сервіси ChatGPT, в свою чергу, впроваджуються на основі API. Відповідно, для його використання необхідно отримання спеціального токена доступу з сайту OpenAI. Після цього його необхідно

передати в заголовку POST-запиту під назвою «Authorization», вказати у тілі запиту назву моделі, текст, на який потрібно сформувавши відповідь, максимальну кількість токенів, та відсоток випадковості вихідного результату. Зробивши це, ми можемо впровадити цей сервіс в сам програмний засіб, дозволяючи користувачу отримати інформацію про структуру, граматику фраз, як це наведено на рисунку нижче на прикладі речення на німецькій мові.

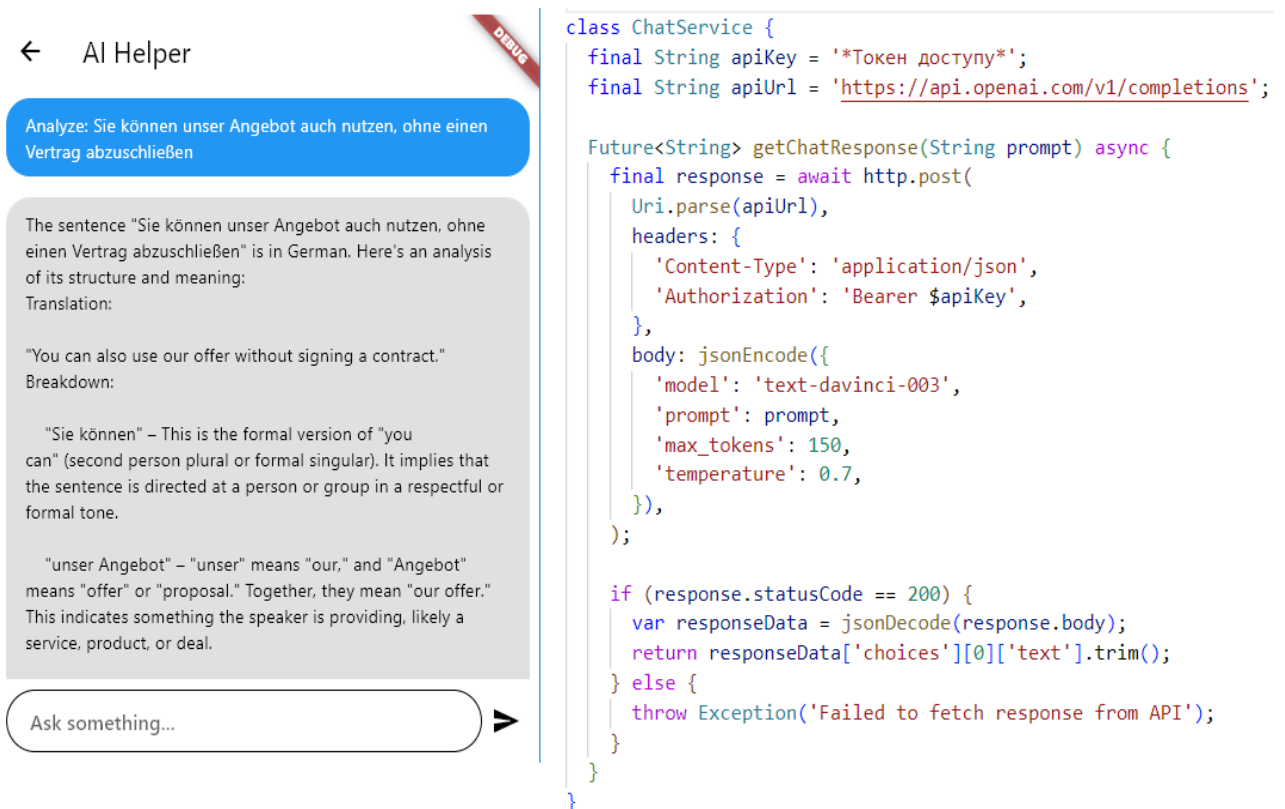


Рисунок 3.4 – Реалізація сервісу аналізатора речень та AI-помічника на його основі.

3.5 Оверлей та особливості його реалізації

Оверлей — це одна з головних функцій програмного засобу, завдяки якій користувач може додавати контент без необхідності відкриття головного інтерфейсу програми.

З точки зору реалізації оверлей представляє собою окремий програмний модуль, прив'язаний до головної програми, що надає спрощений

інтерфейс користувача, який відображається поверх інших програм, й керує відображенням та позиціонуванням цього інтерфейсу.

Основна складність реалізації оверлею полягає в його платформозалежності. Для прикладу, на платформі Android потрібно враховувати той факт, що у користувачів, як правило, мале розширення екрану, що не дозволяє розміщувати велику кількість елементів інтерфейсу поверх інших програм, оскільки це буде сильно заважати користувачу. До того ж, у Flutter не передбачено універсального рішення, що підтримує оверлей для мобільних пристроїв та ПК, що створює певні складності. Якщо розглядати реалізацію оверлею на ПК, то ситуація трохи краща, оскільки більшість операційних для них є багатовіконними, відповідно реалізація оверлею спрощується до головного вікна самого програмного засобу, що може легко позиціонуватися, сильно не заважаючи користувачеві.

У програмному засобі оверлей реалізовується виключно для мобільних пристроїв, оскільки його використання на ПК стає аналогічним до головного вікна програми.

3.6 Деталі впровадження TTS

TTS – це спеціальний сервіс, що відповідає за перетворення текстової інформації в мовленнєву, що є важливим компонентом у вивченні іноземної мови, оскільки дозволяє прослухати звучання фрази або слова в різних контекстах. Існує безліч реалізацій TTS, кожна з яких використовує різні алгоритми та методи для перетворення тексту на природне мовлення. Ці реалізації можуть суттєво відрізнятися за якістю голосу, мовною підтримкою та здатністю працювати з різними акцентами й діалектами. Сучасні системи TTS використовують передові моделі машинного навчання та нейронні мережі для створення високореалістичного і схожого на людське мовлення, що робить їх корисними у вивченні будь-яких мов [44, с. 49].

Одним зі способів впровадження сервісів TTS можна вважати вище згадані API, що дозволяють синтезувати мовлення за допомогою різних

моделей нейронних мереж. Недоліком таких API є необхідність використання Інтернету та ціна.

Реалізація TTS у програмному засобі спирається на встановленні мовні пакети та сервіси, що є доступними для більшості платформ. Це рішення зумовлене тим, щоб дозволити користувачу встановити мовний пакет, націлений на певний діалект мови або говір, у якому він зацікавлений. Конкретна реалізація TTS в програмному засобі досягається за допомогою бібліотеки «Flutter TTS», що дозволяє зчитувати інформацію про встановленні мовні пакети та сервіси мовлення й виконувати синтез мовленнєвого сигналу за допомогою цих сервісів.

3.7 Демонстрація роботи прототипу

Проведемо тестування програмного засобу на прикладі німецькомовної статті про фреймворк Flutter у Вікіпедії з точки зору користувача, незнайомого з німецькою мовою (рисунок 3.5).

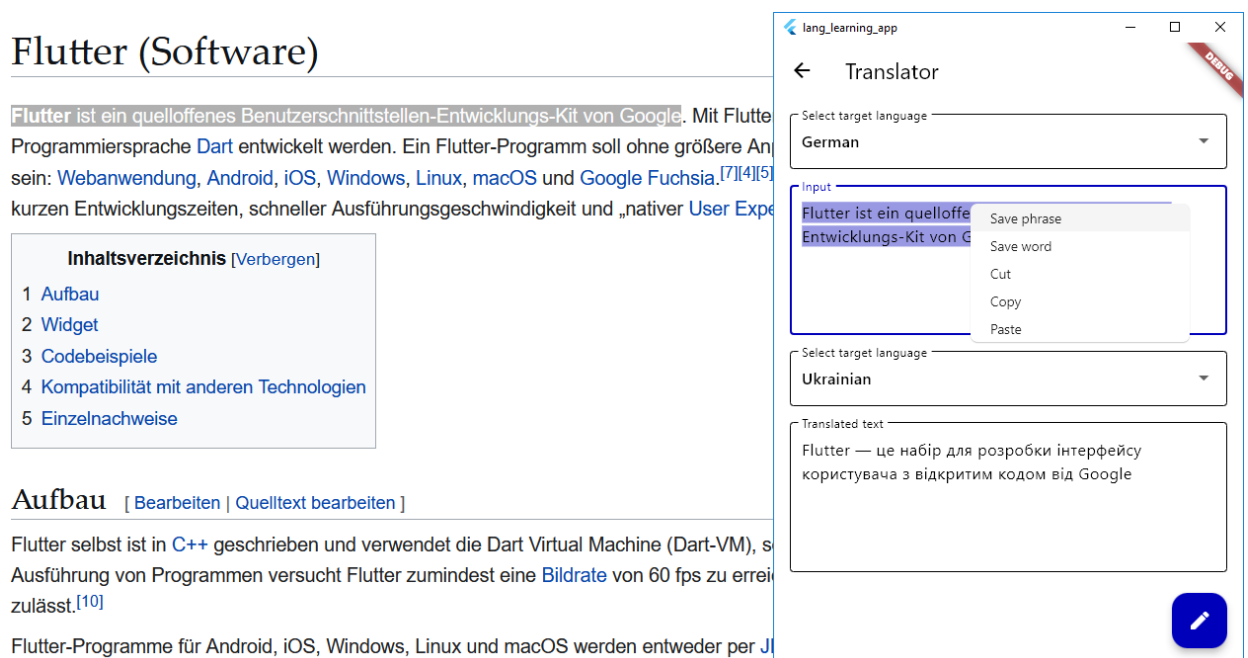


Рисунок 3.5 – Вікно програмного засобу, що відображається поверх вікна браузера.

Перше, що необхідно зробити для користування програмним засобом — це створити новий профіль, де будуть зберігатися усі зібрані матеріали. Після цього можна перейти до читання самої статті.

З чим користувач стикнеться на початку, так це з незрозумілою лексикою та незнанням правил функціонування мови. Тому, ймовірно, спочатку користувач скористається перекладачем, щоб зрозуміти написане. У комп'ютерній версії програмного засобу вікно перекладача викликається комбінацією клавіш «CTRL+SHIFT+T». Дане вікно відображається поверх інших вікон. Згодом користувач може вставити виділений текст у відповідне текстове поле для перекладу, як наведено на рисунку 3.5.

Далі відповідну фразу можна зберегти, виділивши її та обравши пункт «Save phrase» у контекстному меню. Після чого відкриється сторінка редагування фрази, де до неї можна прикріплювати фотографії, виділяти об'єкти на фотографіях (як це було продемонстровано на рисунку 2.4), робити нотатки, дивитися переклад, задавати категорії тощо. Скажімо, користувач захотів швидко знайти фотографії для цього речення в Інтернеті, тому він переходить у вкладку «Images», натискає кнопку додавання та вибирає один з чотирьох варіантів: камера, сховище, пошуковий сервіс, генеративний ШІ. У нашому випадку це пошуковий сервіс (рисунок 3.6).

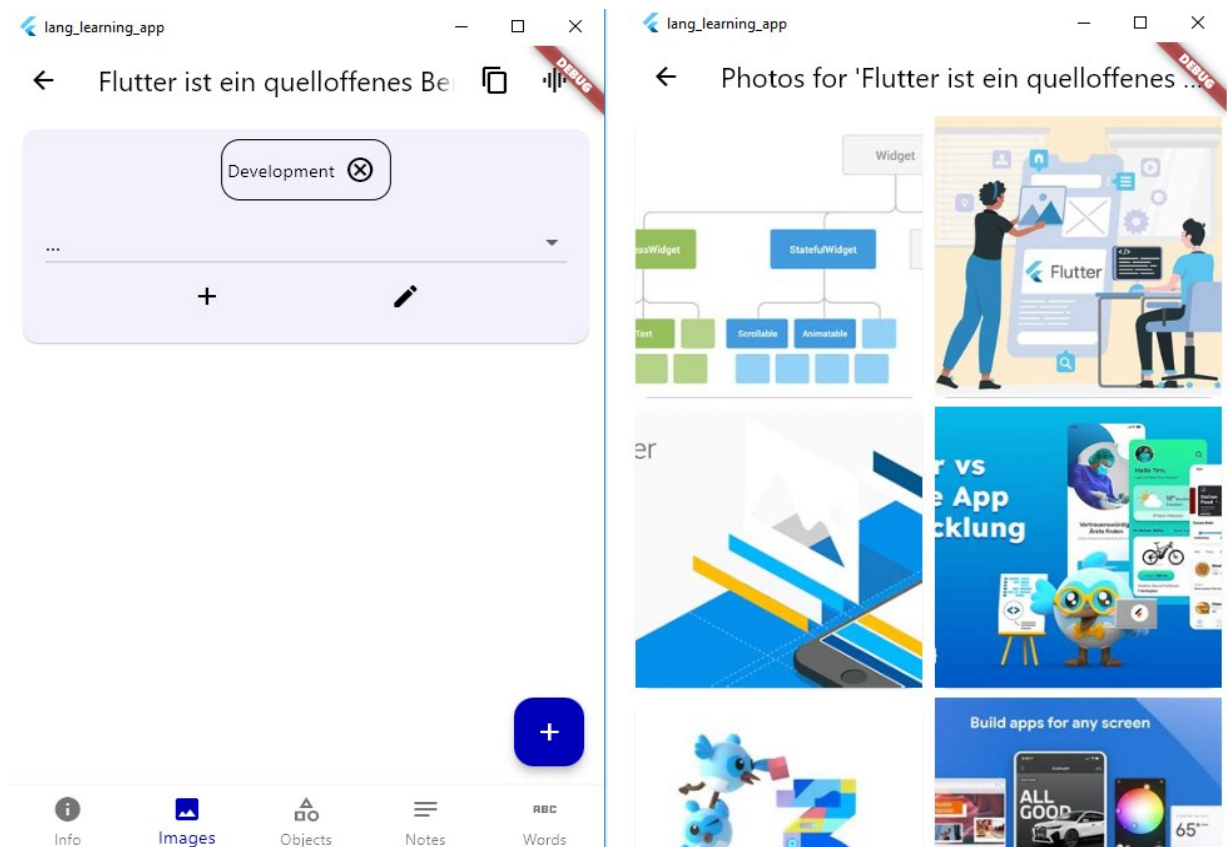


Рисунок 3.6 – Вікна програмного засобу зі сторінками редагування фрази та пошуку зображень в Інтернеті.

Після цього можна вибрати фотографію, яка згодом прикріпиться до фрази. Незважаючи на те, що користувач знає переклад фрази, він все ще не розуміє, як функціонує сама мова. Для цього він може скористатися AI-помічником з метою аналізу речення та його структури. Згенерований текст згодом можна винести в нотатки та прикріпити їх до речення на сторінці редагування фраз, згаданої вище.

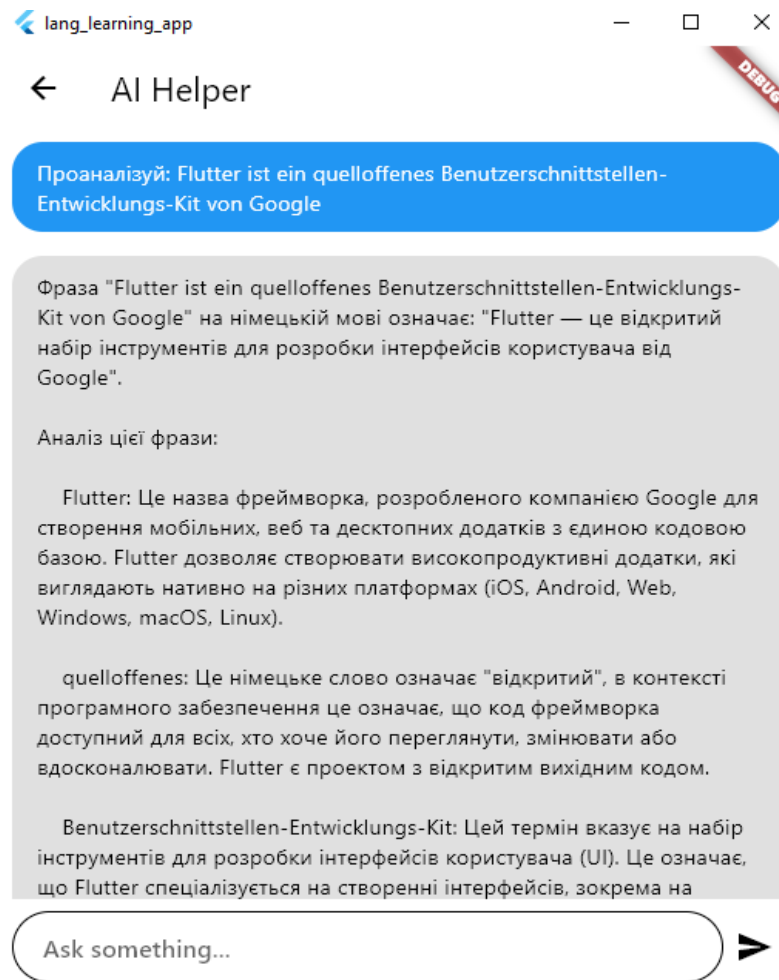


Рисунок 3.7 – Використання AI-помічника для аналізу речення.

Із речення, в свою чергу, можна виокремити слова та прикріпити до них відповідну інформації у вигляді фото, категорій тощо аналогічно процедурам, описаним вище.

Виконавши аналогічні процедури для усіх інших речень у статті, що цікавлять користувача, можна переходити до тесту (рисунок 3.8). Тест складається з усіх матеріалів, що зібрав користувач, та проходиться, як правило, у вільний час.

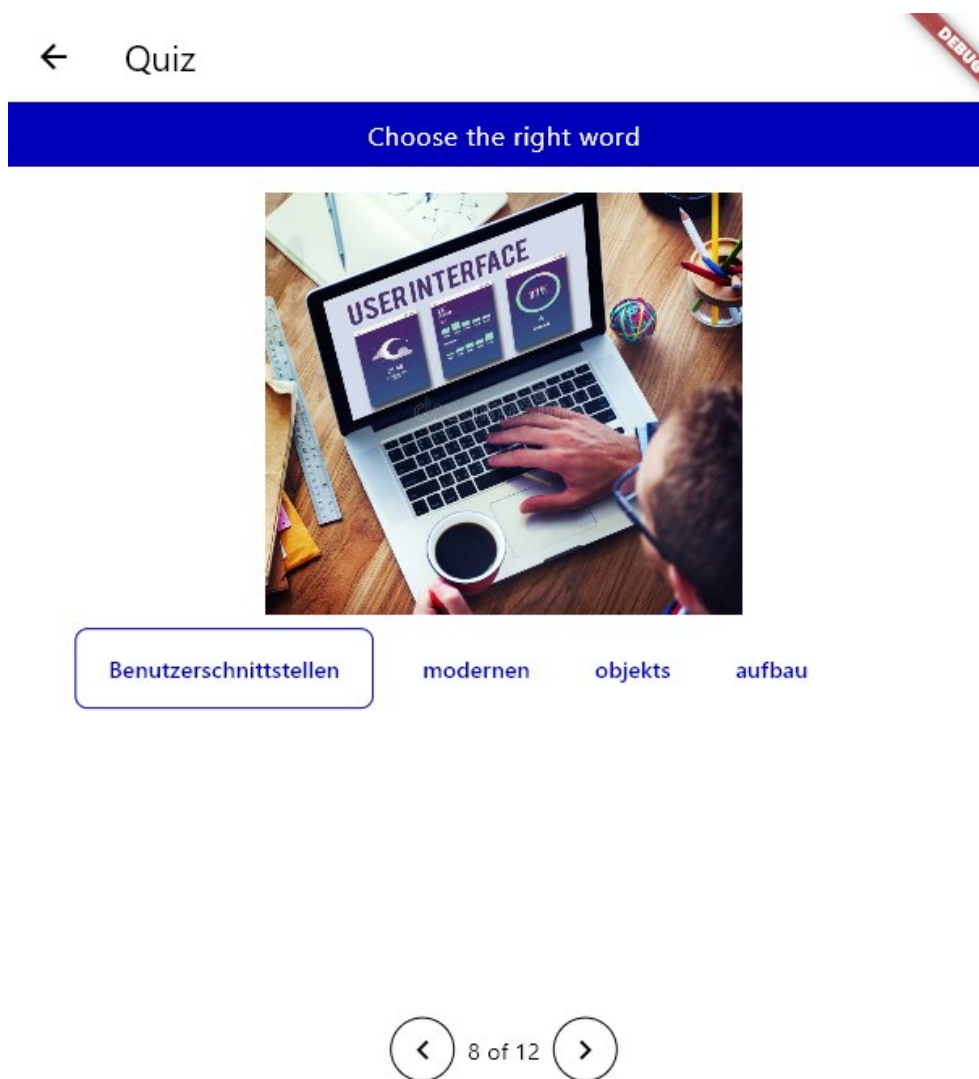


Рисунок 3.8 – Приклад згенерованого питання на основі зібраних даних зі статті про Flutter.

Отже, продемонстровані на реальному прикладі можливості програмного засобу дають змогу прогнозувати успішне і ефективне його використання при вивченні іноземних мов.

ВИСНОВКИ

У кваліфікаційній роботі розв'язана науково-технічна задача створення методів і засобів для підвищення ефективності процесу запам'ятовування іноземних слів та фраз на основі використання розробленого спеціалізованого програмного засобу, що дозволяє формувати асоціативні зв'язки між словами, фразами, зображеннями та іншими фрагментами інформації із залученням пасивної практики. В тому числі отримано такі основні результати.

1. Проаналізовано сучасні методи активізації процесу запам'ятовування інформації, зокрема іноземних слів.
2. Проведено аналіз сучасних методів добування та збереження різних форм навчальної інформації та способів розширення словникового запасу.
3. Розроблено проєкт та дизайн програмного рішення у вигляді спеціалізованого застосунку.
4. Розроблено базовий функціонал програмного інструменту для професійного розвитку, збирання та засвоєння матеріалу на іноземних мовах. Запропоновані способи збирання лексики та іншого контенту зробили процес його накопичення максимально зручним та інтуїтивним.
5. Створено діючий прототип програмного застосунку, в якому втілено нові підходи до архітектури та реалізації програм і додатків для вивчення іноземної мови.

На практиці застосування розробленого програмного застосунку дає змогу користувачам легко інтегрувати вивчення іноземної мови у свою повсякденну діяльність. Акцентуючи увагу на пасивній практиці та формуванні прямих асоціацій між новою лексикою, професійними або повсякденними завданнями користувача, використання функціоналу програмного продукту оптимізує процес навчання та заощаджує час для

активного вивчення іноземної мови. Такий підхід значно відрізняється від традиційних рішень, пов'язаних з вивченням іноземних мов, що у більшості випадків фокусуються виключно на запам'ятовуванні інформації або дотримуються визначеної програми курсів або модулів.

Дослідження включало концептуалізацію та початковий дизайн програмного рішення, планування користувацького досвіду та загальної архітектури системи. Ці компоненти втілено у функціональний прототип кросплатформного програмного засобу, що допомагає користувачу адаптуватися до матеріалів з іноземної мови, зібрати та визначити перспективну лексику, вивчення якої дозволить краще розуміти ті матеріали, які цікаві користувачу.

Подальше дослідження методів та засобів інформаційної підтримки процесу вивчення іноземних мов, розглянутих у цій роботі, може в деякій мірі вплинути на архітектуру та підхід до реалізації програм та додатків для вивчення іноземної мови.

Отже, в підсумку, поставлену мету було досягнуто та виконано усі завдання дослідження.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Іванців Ю. Замітки Поліглота. Практичні поради з вивчення іноземної мови. Lulu Press, Incorporated, 2021. 312 с.
2. Ковриго С. Б. Про використання прислів'їв та приказок у навчальному процесі. Англійська мова та література. 2020. № 34–36 (404–406). С. 37–42.
3. Жолтовський О. О. Програмна агрегація контенту для вивчення іноземної мови. Тези доповідей 10 Міжнародної наукової конференції «Сучасні проблеми математичного моделювання, прогнозування та оптимізації» Кам'янець-Подільський: К-ПНУ, 2024. URL: https://cs.kpnu.edu.ua/wp-content/uploads/2024/07/optima_2024-tezdop_1.pdf (дата звернення: 09.11.2024).
4. Жолтовський О. О. Методи і засоби інформаційної підтримки процесу вивчення іноземної мови. Вісник Кам'янець-Подільського національного університету імені Івана Огієнка. Фізико-математичні науки. Кам'янець-Подільський: К-ПНУ, 2023. Вип. 18. С. 32-34. URL: <http://elar.kpnu.edu.ua:8081/xmlui/bitstream/handle/123456789/7681/Visnyk-K-PNU-im.-I.-Ohienka.-Fiz.-mat.-nauky.-Vyp.16-2023.pdf?sequence=3&isAllowed=y> (дата звернення: 09.11.2024).
5. Жолтовський О. О. Розробка програмного засобу інформаційної підтримки процесу вивчення іноземної мови. Вісник Кам'янець-Подільського національного університету імені Івана Огієнка. Фізико-математичні науки. Кам'янець-Подільський: К-ПНУ, 2024. Вип. 18. С. 32-34. URL: <http://elar.kpnu.edu.ua:8081/xmlui/bitstream/handle/123456789/7681/Visnyk-K-PNU-im.-I.-Ohienka.-Fiz.-mat.-nauky.-Vyp.16-2023.pdf?sequence=3&isAllowed=y> (дата звернення: 10.11.2024).
6. Жолтовський О. О. Application to organize and prioritize content for effective language learning. Вісник Кам'янець-Подільського національного університету імені Івана Огієнка. Фізико-математичні науки. Кам'янець-

Подільський: К-ПНУ, 2024. Вип. 18. С. 32-34.
 URL: <http://elar.kpnu.edu.ua:8081/xmlui/bitstream/handle/123456789/7681/Visnyk-K-PNU-im.-I.-Ohiiienka.-Fiz.-mat.-nauky.-Vyp.16-2023.pdf?sequence=3&isAllowed=y> (дата звернення: 10.11.2024).

7. Онлайн-документація для Windows API [Електронний ресурс]. URL: <https://learn.microsoft.com/en-us/windows/> (дата звернення: 25.11.2024).

8. Онлайн-Документація для віконної системи X [Електронний ресурс]. URL: <https://learn.microsoft.com/en-us/windows/> (дата звернення: 25.11.2024).

9. Al-Shahmeh M. M. Flutter Apps Development: Build Cross-Platform Flutter Apps with Trust. Independently published, 2023. 270 с. ISBN 979-8396111912.

10. Alberto Miola. Flutter Complete Reference 2.0: The ultimate reference for Dart and Flutter / Alberto Miola. [S.l.] : Independently published, 2023. 826 с. ISBN 979-8394957390.

11. Barbara Bregstein. Complete Spanish Step-by-Step, Premium Second Edition / Barbara Bregstein. 2-ге вид. – New York : McGraw Hill, 2020. 624 с. ISBN 978-1260463132.

12. Basil U. App Development with Flutter: Creating Cross-Platform Mobile Apps with Flutter / Basil U. [S.l.] : Independently published, 2024. 138 с. ISBN 979-8332753053.

13. Behrouz A. Forouzan. Data Communications and Networking with TCP/IP Protocol Suite ISE / Behrouz A. Forouzan. 6-е вид. New York : McGraw Hill, 2021. 864 с. ISBN 978-1260597820.

14. Binary Bard, Aether Infektion. Networking Protocols Unleashed: Exploring the Depths of TCP/IP, DNS, and More / Binary Bard, Aether Infektion. [S.l.] : Charles M Dominquez, 2024. 110 с. ISBN 978-1806253050.

15. Bryan Sills, Brian Gardner, Kristin Marsicano, Chris Stewart. Android Programming: The Big Nerd Ranch Guide / Bryan Sills, Brian Gardner, Kristin

Marsicano, Chris Stewart. 5-е вид. Boston : Addison-Wesley Professional, 2022. 688 с. ISBN 978-0137645541.

16. Catalin Ghita. Kickstart Modern Android Development with Jetpack and Kotlin: Enhance your applications by integrating Jetpack and applying modern app architectural concepts / Catalin Ghita. [S.l.] : Packt Publishing, 2022. 472 с. ISBN 978-1801811071.

17. Contextual Language Learning / ред.: Y.-J. Lan, S. Grant. Singapore. Springer Singapore, 2021. 202 с. DOI: <https://doi.org/10.1007/978-981-16-3416-1> (дата звернення: 25.10.2024).

18. Daily Language Learning. German: The Ultimate Guide for Beginners Who Want to Learn the German Language, Including German Grammar, German Short Stories, and Over 1000 German Phrases / Daily Language Learning. [S.l.] : [b. i.], 2019. 514 с.

19. Daria Orlova, Esra Kadah, Jaime Blasco. Flutter Design Patterns and Best Practices: Build scalable, maintainable, and production-ready apps using effective architectural principles / Daria Orlova, Esra Kadah, Jaime Blasco. [S.l.] : Packt Publishing, 2024. 362 с. ISBN 978-1801072649.

20. David Thomas, Andrew Hunt. The Pragmatic Programmer: Your Journey to Mastery, 20th Anniversary Edition / David Thomas, Andrew Hunt. 2-ге вид. Boston : Addison-Wesley Professional, 2019. 352 с. ISBN 978-0135957059.

21. Dawn Griffiths, David Griffiths. Head First Android Development: A Learner's Guide to Building Android Apps with Kotlin / Dawn Griffiths, David Griffiths. 3-е вид. Sebastopol : O'Reilly Media, 2021. 930 с. ISBN 978-1492076520.

22. Erman Baltürk. Advanced Flutter Development / Erman Baltürk. [S.l.] : [b. i.], 2023. 76 с. ASIN B0BYJZTG69.

23. Temidayo Adefioye. Ultimate Flutter for Cross-Platform App Development: Build Seamless Cross-Platform Flutter UIs with Dart, Dynamic

Widgets, Unified Codebases, and Expert Testing Techniques / Temidayo Adefioye. [S.l.] : Orange Education Pvt Ltd, 2024. 416 с. ISBN 978-8196994709.

24. Tom Long. Good Code, Bad Code: Think like a software engineer / Tom Long. [S.l.] : Manning, 2021. 376 с. ISBN 978-1617298936.

25. Tony Gaitatzis. Learn REST APIs: Your guide to how to find, learn, and connect to the REST APIs that powers the Internet of Things revolution / Tony Gaitatzis. [S.l.] : BackupBrain Press, 2019. 109 с. ISBN 978-1989775004.

26. James Phoenix, Mike Taylor. Prompt Engineering for Generative AI: Future-Proof Inputs for Reliable AI Outputs / James Phoenix, Mike Taylor. 1-е вид. – Sebastopol : O'Reilly Media, 2024. 422 с. ISBN 978-1098153434.

27. Jim Arlow, Ila Neustadt. Generative Analysis: The Power of Generative AI for Object-Oriented Software Engineering with UML / Jim Arlow, Ila Neustadt. 1-е вид. Boston : Addison-Wesley Professional, 2024. 672 с. ISBN 978-0138291426.

28. Jo Franco. Fluentish: Language Learning Planner & Journal / Jo Franco. [S.l.] : Teach Yourself, 2023. 240 с. ISBN 978-1399805926.

29. John Russel. 4 Books in 1: Ultimate Beginner's Guide, 7 Days Crash Course, Advanced Guide, and Data Science, Learn Computer Programming and Machine Learning with Step-by-Step Exercises: Monticello Solutions Limited, 2020. 392 с.

30. John Ousterhout. A Philosophy of Software Design / John Ousterhout. 2-ге вид. [S.l.] : Yaknyam Press, 2021. 196 с. ISBN 978-1732102217.

31. Kaiser P. Fun with Programming Languages. BookBaby, 2021. 684 с.

32. Lingo Mastery. Dutch Made Easy Level 1: An Easy Step-By-Step Approach To Learn Dutch for Beginners / Lingo Mastery. [S.l.] : Lingo Mastery, 2024. 232 с. ISBN 978-1951949747.

33. Lütge C., Merse T. Digital Citizenship in Foreign Language Learning and Teaching. Global Citizenship in Foreign Language Education. New York, 2022.

247 р. DOI: <https://doi.org/10.4324/9781003183839-14> (дата звернення: 22.10.2024).

34. Maaïke van Putten, Seán Kennedy. Java Memory Management: A Comprehensive Guide to Garbage Collection and JVM Tuning. Packt Publishing, 2022. 146 с.

35. Neal Ford, Mark Richards, Pramod Sadalage, Zhamak Dehghani. Software Architecture: The Hard Parts: Modern Trade-Off Analyses for Distributed Architectures / Neal Ford, Mark Richards, Pramod Sadalage, Zhamak Dehghani. 1-е вид. Sebastopol : O'Reilly Media, 2021. 459 с. ISBN 978-1492086895.

36. Peter Späth. Pro Android with Kotlin: Developing Modern Mobile Apps with Kotlin and Jetpack / Peter Späth. 2-ге вид. [S.l.] : Apress, 2022. 1009 с. ASIN B0BQZTSHJZ.

37. Raju Gandhi, Mark Richards, Neal Ford. Head First Software Architecture: A Learner's Guide to Architectural Thinking / Raju Gandhi, Mark Richards, Neal Ford. 1-е вид. Sebastopol : O'Reilly Media, 2024. 483 с. ISBN 978-1098134358.

38. Ryan Johnson. Fearlessly Fluent Fast: Learn Your Next Language Fast Like A Kid!: 7 Simple Neuro-Linguistic Hacks to Speak Your Foreign Language in 3 Months Use Everyday ... & Build Fluency Fast Like Children / Ryan Johnson. [S.l.] : Inceptionspace Publishing, 2022. 176 с. ASIN B0BR2FYG1G.

39. Simone Alessandria. Flutter Cookbook - Second Edition: 100+ real-world recipes to build cross-platform applications with Flutter 3.x powered by Dart 3 / Simone Alessandria. 2-ге вид. – [S.l.] : Packt Publishing, 2023. 712 с. ISBN 978-1803245430.

40. Sushmita Roy. This Is How You Learn a New Language and Never Forget It. The science of language learning, 2022. URL: <https://blog.fluent-forever.com/this-is-how-you-learn-a-new-language-and-never-forget-it> (дата звернення: 05.03.2024).

41. Vickler A. Javascript : This Book Includes: Javascript Basics for Beginners + Javascript Front End Programming + Javascript Back End Programming. Independently Published, 2021. 714. с.

42. Why we can't remember what we learn and what to do about it. interactive.wharton.upenn.edu. URL: <https://interactive.wharton.upenn.edu/learning-insights/why-we-cant-remember-what-we-learn-and-what-to-do-about-it/> (дата звернення: 23.11.2024).

43. WordUp Vocabulary App. URL: <https://www.wordupapp.co/> (дата звернення: 26.11.2024).

44. Xu Tan. Neural Text-to-Speech Synthesis (Artificial Intelligence: Foundations, Theory, and Algorithms) / Xu Tan. [S.l.] : Springer, 2023. 381 с. ASIN B0BXNRTTNF.