

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ПРЯМУХІНА ОЛЕНА-МАРІЯ ДМИТРІВНА

Допускається до захисту:

завідувач кафедри
інформаційних технологій,
д. т. н., доцент

_____ Т.В. Нескородева

« _____ » _____ 2022р.

**РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ВИВЧЕННЯ
АНГЛІЙСЬКОЇ МОВИ**

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:

Нескородева Т. В., д.т.н., доцент
кафедри інформаційних технологій

Оцінка: _____ / _____ / _____

(бали за шкалою ЄКТС/за національного шкалою)

Голова ЕК: _____

(підпис)

АНОТАЦІЯ

Прямухін О.-М. Д. Розробка інформаційної системи підтримки вивчення англійської мови. Спеціальність 122 «Комп'ютерні науки», Донецький національний університет імені Василя Стуса, Вінниця, 2022.

Процес інтеграції українського суспільства в світовий економічний, політичний, соціальний простір потребує вільного володіння членами соціуму мовами країн Євросоюзу насамперед англійською.

Розвиток технологій створює передумови прискорення процесу вивчення іноземних мов, створює базу для практичного застосування набутих знань. У зв'язку з цим основною метою кваліфікаційної роботи є проектування та розробка інформаційної системи підтримки процесу вивчення іноземної англійської мови.

У результаті виконання кваліфікаційної роботи здійснено проектування структури бази даних слів, обраних для вивчення, реалізовано інформаційну систему підтримки їх вивчення та виконано тестування готового продукту.

У кваліфікаційній (бакалаврській) роботі проаналізовано актуальність інформаційних систем підтримки процесу самостійного вивчення англійської мови, здійснено порівняльний огляд існуючих інформаційних систем індивідуального навчання.

Сформовано перелік задач, які має виконувати розроблена інформаційна система підтримки вивчення англійської мови, та вимоги до неї.

На етапі проектування системи використано UML-діаграми для розробки її архітектури, здійснено моделювання поведінки запропонованої системи під час роботи за різними сценаріями.

Для реалізації використано мову програмування Swift, контент-менеджер систему Strapi.io, реляційну СКБД SQLite, систему керування базами даних Realm.

Виконання дослідження виявило ряд питань оптимізації серверної частини, використання більш нових методик вивчення слів, які потребують подальших розробок.

Ключові слова: інформаційна система, архітектурне проектування, моделі поведінки, діаграма класів, діаграма пакетів, нейронна мережа, база даних.

68 с., 1 табл., 46 рис., 1 дод., 41 джерело.

ABSTRACT

Priamukhina O.-M. Developing the information system for support learning English. Specialty 122 "Computer Technologies", Vasul' Stus Donetsk National University, Vinnytsia, 2022.

The process of integration of Ukrainian society into the world economic, political, social space requires fluency of members of society in the languages of the European Union, especially English.

The development of technology creates the preconditions for accelerating the process of learning foreign languages, creates a basis for the practical application of acquired knowledge. In this regard, the main purpose of the qualification work is to design and develop an information system to support the process of learning a foreign English language.

As a result of the qualification work, the structure of the database of words selected

for study was designed, an information system to support their study was implemented, and testing of the finished product was performed.

In the qualification (bachelor's) work the relevance of information systems to support the process of independent learning of English is analyzed, a comparative review of existing information systems of individual learning.

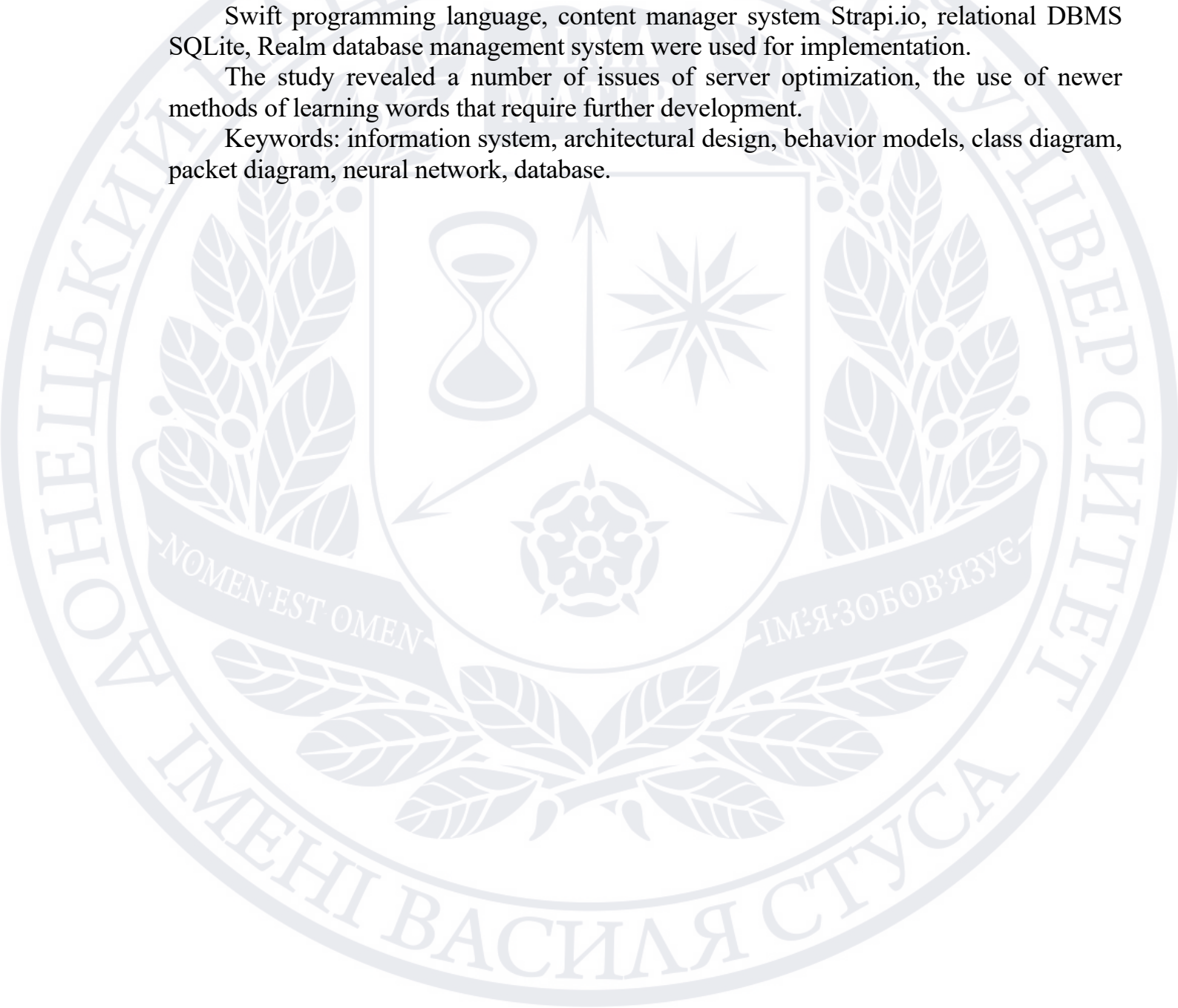
The list of tasks to be performed by the developed information system to support the study of English and the requirements for it has been formed.

At the stage of system design, UML-diagrams were used to develop its architecture, modeling the behavior of the proposed system while working on different scenarios.

Swift programming language, content manager system Strapi.io, relational DBMS SQLite, Realm database management system were used for implementation.

The study revealed a number of issues of server optimization, the use of newer methods of learning words that require further development.

Keywords: information system, architectural design, behavior models, class diagram, packet diagram, neural network, database.

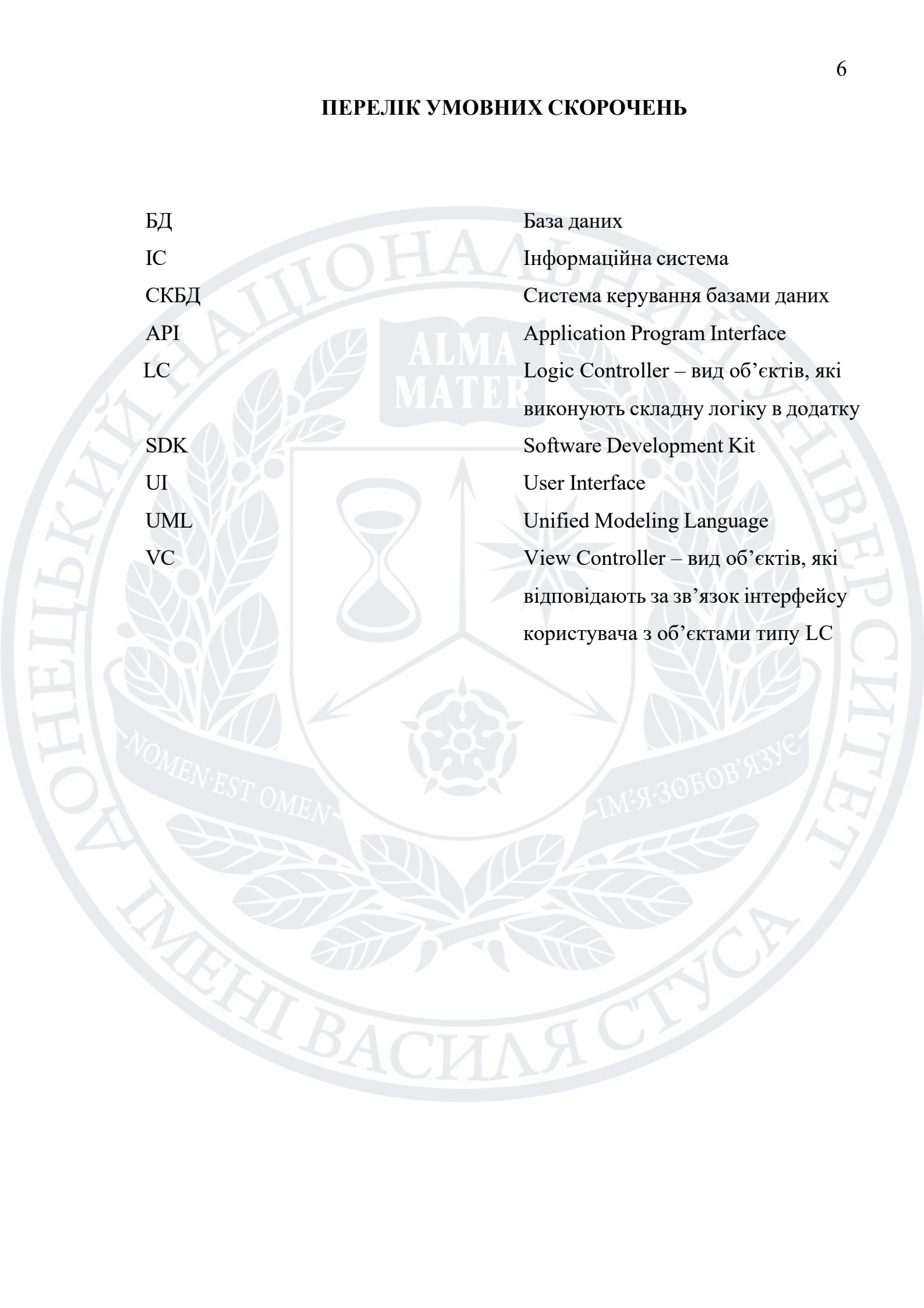


ЗМІСТ

| | |
|---|-----------|
| ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ | 6 |
| ВСТУП..... | 7 |
| РОЗДІЛ 1. ПОРІВНЯЛЬНИЙ ОГЛЯД АНАЛОГІВ СИСТЕМ ПІДТРИМКИ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ..... | 9 |
| 1.1 Область застосування системи підтримки вивчення іноземних мов..... | 9 |
| 1.2 Огляд програмних аналогів підтримки вивчення англійської мови | 11 |
| 1.2.1 Інформаційна система «Memrise» | 11 |
| 1.2.2 Інформаційна система «Duolingo» | 13 |
| 1.2.3 Інформаційна система «Rosetta Stone» | 15 |
| 1.3 Переваги та недоліки існуючих систем вивчення іноземних мов..... | 17 |
| Висновки до розділу 1 | 19 |
| РОЗДІЛ 2. АНАЛІЗ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ | 20 |
| 2.1 Моделювання предметної області, аналіз вимог до ІС підтримки вивчення англійської мови | 20 |
| 2.1.1 Формування та аналіз вимог | 22 |
| 2.1.2 Діаграми прецедентів інформаційної системи | 23 |
| 2.2 Логічна структура ІС підтримки вивчення англійської мови | 25 |
| 2.2.1 Діаграма класів ІС вивчення англійської мови | 25 |
| 2.2.2 Діаграма пакетів ІС вивчення англійської мови | 34 |
| 2.3 Архітектурне проектування..... | 35 |
| 2.3.1 Діаграма модулів ІС вивчення англійської мови..... | 36 |
| 2.3.2 Розгортання ІС підтримки вивчення англійської мови на апаратних засобах | 37 |
| 2.4 Моделювання поведінки ІС підтримки вивчення англійської мови | 39 |
| 2.4.1 Діаграма діяльності ІС..... | 39 |

| | |
|--|----|
| 2.4.2 Діаграма послідовності..... | 44 |
| 2.5 Проектування бази даних ІС підтримки вивчення англійської мови..... | 51 |
| 2.5.1 Концептуальна модель предметної області ІС підтримки вивчення англійської мови | 51 |
| 2.5.2 Логічна модель бази даних ІС підтримки вивчення англійської мови..... | 53 |
| 2.5.3 Фізична модель бази даних ІС підтримки вивчення англійської мови | 54 |
| Висновки до розділу 2..... | 55 |

| | |
|---|-----------|
| РОЗДІЛ 3. РОЗРОБКА І ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ..... | 57 |
| 3.1 Розробка системи..... | 57 |
| 3.1.1 Огляд та вибір мови програмування | 57 |
| 3.1.2 Вибір системи керування базами даних | 59 |
| 3.1.3 Програмне забезпечення використане для розробки інформаційної системи | 62 |
| 3.1.4 Розробка і навчання нейронної мережі для використання в інформаційній системі | 63 |
| 3.1.5 Розробка інформаційної системи підтримки вивчення англійської мови..... | 64 |
| 3.2 Тестування ІС підтримки вивчення англійської мови..... | 72 |
| Висновки до розділу 3..... | 74 |
| ВИСНОВКИ..... | 75 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 77 |

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

| | |
|------|--|
| БД | База даних |
| IS | Інформаційна система |
| СКБД | Система керування базами даних |
| API | Application Program Interface |
| LC | Logic Controller – вид об’єктів, які виконують складну логіку в додатку |
| SDK | Software Development Kit |
| UI | User Interface |
| UML | Unified Modeling Language |
| VC | View Controller – вид об’єктів, які відповідають за зв’язок інтерфейсу користувача з об’єктами типу LC |

ВСТУП

Технології стали невід’ємною частиною життя суспільства. Разом із модернізацією та стрімким розвитком технологій спостерігається також популяризація засвоєння нових навичок, які спрощують сучасне життя. Однією з навичок, яка нині набуває популярності та сприяє досягненню цілей у різних сферах життя, є володіння іноземними мовами, а особливо англійською.

Англійська мова — найпопулярніша у світі. Сьогодні значна частина інформації перш за все публікується саме англійською мовою: написання та подання наукової літератури, міжнародні конференції, семінари і тренінги, ведення наукової документації і т. ін. Володіння англійською мовою відкриває безліч можливостей: спілкування з людьми по всьому світу, можливість перегляду фільмів та читання літератури мовою оригіналу, збільшення шансів для кар’єрного росту, можливість здобуття освіти міжнародного рівня – стають можливими для тих, хто володіє англійською мовою на високому рівні.

Темою дипломної роботи є створення iOS-орієнтованої системи вивчення англійської мови.

Актуальність теми. Володіння англійською мовою дає змогу безперешкодно спілкуватися у будь-якій частині світу, отримувати доступ до іноземних джерел інформації та вивчати культури інших народів світу. Продукт, що є доступним завдяки наявності мобільної версії, а також застосовує різні методики засвоєння нових навичок, полегшує процес вивчення іноземної мови у повсякденному житті.

Мета та завдання дослідження. Метою роботи є проектування та розробка iOS-орієнтованої системи підтримки вивчення англійської мови.

Для досягнення поставленої мети потрібно виконати такі завдання:

- здійснити аналіз предметної області;
- провести огляд наявних аналогів;
- спроектувати інформаційну систему;
- розробити методики для покращення засвоєння здобутих навичок;

- провести тестування додатку.

Об’єкт і предмет розробки. Об’єктом розробки є застосування різних методик спрощення вивчення англійської мови.

Предметом розробки є автоматизація методологій запам’ятовування слів англійської мови та створення програмного додатку для підтримки вивчення англійської мови під iOS.

Практичне значення. Програмний продукт призначений для використання людьми у повсякденному житті не залежно від сфери діяльності. Значення має лише початковий рівень знань та словниковий запас користувача. Отримані знання будуть корисними для поповнення словникового запасу та можуть бути застосовані у повсякденному житті.

Структура та обсяг роботи. Дипломна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел (41 найменування), 46 рисунків, 1 таблиці. Загальний обсяг роботи становить 68 сторінок основного тексту.

РОЗДІЛ 1. ПОРІВНЯЛЬНИЙ ОГЛЯД АНАЛОГІВ СИСТЕМ ПІДТРИМКИ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ

1.1 Область застосування системи підтримки вивчення іноземних мов

Усе живе, що існує сьогодні, пройшло довгий шлях розвитку від найпростіших форм життя до складних за структурою та способами існування систем. Їх тисячолітній розвиток нині характеризується сукупністю захисних та пристосувальних механізмів, сформованих внаслідок потреби віднайти шлях до оптимального співіснування з іншими об'єктами екосистеми. Одним із таких механізмів пристосування є здатність до мовлення.

На початкових етапах свого розвитку представникам людського виду, як іншим тваринам, для обміну інформацією було достатньо мови жестів та сукупності примітивних звуків. Із часом діапазон звуків розширювався, звуки ускладнювалися та формувалися у слова. Кожна група людей формувала свої системи звуків та створювала свою мову, знання якої забезпечувало розуміння між представниками різних груп.

Потреба знання не лише своєї, а й іноземних мов залишається актуальним і тепер. Спрощення пошуку інформації та відсутність потреби її перекладу, підвищення шансів отримання бажаної роботи чи посади, можливість перегляду фільмів та читання літератури мовою оригіналу — лише частка переваг розуміння іноземної мови. Із початком війни в Україні актуалізувалася потреба знання іноземної мови, найбільш корисною з яких стала англійська.

Хоч англійська й не є найбільш поширеною, ця мова вважається інтернаціональною. Згідно зі статистичними даними [1], станом на 1 квітня 2022 р. кількість людей, що спілкуються англійською, становить близько 1.5 млрд осіб. Вона визнана офіційною у понад 50 країнах, а відтак розуміння цієї мови значно спрощує налагодження зв'язку в незнайомій країні, із волонтерами при переїзді за межі України та іноземцями тощо.

Багато творів літератури, кіно та музики публікуються англійською, тому знання цієї мови нівелює потребу очікування перекладу та забезпечує краще розуміння мовних особливостей та культури іншого народу.

Також саме англійська вважається «мовою бізнесу». Більшість ділових зустрічей проводяться англійською. Вона ж визначена для використання під час корпоративного спілкування у більшості міжнародних компаній та часто зазначена серед обов'язкових навичок у вакансіях.

Знання англійської знімає обмеження при пошуку інформації. Так, три чверті кореспонденції ведеться англійською мовою, а частка комп'ютерних даних, написаних англійською, становить 80% [2]. Також із початком пандемії COVID-19 [3] у 2019 р. популярність сервісів для віддаленої роботи та навчання значно підвищилася [4,5].

Для сучасної людини виокремлення часу на очне відвідування занять є справжнім викликом, а для самої системи очного навчання — проблемою. Безупинна діджиталізація та модернізація суспільства створює нові механізми, що полегшують життя людей, виконуючи все більше завдань. Створені інтерактивні курси дозволяють виокремлювати час на здобуття необхідних навичок та вирішують зазначену проблему браку часу на навчання загалом.

Сукупність окреслених факторів, а також фактор нерозривного зв'язку життя та діяльності сучасного суспільства з Інтернетом пояснюють зростання популярності програмного забезпечення, що надає доступ до віддаленого навчання та вивчення мов зокрема.

Розроблена інформаційна система (ІС) підтримки вивчення англійської мови дозволяє оптимізувати час для вивчення матеріалу та зусилля, які витрачаються для набуття та вдосконалення відповідних навичок володіння англійською мовою, завдяки наявності наступних можливостей:

- можливість вивчення матеріалу в зручний час;
- використання додатку не залежно від наявності Інтернет-з'єднання;
- можливість вибору слів для вивчення мови, відстеження прогресу;
- можливість мануального додавання нових слів;

– підтримка мотивації завдяки інтерактивності процесу навчання.

Також представлено допоміжні характеристики слів: опис слова, транскрипція, переклад, відео, що допомагає зрозуміти зміст слова. Усі слова, що додаються для подальшого вивчення, систематизовані та зберігаються у словнику. Передбачено можливість перегляду та додавання слів за допомогою системи машинного зору.

1.2 Огляд програмних аналогів підтримки вивчення англійської мови

Перед початком створення нової інформаційної системи є необхідність аналізу існуючих варіантів інтерактивного вивчення іноземних мов з метою виявлення ефективних методик реалізації поставленої задачі та можливого їх удосконалення.

1.2.1 Інформаційна система «Memrise»

Інформаційна система «Memrise» [6, 7] існує у вигляді мобільного додатку та web-сервісу. Система використовує мнемоніми та інтервальне повторення для вивчення нових слів.

На платформі підтримується два види курсів: офіційні, створені командою розробників, та користувацькі. Курси також різняться за темами. Деякі розподілені за рівнями та містять набори слів, знання яких є бажаним для конкретного рівня, інші містять підхід наборів найбільш поширених слів.

На платформі представлено курси вивчення десятків різних іноземних мов та їх діалектів, які постійно модифікуються та поповнюються (рис. 1.1).

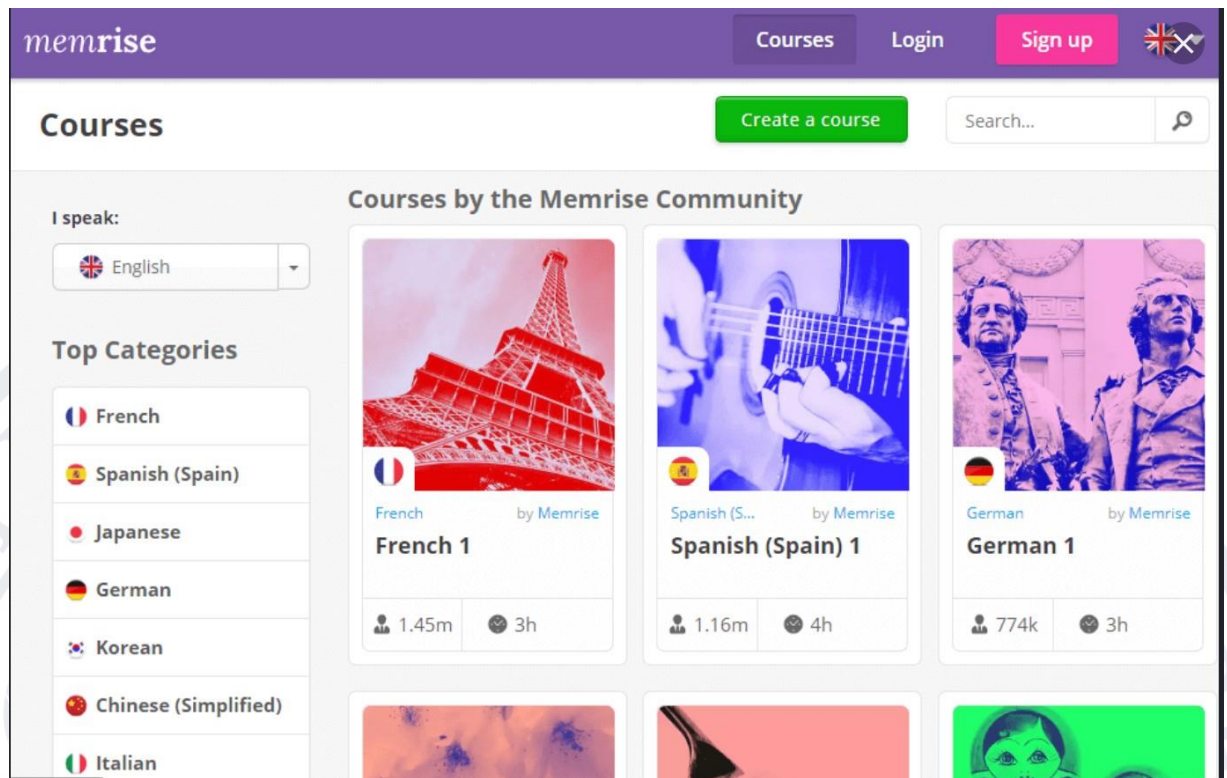


Рисунок 1.1 – Перелік курсів та мов у системі Memrise

Методика вивчення слів Memrise наступна: спочатку представляються слова та їх характеристики (звучання, значення, переклад). Далі проходить вивчення із застосуванням різних вправ: поєднання слів та їх значень, вимовляння слів, написання слів. Також платформа містить навчально-орієнтовані відео, записані носіями мови для ознайомлення з розмовними висловами та акцентами мови (рис. 1.2).

За основу системи розробниками взято три принципи. Перший – яскравий та живий вміст. Програма містить яскравий інтерфейс та цікавий контент, що дозволяє користувачу поринути у вивчення мови. Наявні відеоролики, створені носіями відповідних мов, забезпечують відчуття присутності живого спілкування. Через такі ролики користувачі навчаються розуміти різні акценти та живі голоси, а також можуть ознайомитися з культурою та специфікою комунікації відповідного народу.



Рисунок 1.2 – Представлення роботи додатку в режимі «Відео від носія мови»

Колоритне поєднання науки та технологій – другий принцип. Мається на увазі застосування технологічних інновацій для оновлення інформації та застосування різних комбінацій запам'ятовування слів відповідно до схильностей користувача.

Останнім принципом є забезпечення максимальної зацікавленості користувача у безпосередньому вивченні мови через використання ігрової форми вивчення нових слів.

1.2.2 Інформаційна система «Duolingo»

Нині «Duolingo» [8, 9] є найпопулярнішою системою вивчення іноземних мов [10]. Навчання на платформі формується із застосуванням систем штучного інтелекту та орієнтоване на набування навичок читання, аудіювання та говоріння.

Основний функціонал додатку безкоштовний. Під час навчання користувачі поповнюють словниковий запас і вивчають граматичні правила, щоб досягти поставленої мети.

Також застосовується підхід «неявного вивчення» [11, 12], згідно з яким користувачі знаходять закономірності самостійно, не зациклюючись на

правилах.

Присутня можливість миттєвої перевірки правильності вимови (рис. 1.3).

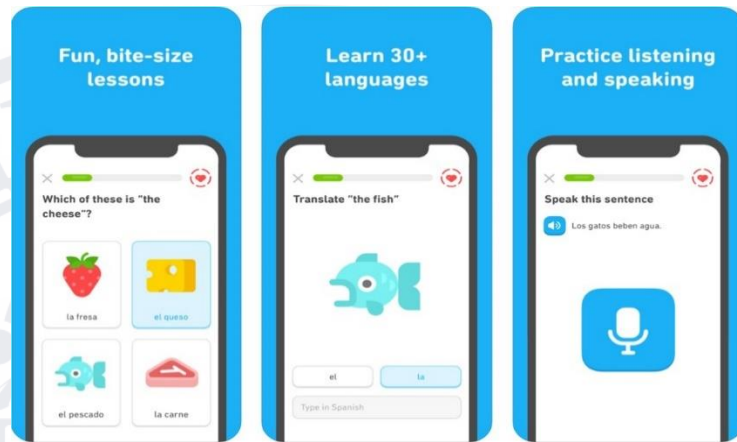


Рисунок 1.3 – Представлення основних функцій додатку «Duolingo»

Мотивація до вивчення обраної мови підтримується через ігрову форму навчання та систему рівнів, які відображають прогрес вивчення матеріалу [13]. Слова та матеріал в цілому розподілений за рівнями знання мови, що вивчається, а також за тематикою цієї інформації (рис 1.4).

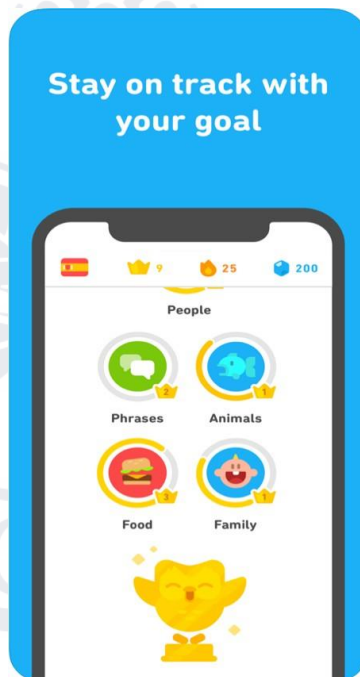


Рисунок 1.4 – Тематика матеріалу, запропонованого до вивчення

Загалом, процес вивчення мови побудовано на перевірній методології, яка допомагає забезпечити запам'ятовування матеріалу на довгий час. Структура деяких речень сформована таким чином, що за рахунок своєї незвичності за змістом забезпечує запам'ятовування базових граматичних конструкцій та робить навчання веселішим для користувача.

Також для забезпечення взаємодії з мовою пропонується прослуховування інтерактивних історій та підкастів – ці формати допомагають користувачам взаємодіяти, розмовляти, читати та розуміти на слух.

1.2.3 Інформаційна система «Rosetta Stone»

«Rosetta Stone», створена у 1992 р., є однією із найстаріших освітніх платформ [14]. Для покращення вивчення іноземної мови на платформі застосовуються основні методики вивчення, такі як читання, вивчення слів, вивчення особливостей спілкування та письмо.

Платформа не використовує рідної мови користувача для пояснення значення слів — усе описується англійською. Перелік мов, доступних до вивчення на платформі, складає 24 мови. Також окремим є вивчення різних акцентів англійської: британська, американська.

Основа методики вивчення англійської з використанням платформи Rosetta Stone — вивчення нового матеріалу, практика та повторення (рис. 1.5). Також надано можливість вивчення мови та проведення занять із носіями мови [15].

Функціонал платформи забезпечує синхронізацію облікового запису на різних девайсах та можливість навчання без підключення до Інтернету за рахунок завантаження уроків.

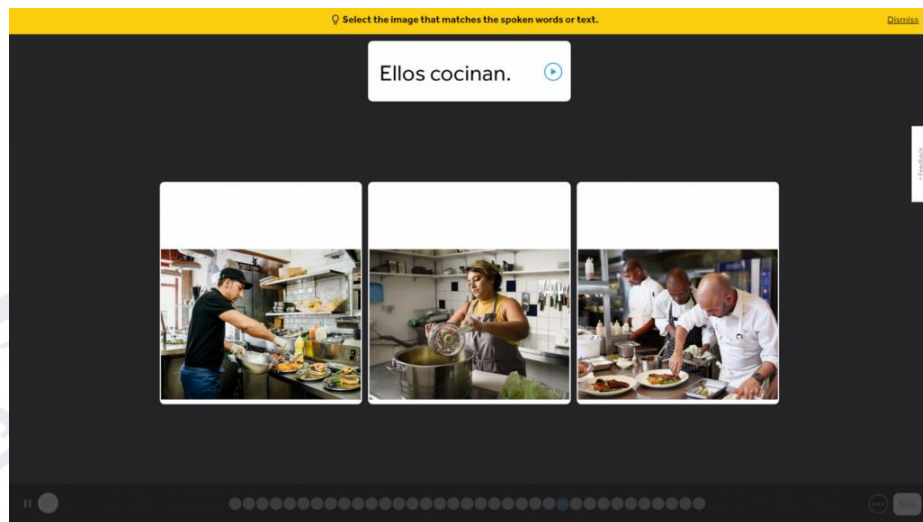


Рисунок 1.5 – Пошук відповідності матеріалу до його змісту

Основними особливостями системи вивчення мов є:

- використання цільової мови для пояснення та опису матеріалу;
- аудіо- та відеосупровід під час ознайомлення та вивчення матеріалу;
- науковий підхід до вивчення правильної вимови;
- розподіл навчального процесу за рівнями та уроками;
- відслідкування прогресу.

Тематичні блоки для кожної мови дозволяють систематизувати слова, які подаються для вивчення. Зокрема, представлені такі тематики: основні правила мови, робота та школа, щоденні справи, надзвичайні ситуації, привітання, відпочинок тощо (рис. 1.6).

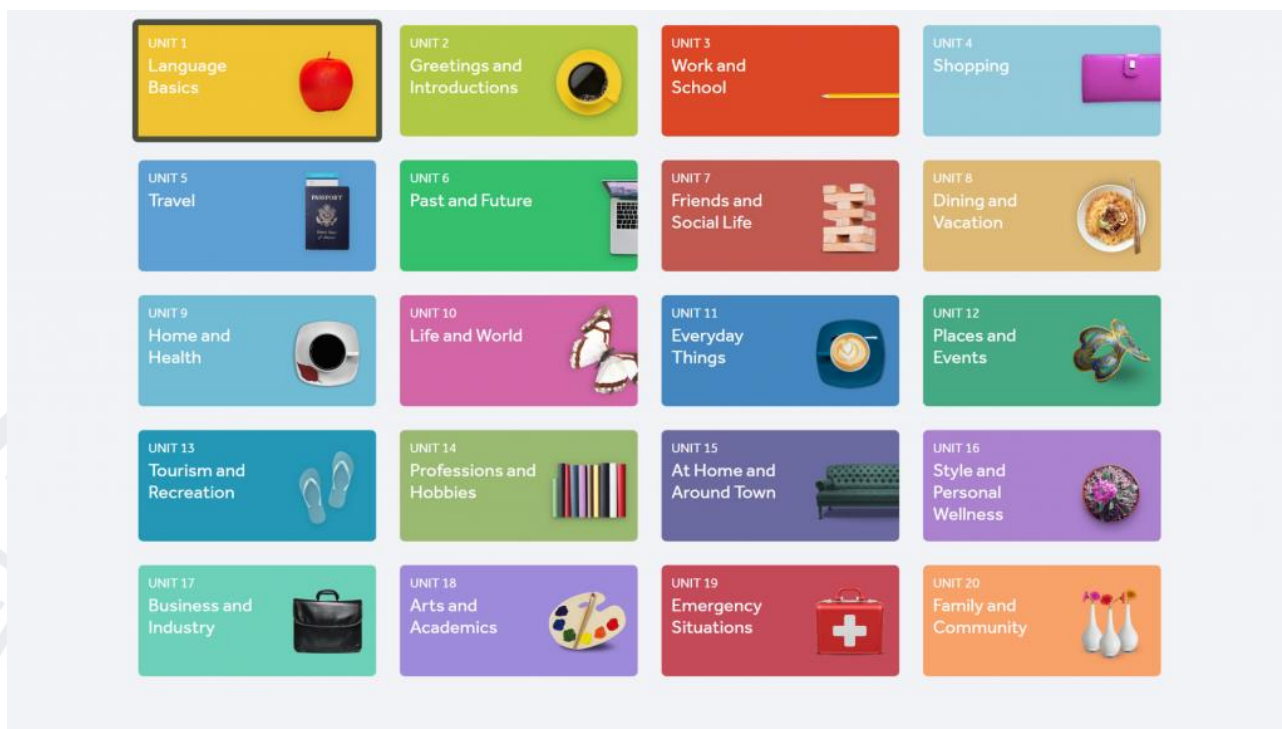


Рисунок 1.6 — Перелік тем для вивчення мови

1.3 Переваги та недоліки існуючих систем вивчення іноземних мов

Виділимо переваги та недоліки розглянутих систем, орієнтованих на вивчення іноземної мови.

Система інтерактивного вивчення мов «Memrise» має широкий вибір іноземних мов для вивчення, а також містить яскравий та живий контент у своїх навчальних курсах. Програми курсів насичені матеріалами, що дозволяють спілкуватися в реальному житті, а також сформовані в ігровій формі за допомогою застосування різних методів та підходів запам'ятовування матеріалу. Особливістю додатку є вивчення іноземних мов лише керуючою англійською, що може становити труднощі для користувачів, які не володіють навичками знання англійської мови на достатньому для цього рівні.

Серед недоліків виділяють [16]:

- невпевненість щодо якості вибраного курсу;
- низький рівень розвитку UX-складової;
- складність синхронізації web-версії та додатку;

– складність пошуку потрібного курсу.

Наступною розглянутою системою є Duolingo – вільна платформа для вивчення мови та перекладу тексту. Сервіс додатку розроблений так, що в міру проходження уроків користувачі допомагають перекладати веб-сайти, статті та інші документи. Пропонуються різні письмові уроки та диктанти, а також словниковий розділ, де можна практикувати раніше вивчені слова.

Статистика досягнень відображається у вигляді ігрового дерева навичок. Мотивація зберігається за рахунок отримання балів за умови успішного проходження тестування та наявності нагадувань про заняття.

Серед недоліків можна виділити [17]:

- недостатність методів набуття розмовних навичок;
- відсутність режиму офлайн;
- слабкий набір лексичних конструкцій, що не може в повній мірі забезпечити розмовними навичками для буденного спілкування;
- відсутність інформації щодо процесу вивчення мови;
- неможливість вибору тем для вивчення.

Останньою розглянутою системою для вивчення мов, зокрема англійської, є Rosetta Stone. Науковий підхід та професійність при розробці матеріалів може забезпечити вивчення нового матеріалу, а комбінація різних методик запам'ятовування – засвоїти його. Функція спілкування з носієм мови здатна розширити словниковий запас, покращити навички сприйняття на слух та позбавитися мовного бар'єру. Для розвитку також додатково присутні розвиваючі ігри, історії та інші інтерактиви, які підтримують зацікавленість до вивчення мови.

Недоліками системи «Rosetta Stone» є [18]:

- висока вартість пакетів навчання;
- набридливість повторення пройденого матеріалу;
- збої в аналізі аудіо при виконанні завдань на вимову;
- невідповідність представлених ситуацій культурі народу, мова якого

вивчається;

– надмірна офіційність матеріалу.

Отже, серед проаналізованих систем за основу обрано систему Memrise, оскільки вона найбільше відповідає сформульованим цілям, при цьому маючи простий інтерфейс. Також цей програмний продукт володіє широкою системою методів запам'ятовування матеріалів, що обираються користувачем для вивчення.

Висновки до розділу 1

Розділ містить інформацію щодо актуальності вивчення іноземних мов. Виявлено, що у сучасних умовах життя суспільства найбільш актуальною для вивчення мовою є англійська: володіння цією мовою нівелює потребу перекладу інформації, спрощує комунікацію з іноземцями, є перевагою при пошуку роботи та кар'єрного росту тощо.

Здійснено огляд подібних ІС, які є нині найбільш популярними серед користувачів. Такими обрано Memrise, Duolingo, Rosetta Stone. Приведено особливості навчання на платформах цих систем, застосовані методики, додаткові функції тощо з метою окреслення переваг та недоліків існуючих платформ для вивчення мов.

Сформовано завдання для реалізації програмного продукту з урахуванням їх недоліків та формуванням необхідних функцій роботи ІС підтримки вивчення іноземних мов.

РОЗДІЛ 2. АНАЛІЗ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ

2.1 Моделювання предметної області, аналіз вимог до ІС підтримки вивчення англійської мови

Для моделювання продукту необхідно виділити головні об'єкти, з якими взаємодіятиме інформаційна система, а також зв'язки між ними [19]. Центральною сутністю ІС є модель під назвою «Слово». Ця модель зберігається у базі даних і має словесний, аудіо- та відеоопис об'єктів або діяльностей, які оточують людей у житті. Слово може бути вже перевіреном і зберігатись на сервері, або представляти власноруч додану користувачем структуру та зберігатись на пристрої. Власноруч додані слова можуть бути перевірені адміністратором і, після валідації, зберігатись у хмарній базі даних. Для перевірки слів користувач має створити відповідний запит та відправити їх перелік на сервер, що також передбачено функціоналом ІС.

За зручний перегляд слів відповідатиме модуль «Словник». Цей модуль являє собою список, у якому всі слова розміщуються в алфавітному порядку за зростанням. Користувач має змогу переглядати детальний опис кожного слова, його транскрипцію, зображення та відео, пов'язані з визначенням об'єктом або явищем, а також прослухати правильну вимову. ІС надає можливість зручного пошуку за допомогою зчитування літер з рядка пошуку.

Розширенням попереднього модуля є менеджер перевірки слів. Він відповідає за попередній перегляд власноруч доданих слів і надає можливість відправити їх на сервер для валідації адміністратором.

Для забезпечення функції додавання слів власноруч розроблено окремий менеджер додавання слів. Такий менеджер являє собою форму з декількома обов'язковими полями, а саме: назва об'єкта (слово), словесний опис, переклад українською мовою, та необов'язковими: транскрипція, посилання на зображення, відеофайл YouTube.

Очевидно, що жодна система вивчення не буде ефективною без періодичного повторення вивченого матеріалу. Для реалізації функції повторення створено менеджери вибору слів для тестування (являє собою список слів, подібний за функціоналом до модуля «словник», проте містить додаткову можливість позначення слова для подальшого його використання у тестуванні) та менеджер перевірки вивчених слів. Менеджер перевірки представлений у вигляді міні гри: надається словесний опис, картинка та відео, що пояснюють слово. Користувачу потрібно розмістити букви у правильному порядку для того, аби сформоване слово відповідало опису.

Ще одним менеджером в ІС є менеджер графічного розпізнавання слів, який дає змогу користувачеві за допомогою машинного зору розпізнавати об'єкти з камери пристрою та шукати їх у словнику. Якщо пошук здійснено успішно, на екран буде виведено детальний опис об'єкта. В іншому випадку користувачу пропонується додати слово до словника.

Головною метою моделювання предметної області є побудова наочної (графічної) моделі логічної структури досліджуваної області, яка ілюструє концептуальні класи, залежності та відносини між ними. Для того, аби візуально зобразити структуру, використано різні види UML діаграм.

UML (Unified Modeling Language) – уніфікована мова моделювання, яка використовується у парадигмі об'єктно-орієнтованого програмування. Ця мова є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення [20]. На рисунку 2.1 можна побачити графічну модель предметної області розроблюваної ІС.

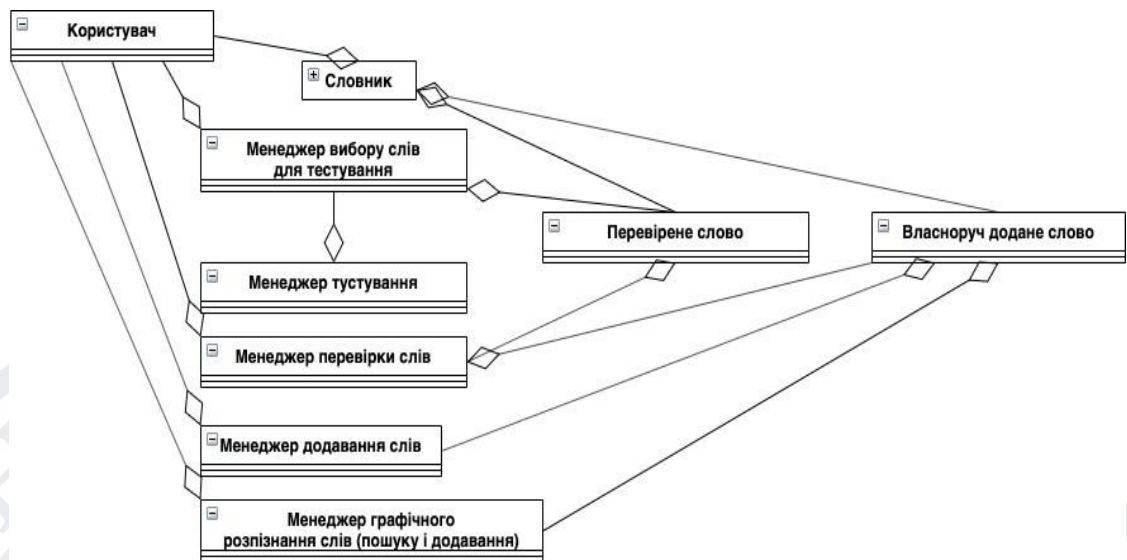


Рисунок 2.1 – Діаграма предметної області ІС підтримки вивчення англійської мови

Представлена діаграма репрезентує, зокрема, взаємодію користувача з кожним модулем системи, які, в свою чергу, взаємодіють між собою. Разом із тим, кожна функціональна одиниця є самостійною і взаємодіє з іншими лише для розширення свого функціоналу або для формування переходів між модулями.

2.1.1 Формування та аналіз вимог

Перед початком розробки будь-якої системи необхідно сформулювати вимоги до неї, обрати архітектурні та поведінкові патерни, які будуть застосовані в процесі реалізації продукту. Для зручності проектування системи усі вимоги розділено на дві категорії: архітектурні та вимоги користувача.

Серед основних архітектурних вимог варто виділити:

- автономність;
- швидкість роботи;
- простота підтримки.

Нижче приведено більш детальний опис зазначених вимог.

Автономність системи забезпечує доступ до основного функціоналу

додатку у разі виникнення збоїв у роботі сервера. Для цього необхідно розробити базу даних, яка зберігатиме необхідні дані на стороні клієнта (на пристрої користувача). Також усі модулі проектуються таким чином, щоб обов'язковим було лише перше звернення до сервера: для отримання усіх даних про слова, які зберігаються у БД на стороні сервера при першому запуску додатка. У подальшому користуванні ІС та виникненні збоїв у роботі сервера користувачеві буде обмежено лише у отриманні нових слів та можливості надсилати власні слова на перевірку. Модулі тестування, додавання нових слів, перегляду слів та визначення об'єкта за допомогою камери ніяк не залежать від мережевих запитів і можуть продовжувати роботу в офлайн режимі.

Для забезпечення швидкості роботи ІС необхідно всі процеси, які не залежать від інтерфейсу користувача, перевести на окремі потоки [21]. Також важливою умовою швидкодії є коректна робота локальної БД. Для реалізації цієї вимоги як локальну БД обрано вбудовану в SDK базу CoreData. Хоч зазначена база складна у використанні та проектуванні, за умови правильного налаштування усіх процесів та перенесення всіх запитів до неї на окремий потік, дана БД забезпечує високу швидкість виконання операцій.

Простота підтримки системи передбачає використання розробником найновіших версій бібліотек. Також важливим є вибір архітектурного підходу, який буде легко масштабуватися, а коректність роботи – перевірятися unit-тестами. Для виконання зазначених умов вважаємо раціональним вибір підходу «Data driven view controllers» у якості архітектурного патерну: цей підхід є поєднанням MVVM та принципу інверсії керування.

2.1.2 Діаграми прецедентів інформаційної системи

Для покращення сприйняття щодо формування вимог користувача розроблено діаграму прецедентів за допомогою UML [22].

Діаграма прецедентів — це граф, що складається з множини акторів, прецедентів (варіантів використання), обмежених границею системи

(прямокутник), асоціацій між акторами та прецедентами, відношень між прецедентами та відношень узагальнення між акторами. Також цей вид діаграм відображає елементи моделі варіантів використання [23].

На таких діаграмах показують варіанти взаємодії акторів із системою. Діаграму прецедентів ІС підтримки вивчення англійської мови представлено на рисунку 2.2.

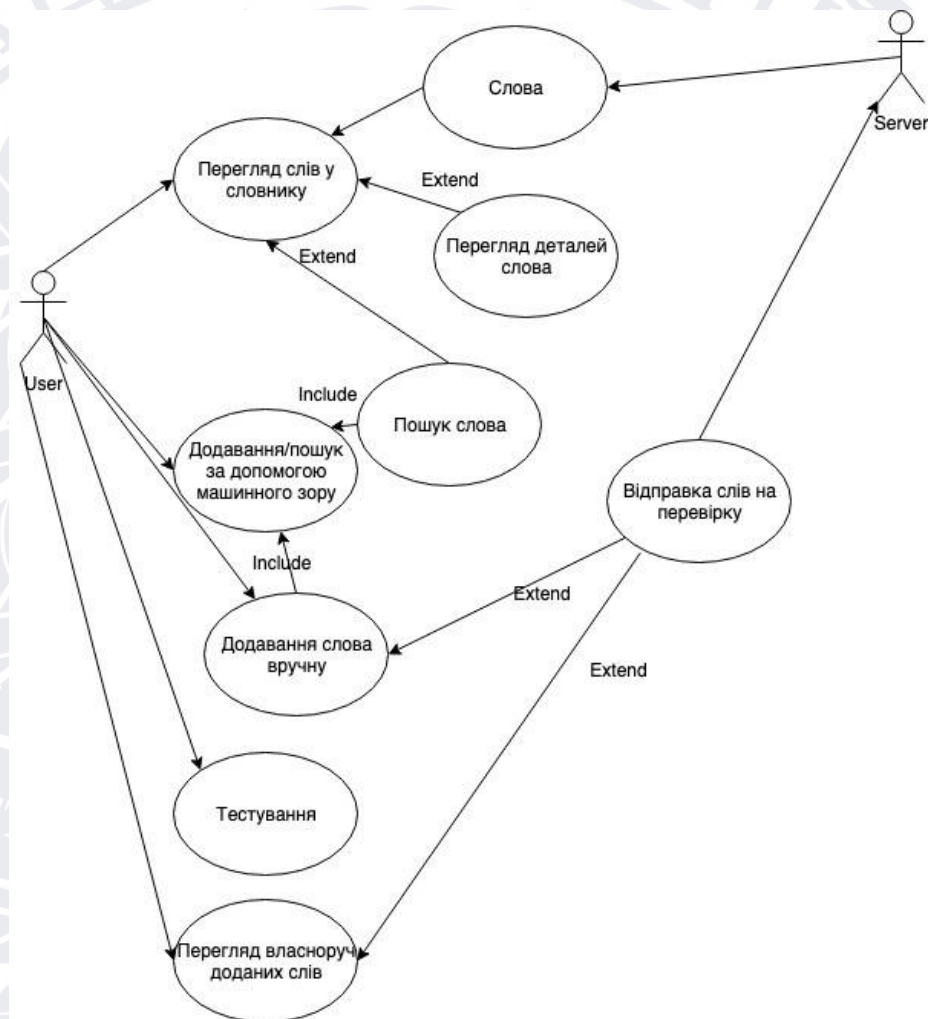


Рисунок 2.2 – Діаграма прецедентів (UseCase) ІС підтримки вивчення англійської мови

Діаграма репрезентує наявність у системі двох акторів: користувач та сервер. Користувач взаємодіє з додатком і всіма його модулями (перегляд слів, додавання слів, використання машинного зору, тестування) за допомогою інтерфейсу; сервер виконує роль бази даних.

Отже, ІС має виконувати такі користувацькі вимоги:

- зручність і простота користування;
- підтримка роботи в офлайн-режимі;
- перегляд слів і їх опис (текстовий, звуковий та візуальний);
- додавання власних слів;
- можливість надсилення доданих слів на перевірку;
- визначення назви об'єкта за допомогою камери пристрою, представлення опису об'єкта за наявності відповідного слова у словнику або здійснення переходу в режим додавання нового слова у разі його відсутності.

Зручність та простота користування забезпечується простим інтуїтивно зрозумілим користувацьким інтерфейсом. Завдяки оптимізації і зміщенню складної вимогливої до ресурсів логіки на сторонні потоки інтерфейс користувача не матиме проблем з кількістю відмальованих кадрів на секунду навіть під час виконання складних задач або за наявності додатково відкритих програм. Офлайн режим забезпечено використанням локальної бази даних. Інші вимоги виконано завдяки реалізації всіх модулів, описаних у пункті 2.1, а також дотриманню загальних вимог, описаних у пункті 2.1.1.

2.2 Логічна структура ІС підтримки вивчення англійської мови

2.2.1 Діаграма класів ІС вивчення англійської мови

Діаграма класів (англ. class diagram) – статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як класи, типи даних, їх зміст та відношення, позначення для пакетів та вкладених пакетів. Діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм. Діаграма класів використовується для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування [24]. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відношення [25].

Нижче здійснено опис створених компонентів діаграми класів для реалізації ІС підтримки вивчення англійської мови.

Для зберігання інформації про слово створено клас `AbstractWord`, який є батьківським для класів `SelfWord` та `Word`. Клас `SelfWord` успадковує всі поля суперкласу та не має власних додаткових полів, клас `Word` розширюється двома власними полями `isApproved` та `isKnown`. Таке архітектурне рішення щодо батьківського та класів-нащадків зумовлене специфікою та вимогами `CoreData` – стандартною базою даних для ОС `iOS`.

Зазначені класи є провайдерами між кодом, написаним мовою `Swift`, та `SQL`-таблицями. Діаграму відповідних класів представлено на рисунку 2.3.

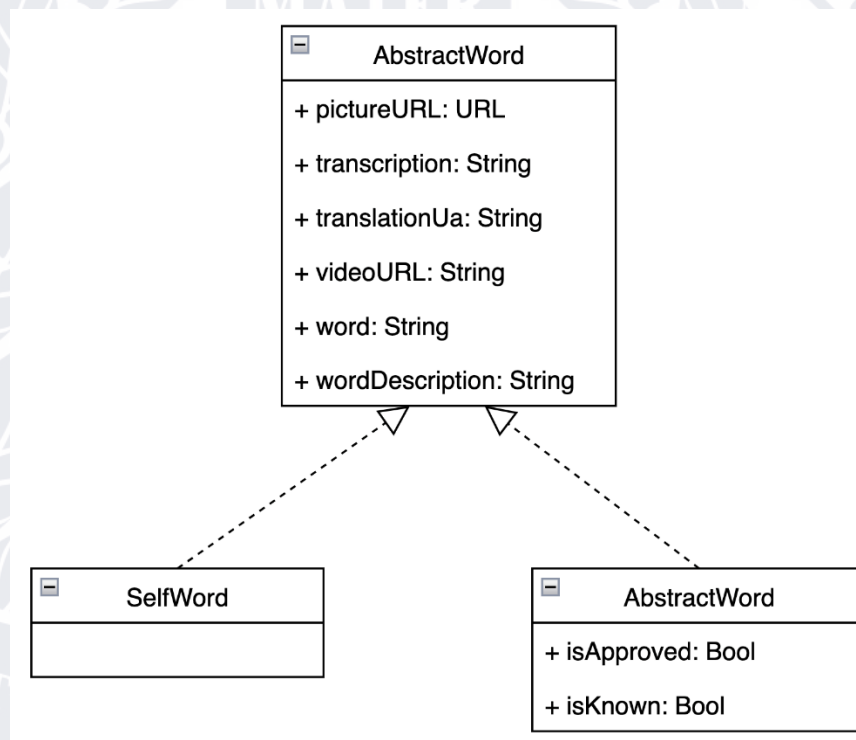


Рисунок 2.3 – Діаграма класу `AbstractWord` та підкласів `SelfWord` і `Word`

Для реалізації роботи з `API` та локальною базою даних створено клас `WordRepository`, який виконує всі запити на сервер та до локальної бази даних, яка зберігається на пристрої. Структуру даного класу наведено на рисунку 2.4.

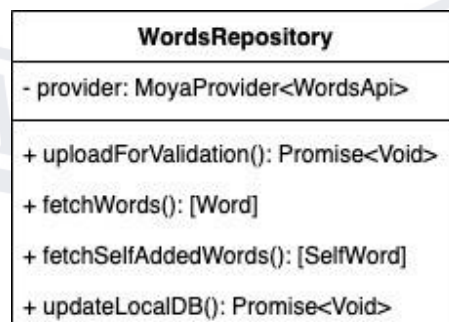


Рисунок 2.4 – Діаграма класу `WordRepository`

Для керування словником сформовано класи DictionaryVC та DictionaryLC: DictionaryVC обробляє усі події, які відбуваються на UI частині, а також відображає моделі, які надає LC. Клас DictionaryLC виконує всю логічну роботу, таку як зв'язок з класом-провайдером WordRepository (див. рис. 2.4), генерація моделі для рендера тощо. Архітектуру описаних класів представлено на рисунку 2.5.

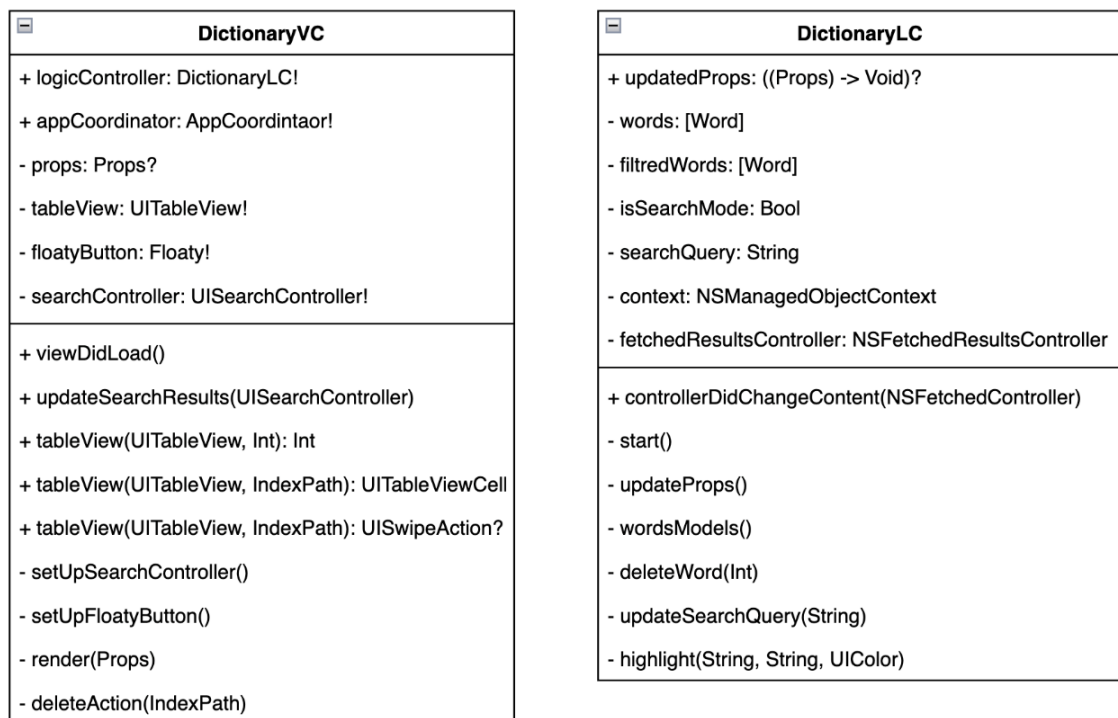


Рисунок 2.5 – Діаграма класу DictionaryVC (UI) та супутнього класу DictionaryLC (Logic controller)

Для перегляду докладної інформації про слово, яке вибрав користувач, реалізовані класи WordDetailsVC та WordDetailsLC. Як і у випадку попередніх класів, VC відповідає за обробку подій на UI та показ моделі, а LC — за логічні дії та створення моделі для показу. Діаграма класів зображена на рисунку 2.6.

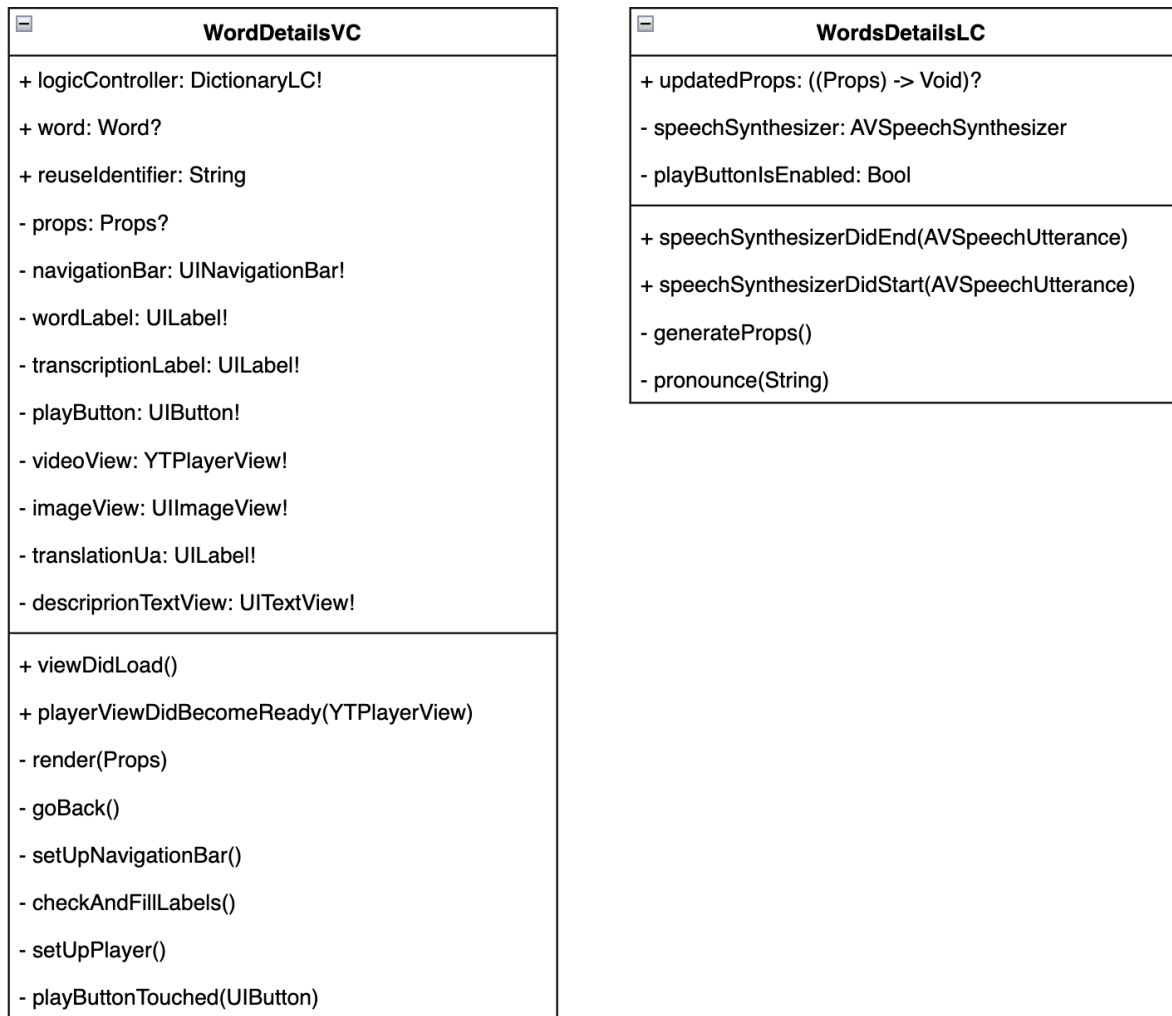


Рисунок 2.6 – Діаграма класу WordDetailsVC (UI) та супутнього класу WordDetailsLC (Logic controller)

Класи AddEditWordVC та AddEditWordLC відповідають за додавання нових та редагування деталей власноруч доданих слів. Залежно від попередніх дій користувача та сценарію, класи можуть відображати як режим додавання нового слова, так і режим редагування раніше доданого. Архітектуру представлено на рисунку 2.7.

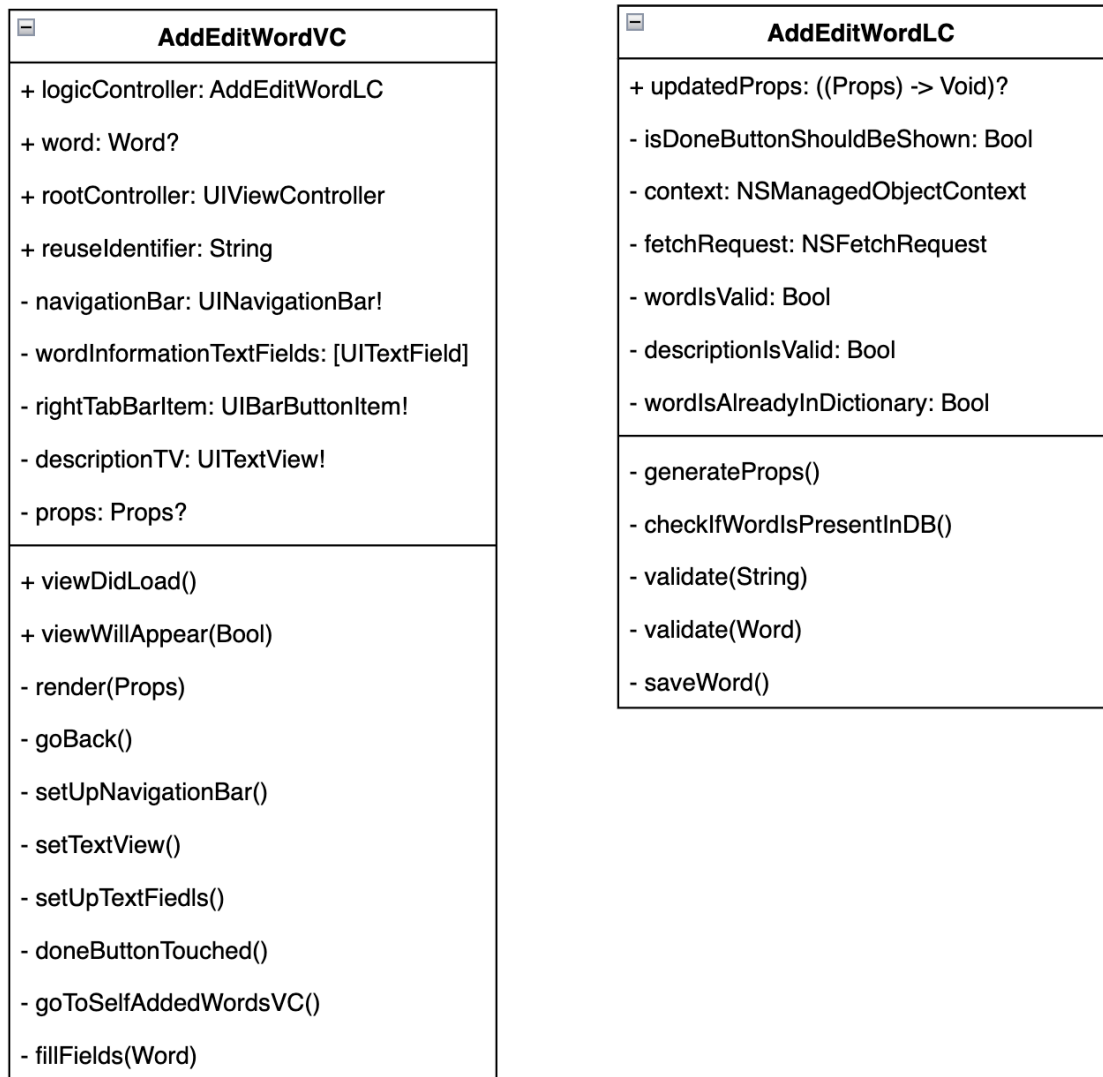


Рисунок 2.7 – Діаграма класу AddEditWordVC (UI) та супутнього класу AddEditWordLC (Logic controller)

Класи RecognizeWordVC та RecognizeWordLC відповідають за розпізнавання об'єкта, який знаходиться в центрі поля зору камери. Ці класи можуть перенаправити користувача до деталей слова за наявності назви цільового об'єкта в базі даних, або до екрану додавання. Архітектура зображена на рисунку 2.8.

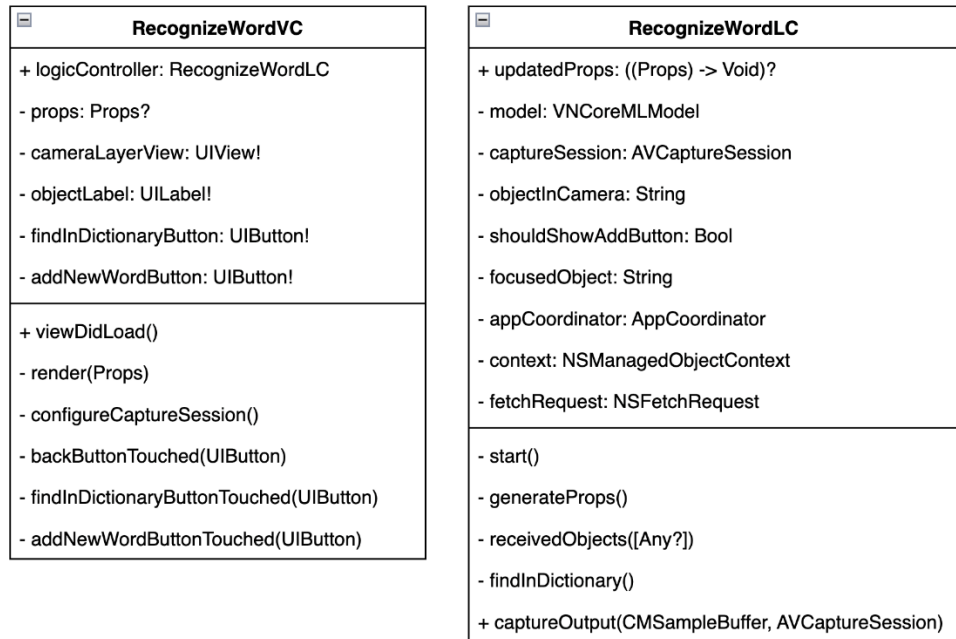


Рисунок 2.8 – Діаграма класу RecognizeWordVC (UI) та супутнього класу RecognizeWordLC (Logic controller)

Класи TestWordsVC та TestWordsLC призначені для забезпечення можливості вибору слів для перевірки їх запам'ятовування. Клас TestWordsVC відповідає за обробку подій на UI, TestWordsLC за логічні функції. Описані класи представлено на рисунку 2.9.

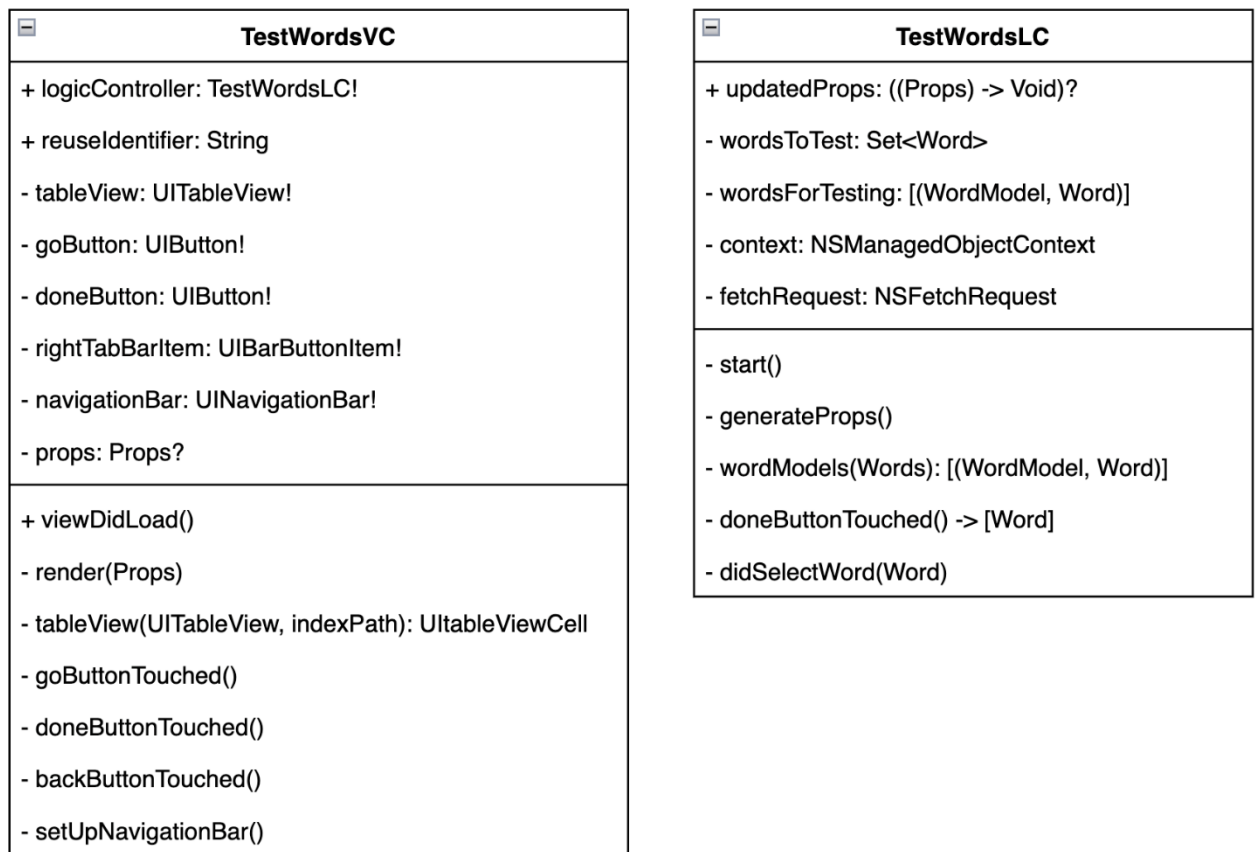


Рисунок 2.9 – Діаграма класу TestWordsVC (UI) та супутнього класу TestWordsLC (Logic controller)

Класи ComposeWordsVC та ComposeWordsLC відповідають за безпосереднє тестування знань користувача. На UI частині, за яку відповідає ComposeWordsVC, є дві секції. Перша містить букви слова, розкидані в довільному порядку. Завдання для користувача: розмістити букви у правильній послідовності у другій секції, щоб слово, яке ці букви утворюють, відповідало опису, який також відображається на UI. За всі логічні операції відповідає клас ComposeWordsLC [26]. Архітектура даних класів представлена на рисунку 2.10.



Рисунок 2.10 – Діаграма класу ComposeWordsVC (UI) та супутнього класу ComposeWordsLC (Logic controller)

Класи SelfAddedWordsVC та SelsAddedWordsLC пов'язані з роботою системи у режимі відображення локального списку слів, доданих користувачем (рис. 2.10).

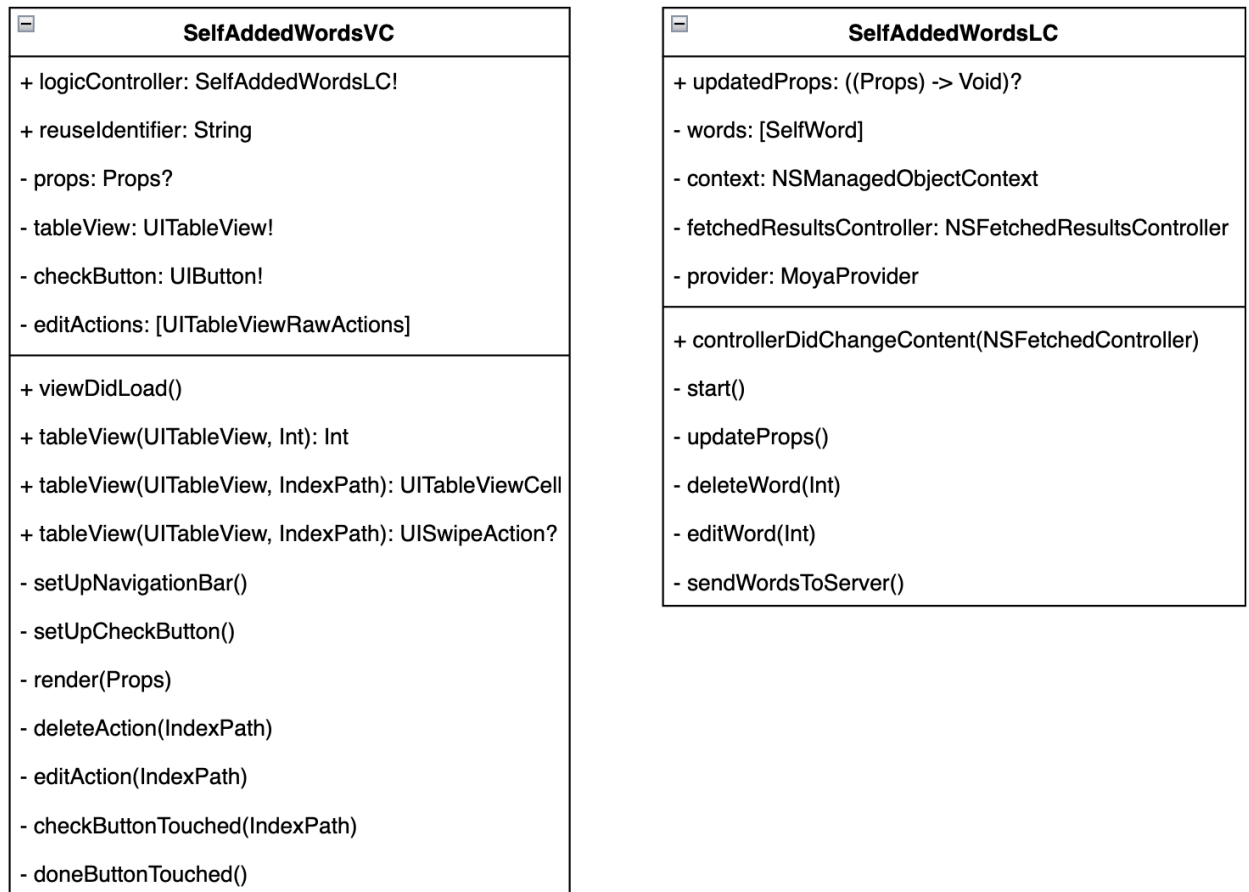


Рисунок 2.11 – Діаграма класів SelfAddedWordsVC та SelsAddedWordsLC

Клас SelfAddedWordsVC відмальовує інтерфейс та відповідає за обробку інтерактивних дій користувача. Клас SelsAddedWordsLC оброблює логічні операції: редагування слова, його видалення, додавання нового, надсилання вибраних слів на перевірку.

Клас AppCordinator реалізує логіку створення екранів та переходів між ними (рис. 2.12).

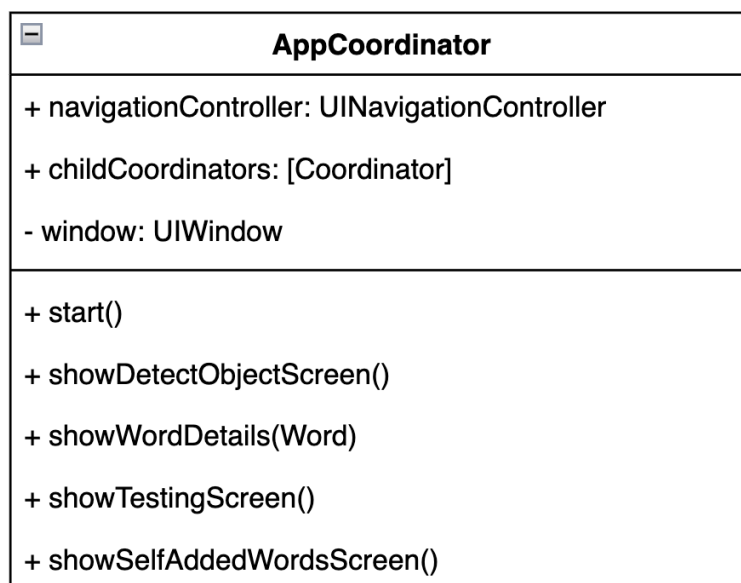


Рисунок 2.12 – Діаграма класу AppCoordinator

2.2.2 Діаграма пакетів ІС вивчення англійської мови

Діаграми пакетів уніфікованої мови моделювання UML відображають залежності між пакетами, з яких складається модель. Пакет (package) – елемент моделі, який використовують для групування інших елементів моделі [27, 28].

Ці діаграми використовують для виділення функціональної залежності системи та представлення способів об'єднання або розширення класів. Розроблювальна ІС складається з пакетів: тестування вивчених слів, роботи зі словником, роботи з камерою, роботи з API та роботи з базою даних. Головним є пакет роботи з БД, оскільки всі інші пакети, хоч і самостійні та самодостатні, функціонально включають в себе методи обробки запитів до бази даних. Нижче здійснено розширений опис кожного з пакетів.

Пакет роботи зі словником надає можливість перегляду, пошуку, додавання, надсилання слів.

Пакет тестування вивчених слів надає можливість вибору зі словника слів для тестування, а також переходу в режим тестування і проведення тестування кожного слова.

Пакет роботи з камерою відповідає за функціонал відображення живого

відео з камери, визначення назви об'єкта, який знаходиться у фокусі камери (центрі екрана), та переадресацію запитів до пакету роботи з БД.

Пакет роботи з API відповідає за зв'язок із сервером, а також формує всі запити до нього, серіалізацію даних.

Пакет роботи з базою даних відповідає за всі запити, які надходять до БД, серіалізацію даних у та з формату, необхідного для локального зберігання даних.

Діаграма пакетів зображена на рисунку 2.13.

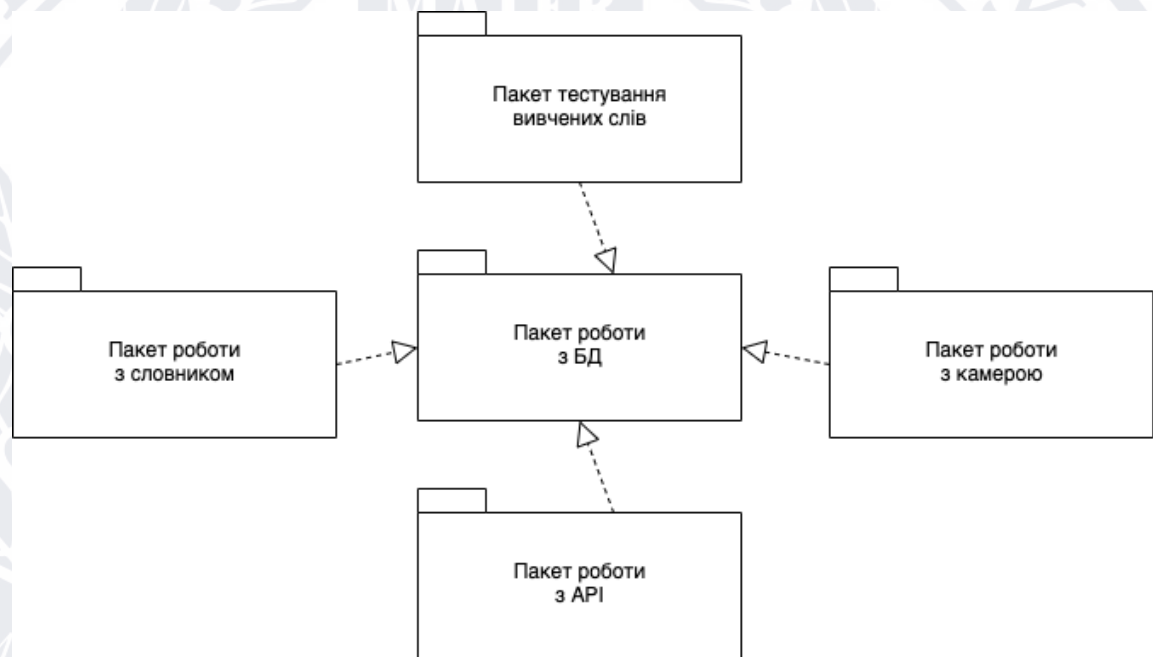


Рисунок 2.13 – Діаграма пакетів ІС підтримки вивчення англійської мови

2.3 Архітектурне проектування

Створення архітектури є етапом проектування системи на найвищому рівні. Логічна архітектура описує систему в термінах її принципової організації у вигляді пакетів, програмних класів і підсистем. Вона називається логічною, оскільки не визначає способи розгортання цих елементів у різних операційних системах або на фізичних комп'ютерах у мережі (це відноситься до архітектури розгортання). Етап архітектурного проектування ІС повинен бути відображений шляхом опису процесу об'єктно-орієнтованої декомпозиції системи до рівня

переліку підсистем та їх зв'язків, опису її логічної структури у вигляді пакетів компонентів [29,30].

2.3.1 Діаграма модулів ІС вивчення англійської мови

Діаграма компонентів відповідає за відображення внутрішньої структури і залежностей між наборами компонентів. Компоненти, у свою чергу, можуть бути представлені файлами, бібліотеками, фізичними модулями програмного коду, інтерфейсами тощо. Кожен компонент є фізичною реалізацією пакетів логічного проектування (див. рис. 2.13). На рисунку 2.14 представлено діаграму модулів розроблюваної ІС підтримки вивчення англійської.

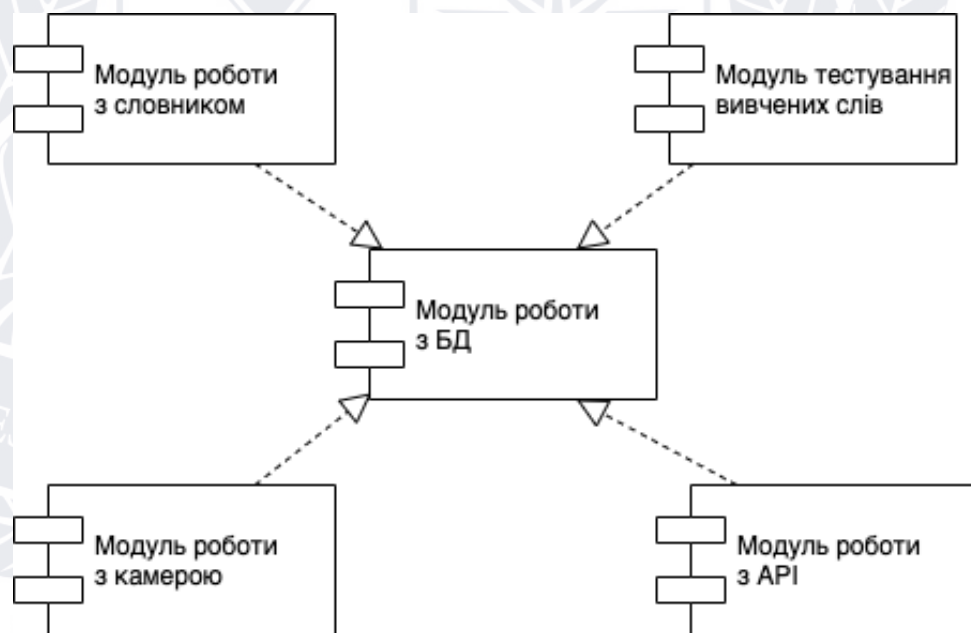


Рисунок 2.14 – Діаграма модулів ІС вивчення англійської мови

Модуль роботи зі словником відповідає за всі маніпуляції зі словами, які зберігаються на пристрої. Він містить алгоритми користувацького інтерфейсу для коректного відображення списку наявних слів, власноруч доданих, а також показу деталей слова. Він звертається до модуля роботи з БД для виконання усіх запитів, що стосуються даних.

Модуль тестування вивчених слів відповідає за вибір слів для тестування

та, власне, тестування. Включає в себе набір логіки для користувацького інтерфейсу (для коректного відображення анімацій і переходів), логіку вибору, а також логіку тестування (мінігри). Як і попередній модуль, модуль тестування взаємодіє з базою даних через модуль роботи з БД.

Модуль роботи з камерою відповідає за визначення об'єктів у фокусі камери (програмно зафіксований у центрі екрану). Включає в себе логіку обробки зображень, які надає камера, обробки запитів до нейронної мережі (представленої у вигляді CoreML-моделі), обробки результатів аналізу нейромережі, а також перенаправлення до функціоналу додавання слова або перегляду деталей. Взаємодіє з БД так само, як і попередні модулі.

Модуль роботи з API відповідає за зв'язок клієнтської частини ІС із сервером-хмарою. Містить логіку сереалізації структур даних з та в JSON-об'єкти, надсилання даних об'єктів на сервер, або ж структур для модуля роботи з БД.

Модуль роботи з базою даних відповідає за весь зв'язок ІС з файлами БД. Містить логіку додавання, редагування, видалення, пошуку об'єктів в БД. Також відповідає за формування оптимізованих запитів до БД. Наприклад, замість фільтрування та сортування отриманих об'єктів в інших модулях ефективніше сформулювати правильний запит до БД, який поверне масив вже відсортованих об'єктів.

2.3.2 Розгортання ІС підтримки вивчення англійської мови на апаратних засобах

Для представлення фізичної структури ІС та особливостей роботи її компоненти використано діаграму розгортання.

Діаграма розгортання (англ. deployment diagram) – UML-діаграма, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах [31].

Дана ІС розгортається на двох апаратних засобах: сервер, який відповідає

за зберігання бази знань для всіх користувачів, та персональний пристрій кожного з користувачів. На рисунку 2.15 зображено взаємодію користувача з пристроєм (телефоном або планшетом Apple), а також взаємодію пристрою з хмарним сховищем.

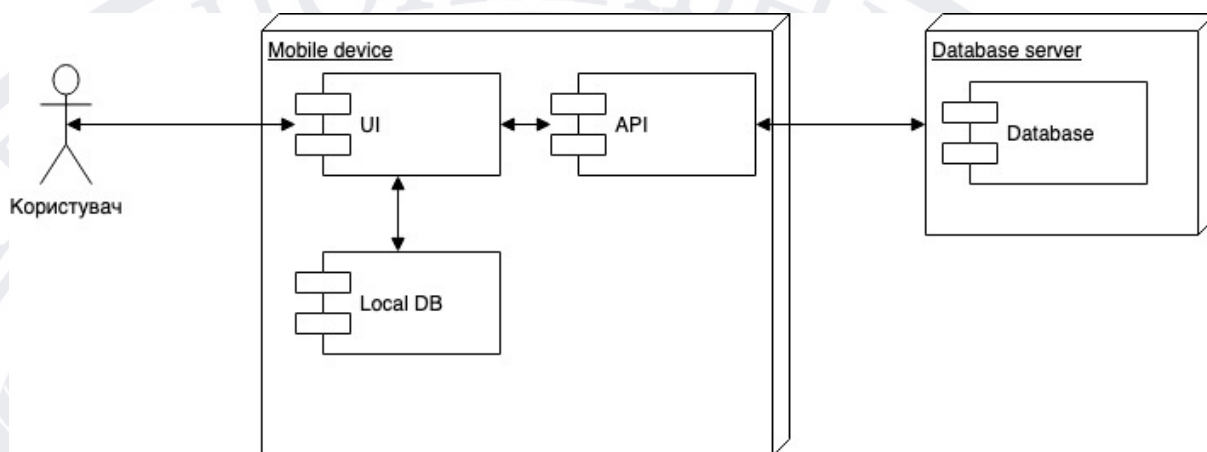


Рисунок 2.15 — Розгортання ІС підтримки вивчення англійської мови на апаратних засобах

Основне навантаження відводиться потужностям мобільного пристрою, оскільки він відповідатиме за всі маніпуляції з даними, відмальовуватиме інтерфейс та надаватиме всі функції, які сприяють вивченню мови. З цієї причини розгортанню ІС на пристрої користувача необхідно приділити найбільше уваги.

Правильне моделювання серверної частини забезпечить швидкодію та коректність обробки запитів. Важливим є також питання щодо усунення проблем, спричинених перевантаженням. Оскільки точна кількість користувачів, які використовуватимуть додаток одночасно, невідома, важливим є забезпечення стабільності роботи ІС при великому навантаженні. Для забезпечення безпеки під час користування додатком нині особливо актуальним є також розробка механізмів протидії DDOS-атакам.

2.4 Моделювання поведінки ІС підтримки вивчення англійської мови

Зазвичай поведінка системи описується множиною об'єктів, які обмінюються даними (повідомленнями), відображених на діаграмах діяльності або послідовності [32]. Цей етап розробки системи визначає характер зв'язків об'єктів між собою, умови утворення та підтримки цих зв'язків, а також їх послідовності.

2.4.1 Діаграма діяльності ІС

Діаграма діяльності (англ. activity diagram) — в UML — візуальне представлення графа діяльностей. Граф діяльностей є різновидом графа станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій [33].

Оскільки побудова діаграми діяльності є складною з точки зору громіздкості та складності розуміння кінцевого результату, для представлення зв'язків між компонентами реалізовано декомпозицію та побудовано діаграму діяльності для кожного модуля окремо.

Робота додатку починається з відкривання самого додатку та переходу до словника. Далі є два варіанти роботи з додатком: перегляд списку доданих слів у поточному вікні, вибір слова для перегляду детальної інформації про нього, закриття екрану та завершення роботи з додатком. Графічно ці події представлено на рисунку 2.16.



Рисунок 2.16 – Діаграма діяльності, що показує дії користувача під час перегляду слів у словнику

Також словник забезпечує можливість пошуку слів за допомогою рядка пошуку. Зазначений процес зображено на рисунку 2.17.

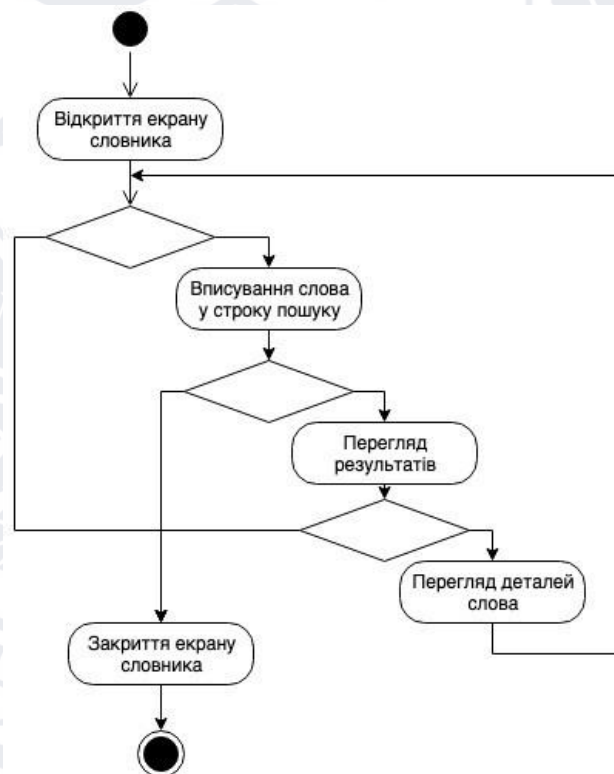


Рисунок 2.17 – Діаграма діяльності, що показує дії користувача під час пошуку слова у словнику

Представлена на рисунку 2.17 діаграма відображає можливість користувача вписати ключові букви, які будуть використані для пошуку, переглянути результати фільтрації, переглянути деталі слова або ж вийти з режиму пошуку та словника.

Однією з вимог до розроблюваної ІС визначено додавання власних слів. Рисунок 2.18 містить діаграму діяльності додавання нового слова. Користувач має змогу в будь-який момент залишити цей режим та перейти до перегляду власноруч доданих слів. Зберегти слово можна лише за умови внесення обов'язкових даних у форму.

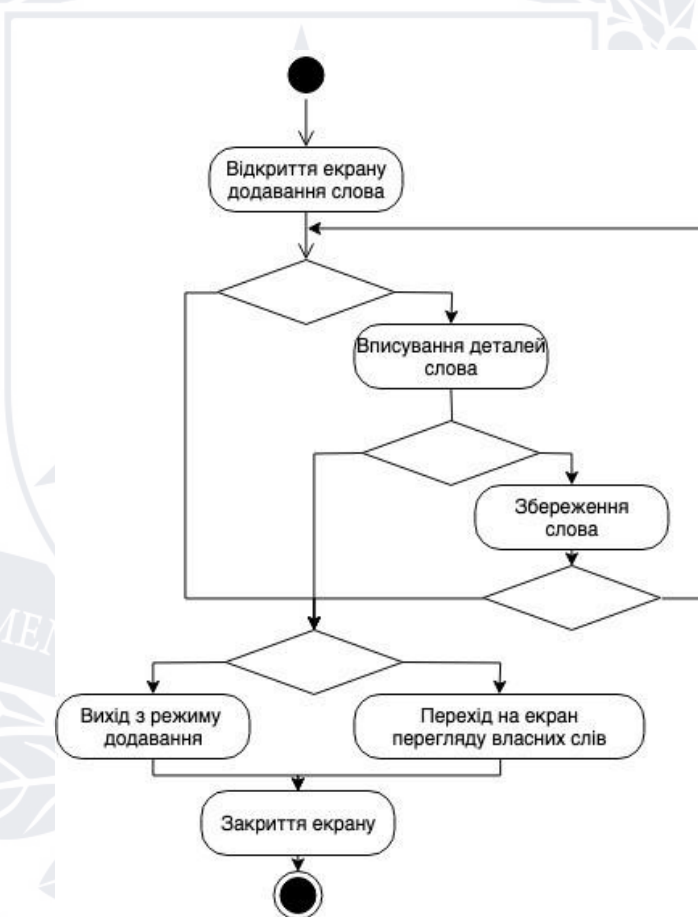


Рисунок 2.18 – Діаграма діяльності, що описує дії користувача під час додавання слова вручну

Також користувач може використовувати камеру пристрою для визначення назви того чи іншого об'єкта, навівши фокус на нього. Завершення роботи додатку у цьому режимі можливе у будь-який зручний користувачу

момент часу. Після розпізнання об'єкта користувач може використати функцію перевірки наявності слова-назви об'єкта у базі даних: якщо такий запис присутній – система відкриє екран з деталями слова. У разі відсутності назви об'єкта система запропонує додати слово до словника. Відповідна діаграма діяльності представлена на рисунку 2.19.

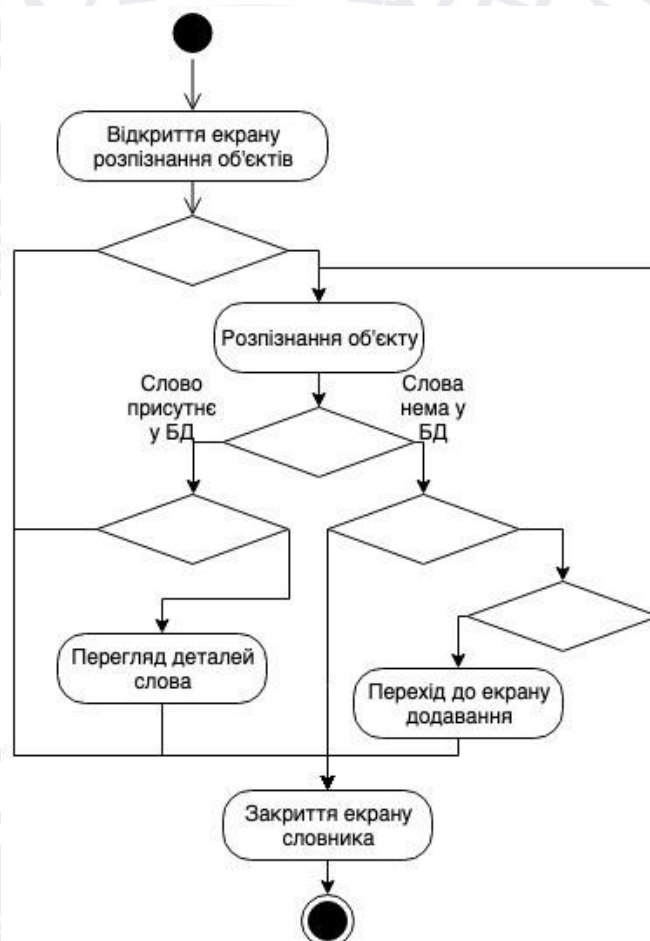


Рисунок 2.19 – Діаграма діяльності дій користувача під час додавання слова за допомогою автоматичного розпізнання об'єкта

Перегляд власноруч доданих слів відбувається за наступним сценарієм. Після відкриття екрану зі списком, користувач може переглянути деталі будь-якого слова зі списку, відправити всі слова на перевірку, відредагувати слово або ж перейти на екран для додавання нових слів. Будь-яку операцію можна закінчити у будь-який момент часу та перейти до попередньої сцени. Графічне зображення даної діяльності проілюстровано на рисунку 2.20.

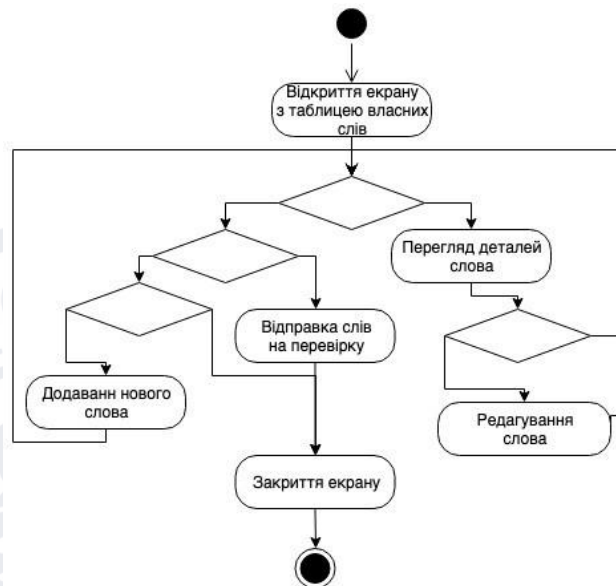


Рисунок 2.20 – Діаграма діяльності дій користувача під час перегляду власноруч доданих слів та відправки їх на перевірку

Діаграму діяльності, яка відповідає режиму тестування слів у системі, зображено на рисунку 2.21.

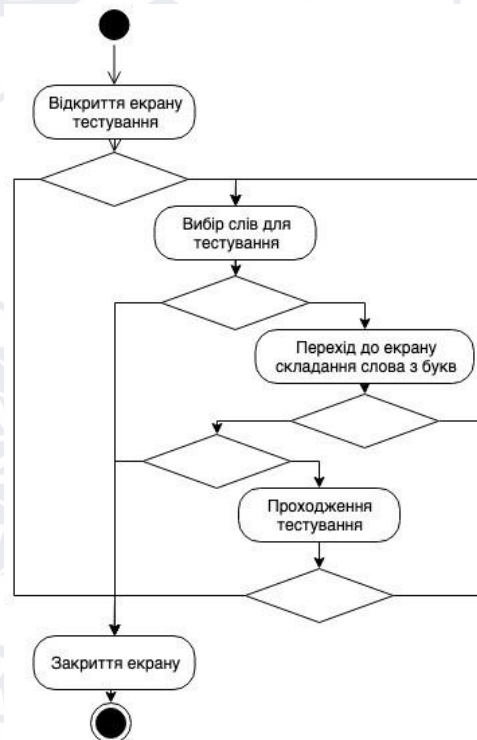


Рисунок 2.21 – Діаграма діяльності дій користувача під час тестування вивчених слів

Представлена діаграма показує, що після відкриття екрана тестування користувач може виконати наступні дії заданому порядку:

- вибрати слова для мінігри;
- перейти на екран мінігри (виставлення букв на правильні позиції);
- перехід до перевірки наступного слова після перевірки поточного.

Варто зазначити, що кожна з дій може бути перервана виходом з поточного режиму.

2.4.2 Діаграма послідовності

Діаграма послідовності відображає взаємодію об'єктів, впорядкованих за часом, а також послідовність відправлених повідомлень. На діаграмі послідовностей у вигляді вертикальних ліній показано різні процеси або об'єкти, що існують водночас. Надіслані повідомлення зображуються у вигляді горизонтальних ліній в порядку відправлення [34, 35].

Зазвичай на таких діаграмах будують алгоритмічну роботу інформаційної системи, а також визначають послідовності та способи взаємодії об'єктів для обміну даними.

Зважаючи на складність відображення усіх процесів, які мають місце під час роботи ІС на одній діаграмі, а також на зрозумілість та інформативність кінцевого результату, проведено декомпозицію роботи ІС та пообудовано кілька діаграм послідовностей, кожна з яких репрезентує один із варіантів використання додатку.

На рисунку 2.22 зображено сценарій взаємодії користувача з модулем словника.

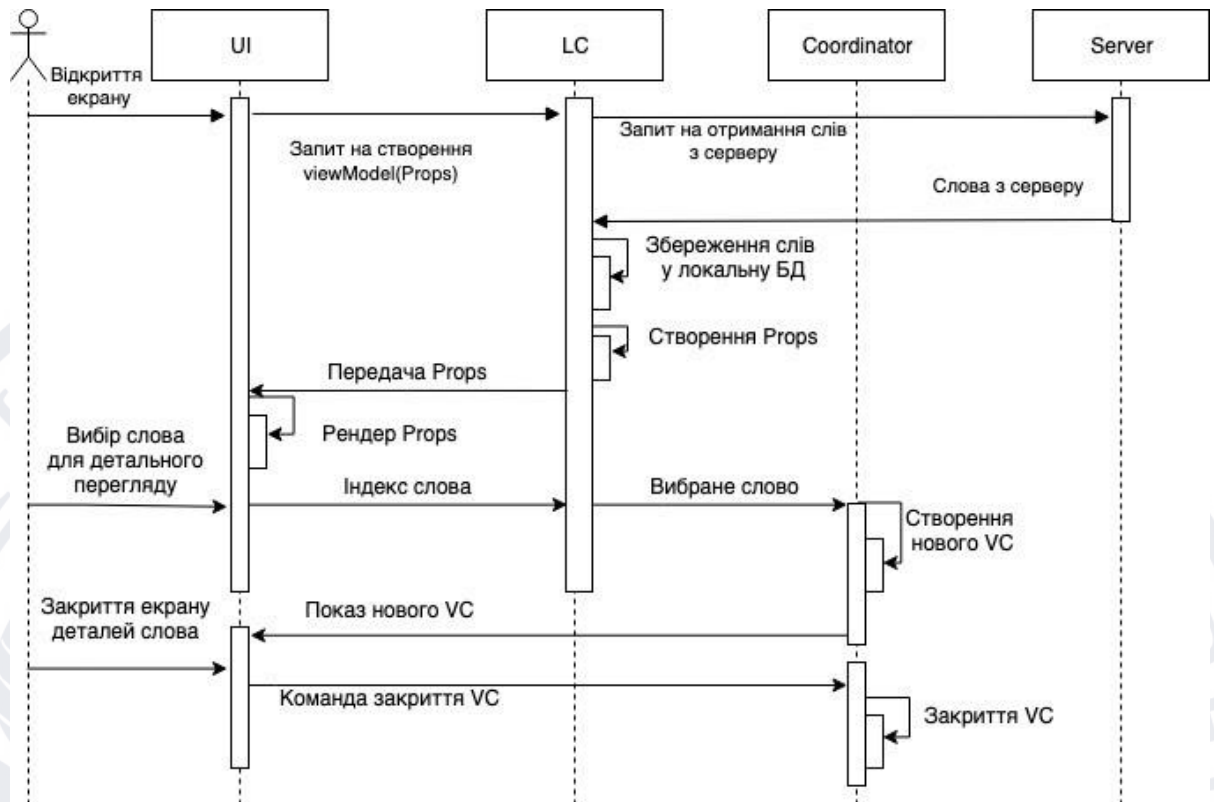


Рисунок 2.22 – Діаграма послідовності перегляду користувачем слова у словнику

Діаграма містить наступні дійові об'єкти:

1. Користувач – взаємодіє з інтерфейсом користувача та є ініціатором будь-яких дій.
2. UI – клас, який відповідає за відмалювання view model'i, а також реагує на всі дії користувача, передає їх далі до класу LC.
3. LC – об'єкт, який займається обробкою всіх логічних функцій та взаємодіє з іншими модулями системи.
4. Coordinator – об'єкт, який відповідає за навігацію всередині IC. Саме він вирішує, який саме екран слід показати наступним.
5. Server – відповідає за збереження слів у хмарі.

Основний алгоритм можна поділити на два етапи: підготовчий та роботи зі словником.

На етапі підготовки, після того як користувач ініціює оновлення словника,

об'єкт UI звертається до LC для створення view model. LC, у свою чергу, звертається до Server для отримання актуальної інформації та подальшого її збереження у локальній базі даних. Після збереження, на основі нових даних формується view model, яка передається UI для відмалювання інтерфейсу. На цьому підготовчий етап завершено.

Після вибору користувачем слова для детального перегляду UI формує відповідний запит з індексом у списку і звертається до LC, який визначає об'єкт і передає його Coordinator'у, який створює новий екран і відправляє до UI. На цьому життєвий цикл LC завершено.

Тепер користувач може ініціалізувати подію закриття екрана деталей, яка від інтерфейсу передається координатору, після чого відбувається закриття екрану.

Розглянемо сценарій пошуку слова у словнику, представленого на рисунку 2.23.

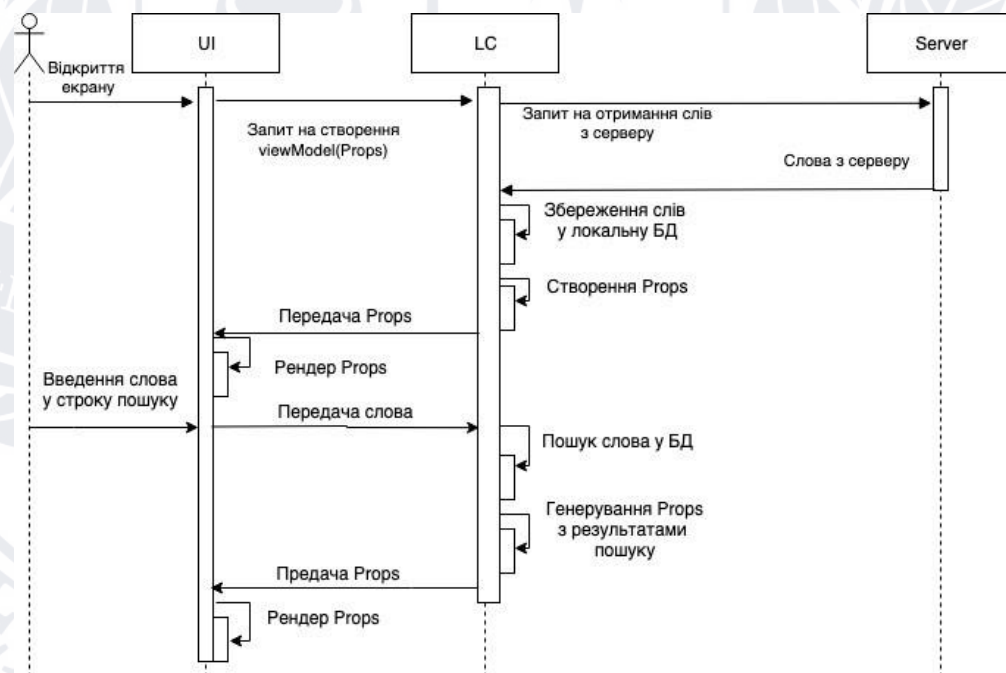


Рисунок 2.23 – Діаграма послідовності пошуку користувачем слова у словнику за допомогою частини слова або цілого слова

Під час роботи системи у режимі пошуку слова користувач має

можливість вписати ключову послідовність символів у рядок пошуку, який потім буде передано об'єкту класу LC. Клас LC ініціює пошук слова у базі даних і відповідно до результату пошуку створить view model, яка передається до UI для подальшого її відмалювання.

Розглянемо сценарій додавання нового слова (рис. 2.24). Діючі об'єкти UI, LC і User описані вище і виконують ті ж функції. Якщо екран додавання нового слова використовується як редактор уже наявного в БД слова, UI відправляє запит на створення view model'і об'єкта LC. LC, базуючись на даних про слово, генерує view model і відправляє її UI, який займається рендером. Користувач може змінювати інформацію у формі, кожна зміна даних ініціює запит на генерування нової view model'і, яку генерує LC і рендерить UI. Якщо ж екран відкривається у режимі додавання, генерується модель з пустими даними.

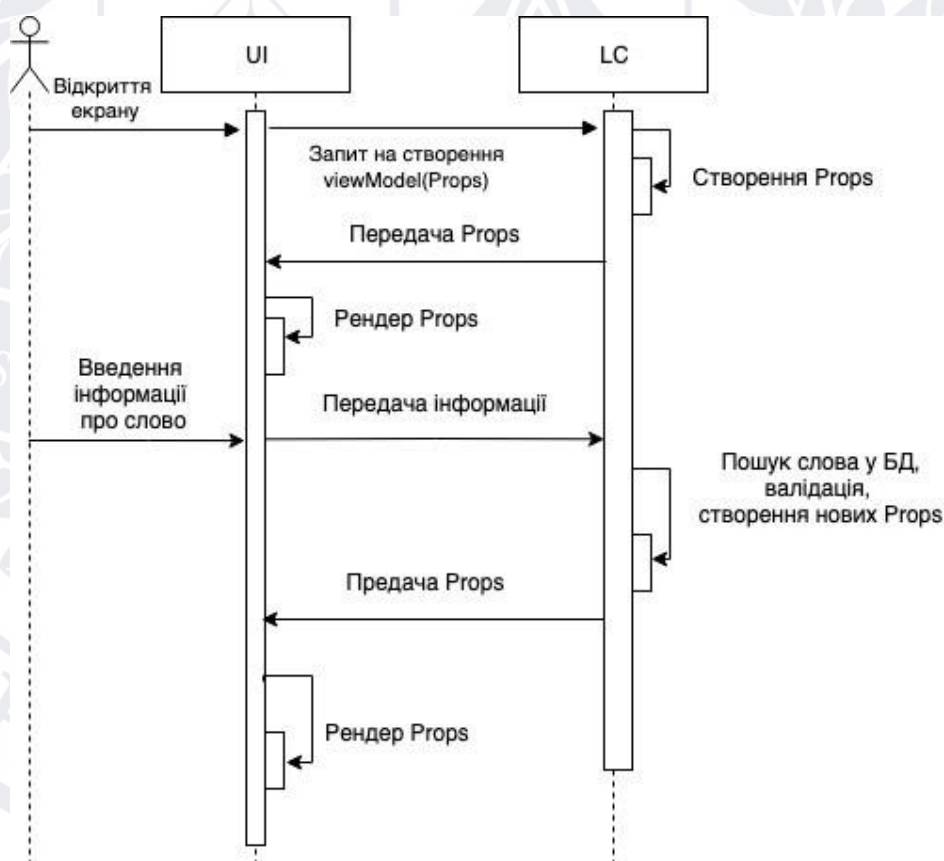


Рисунок 2.24 – Діаграма послідовності додавання користувачем нового слова у базу даних вручну

Користувач може внести зміни у дані, що, в свою чергу, викликає зміну view model'i. Зберегти внесені зміни можна за умови заповнення обов'язкових полів і проходження валідації цих полів.

Наступний сценарій – розпізнавання об'єкта за допомогою камери, а також пошук назви даного об'єкта в словнику та додавання власноруч, якщо в базі даних немає відповідного запису. Діючими класами є User, UI, LC, Coordinator, описані у попередніх сценаріях. Новим діючим об'єктом є «Camera + TF», який відповідає за отримання кадрів з камери та визначення об'єкта на них. Діаграма послідовності розпізнавання об'єктів представлена на рисунку 2.25.

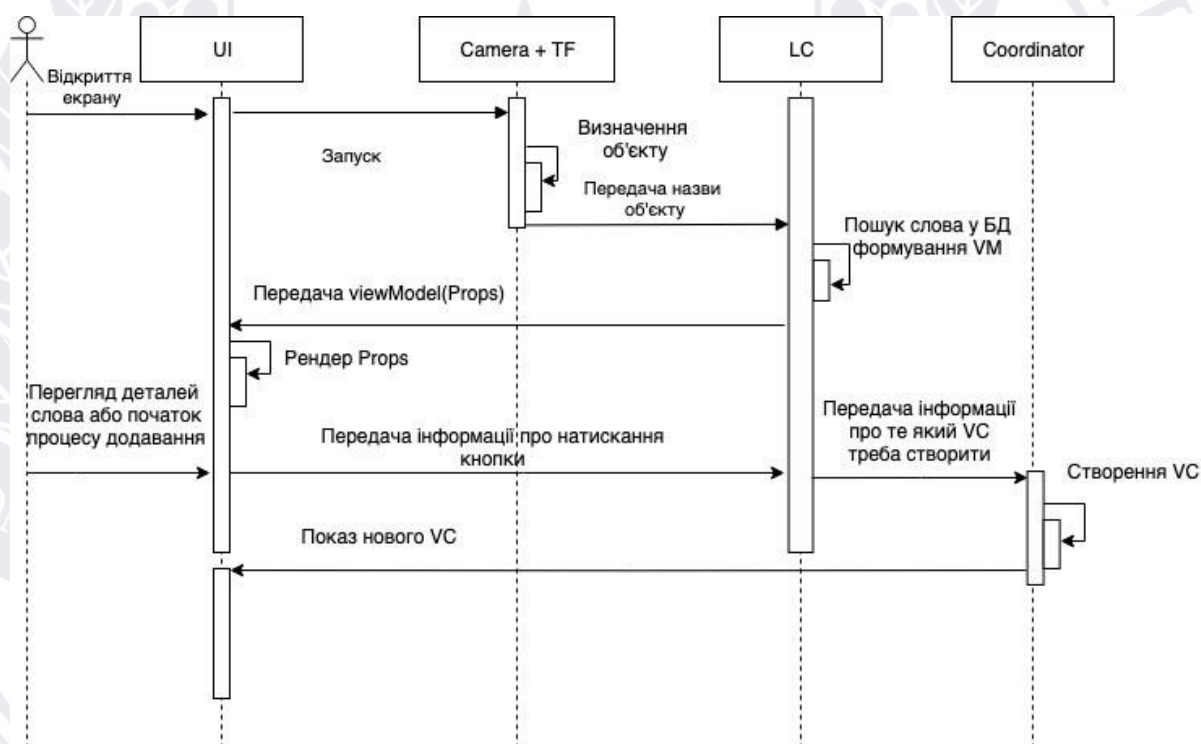


Рисунок 2.25 – Діаграма послідовності додавання/пошуку користувачем слова у базу даних за допомогою функції автоматичного розпізнавання об'єктів

Після відкриття екрана UI посилає запит до Camera+TF на початок сесії отримання кожного фрейма, який надає камера. Після цього відбувається визначення назви об'єкта, який знаходиться у точці фокусу. Назва об'єкта передається до LC, який визначає наявність запису в БД, що відповідає даному об'єкту, та формує view model, яка далі передається до UI. UI відмальовує

модель, після чого користувач може натиснути одну з двох кнопок (залежно від результату пошуку у БД): відкриття деталей слова або додавання до словника нового запису. Інформація про натискання передається LC, далі — до Coordinator, який створює новий екран і передає його до UI. UI показує новий екран.

Сценарій відправки неперевірених слів на сервер містить такі дійові об'єкти: User, UI, LC, Server (рис. 2.26). Після відкриття екрана UI передає LC запит на генерування view model'i. LC збирає з бази даних усі об'єкти невалідованих слів та створює на їх основі модель для показу, передає її до UI. UI відмальовує модель далі користувач тисне на кнопку відправки, інформація про натискання передається LC, який формує запит до Server і надсилає йому необхідну інформацію. Server, у свою чергу, залежно від успішності виконання запиту, формує відповідь і надсилає її назад додатку. LC отримує відповідь від сервера, знову генерує view model і передає її UI. UI створює інтерфейс на основі моделі.

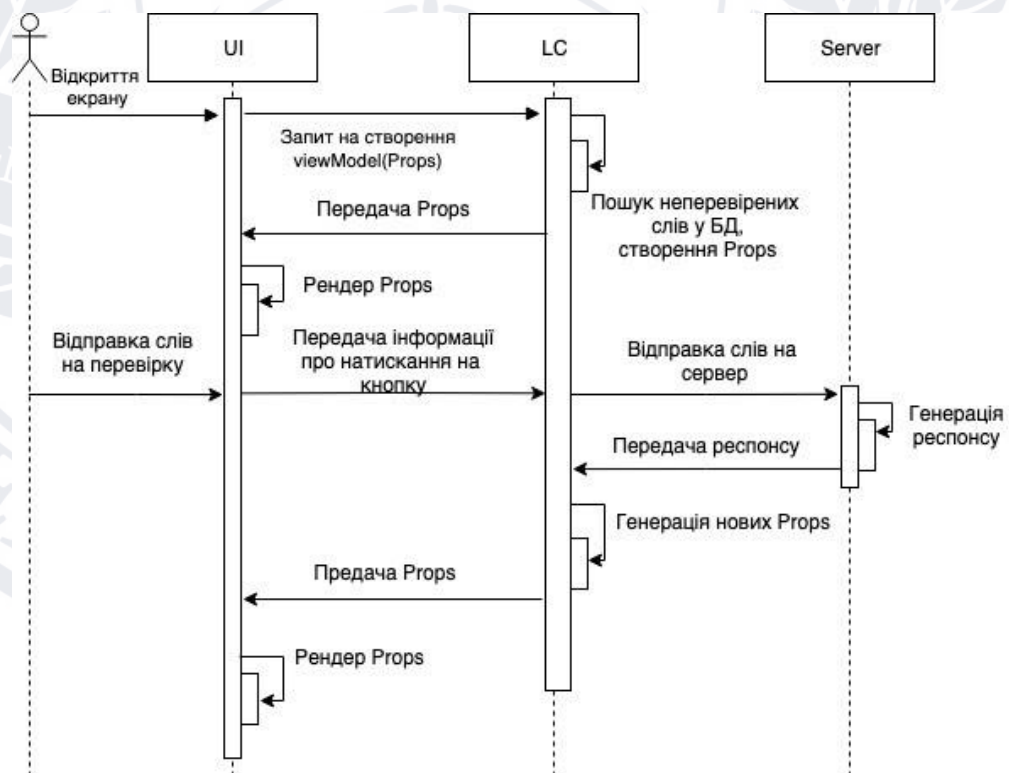


Рисунок 2.26 – Діаграма послідовності перегляду та відправки на перевірку власноруч доданих слів

Останній сценарій роботи системи – тестування слів, вивчених користувачем (рис. 2.27). Дійові об’єкти: User, UI, LC. Після відкриття екрану вибору слів для тестування, UI створює запит на формування view model і передає його LC. LC створює список слів, які знаходяться в базі даних, та формує модель для інтерфейсу, передає модель до UI, який формуватиме інтерфейс.

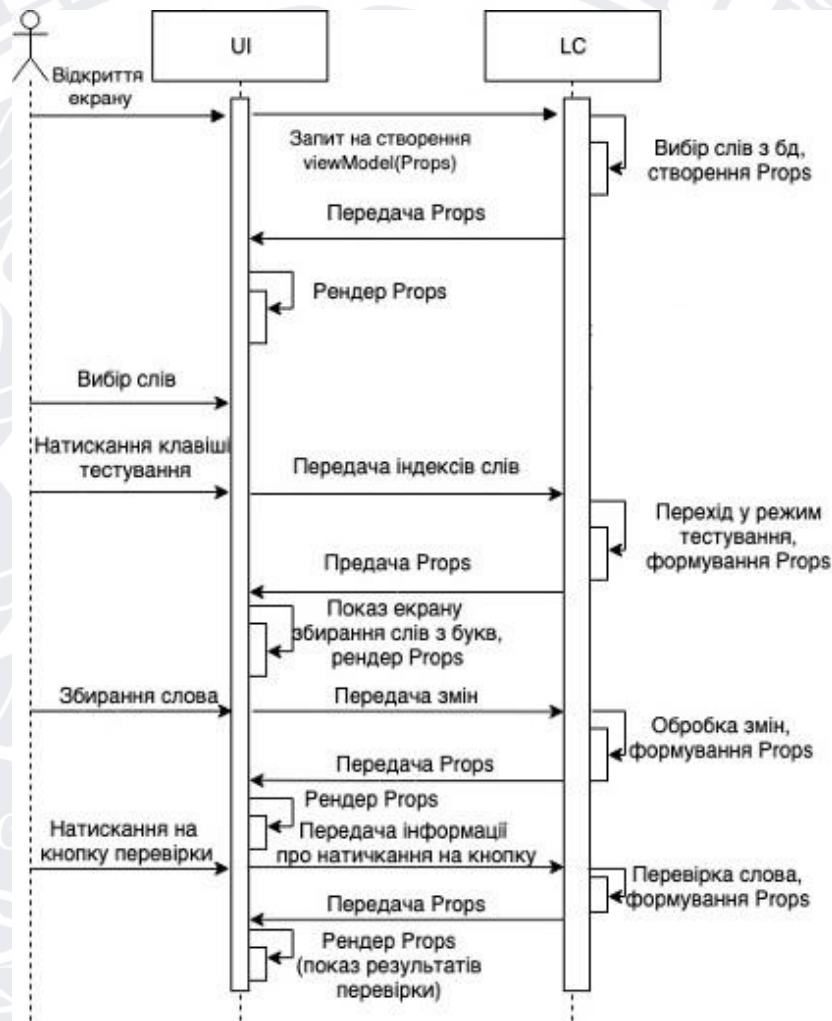


Рисунок 2.27 – Діаграма послідовності тестування вивчених користувачем слів

Після того, як користувач вибрав слова із списку, він натискає на клавішу тестування. UI та LC переходять у режим тестування, LC отримує індекси необхідних слів, формує нову модель. UI показує користувачеві змінений екран, де потрібно розставляти букви у правильному порядку. Кожна зміна положення букви викликає запит на створення нової view model'i, чим і займається LC. Після того, як користувач завершив розміщення букв, він натискає на кнопку

перевірки.

LC перевіряє правильність, формує нову модель з результатом перевірки, UI показує результат.

2.5 Проектування бази даних ІС підтримки вивчення англійської мови

Для побудови правильної та лаконічної бази даних для інформаційної системи підтримки вивчення англійської мови проектування розділено на три етапи:

1. Побудова ER-моделі, а на її основі — концептуальної моделі з виділенням основних моделей сутностей.
2. Уточнення концептуальної моделі через побудову логічної моделі.
3. Побудова фізичної моделі на основі логічної моделі із додаванням всіх типів даних.

Важливим також є те, що побудована модель бази даних має відповідати стандартам Apple, оскільки розроблювана ІС використовуватиме систему керування базою даних, розроблену та підтримувану саме цією компанією. Модель БД для сервера матиме аналогічну будову.

2.5.1 Концептуальна модель предметної області ІС підтримки вивчення англійської мови

Концептуальна модель – це деяка наглядна діаграма, побудована з використанням загальноприйнятих означень, яка представляє зв'язки між об'єктами та їх характеристиками. Створюється концептуальна модель для подальшого проектування бази даних. Загальноприйнятими означеннями в концептуальних моделях є: об'єкт – реальна сутність або об'єкт, позначається прямокутником, та атрибут – характеристика об'єкта, яку необхідно відобразити і зберігати у БД, позначається овалом.

Після детального аналізу об'єктів, які зберігатимуться у БД для коректної роботи ІС, побудовано ER діаграму сутностей, зображену на рисунку 2.28.

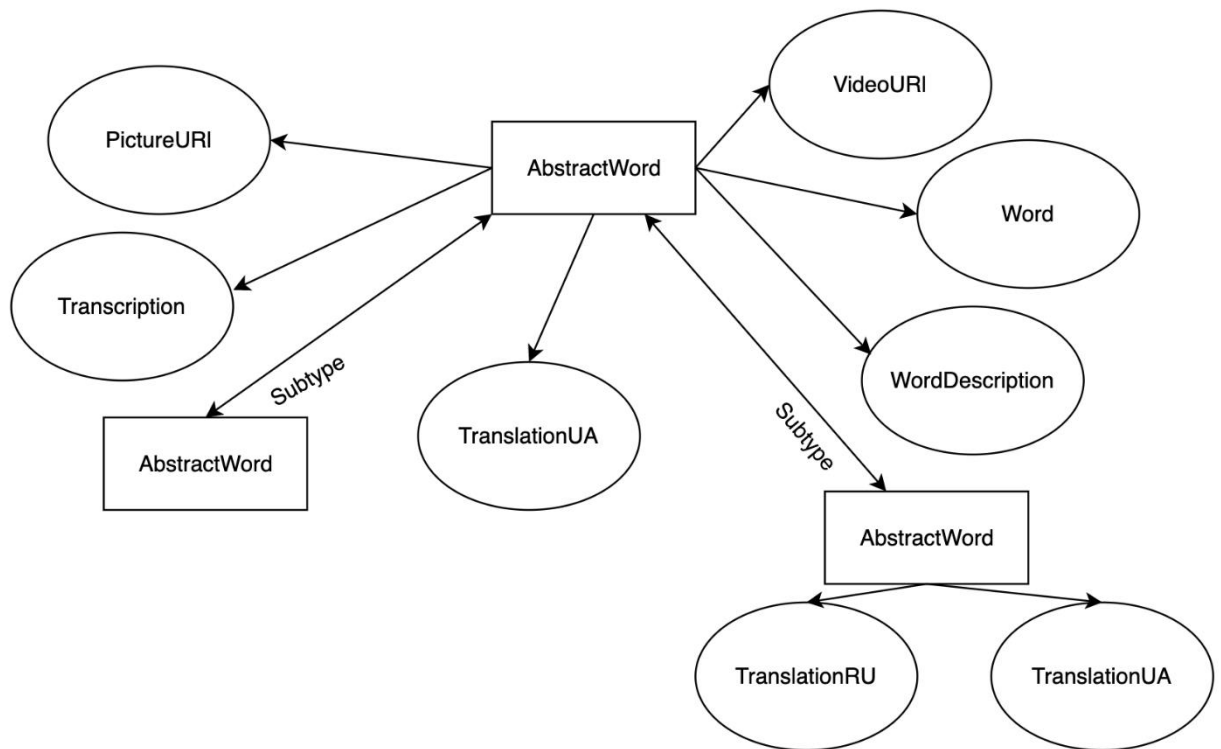


Рисунок 2.28 – ER-діаграма сутностей бази даних ІС вивчення англійської мови

Варто зазначити, що для швидкодії та оптимізації використання пам'яті об'єкт «AbstractWord» є абстрактним об'єктом, який описує спільні для підкласів характеристики. Об'єкти SelfWord та Word наслідують ознаки абстрактної сутності, при чому Word має розширення у вигляді двох додаткових ознак «isApproved» та «isKnown».

На основі ER-діаграми побудовано концептуальну модель, представлену на рисунку 2.29.

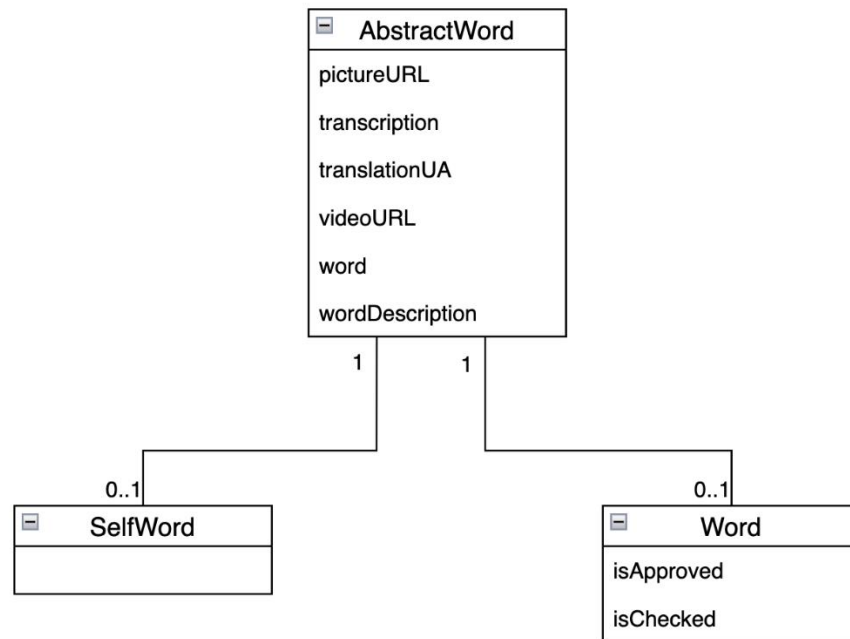


Рисунок 2.29 – Концептуальна модель бази даних ІС вивчення англійської мови

Наведена модель зображує об'єкти БД та зв'язки між ними. Надалі концептуальна модель буде використана для побудови логічної моделі бази даних.

2.5.2 Логічна модель бази даних ІС підтримки вивчення англійської мови

Логічна модель бази даних – модель даних конкретної предметної області, виражена незалежно від конкретного продукту керування базами даних або технології зберігання (фізична модель даних), але в термінах структур даних, таких як реляційні таблиці та колонки, об'єктно-орієнтовані класи чи теги XML [36]. Логічна модель є уточненням концептуальної моделі з урахуванням обмежень, що накладаються предметною областю, а також з урахуванням моделі даних.

На рисунку 2.30 зображена логічна модель бази даних ІС.

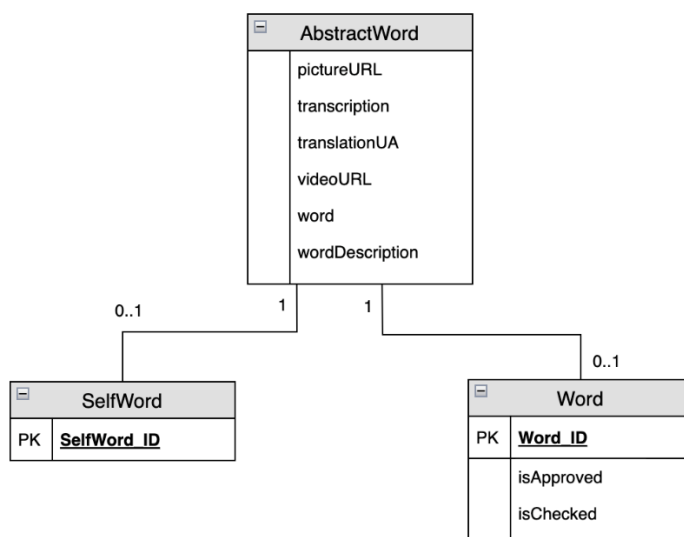


Рисунок 2.30 – Логічна модель бази даних ІС підтримки вивчення англійської мови

З рисунку 2.30 видно, що об'єкти SelfWord та Word мають первинні ключі, а AbstractWord – ні. Це пояснюється абстрактністю класу AbstractWord, створення об'єкта цього класу в обраній системі керування базою даних (CoreData) неможливе.

2.5.3 Фізична модель бази даних ІС підтримки вивчення англійської мови

Фізична модель – подання дизайну даних як реалізованого чи призначеного для реалізації у системі керування базами даних. У життєвому циклі проекту вона типово походить від логічної моделі даних, хоча може бути зворотно розроблена з конкретної реалізації бази даних. Завершена фізична модель даних включає всі артефакти бази даних, необхідні для створення відношень між таблицями чи для досягнення мети продуктивності, наприклад, індексів, визначень обмежень, зв'язаних і секціонованих таблиць або кластерів [37]. Графічне зображення моделі проілюстровано на рисунку 2.31.

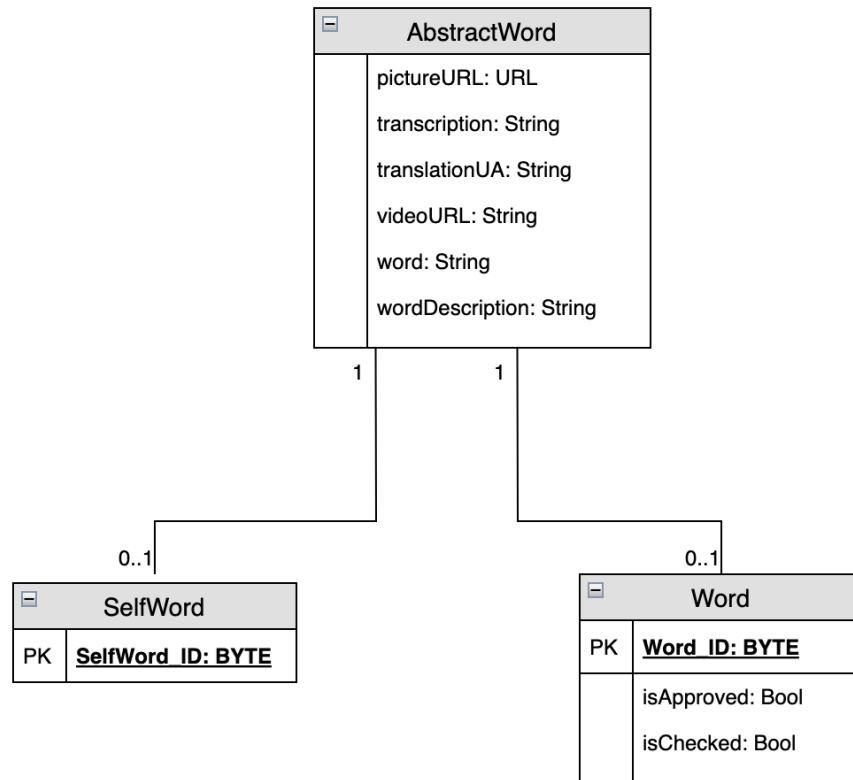


Рисунок 2.31 – Фізична модель бази даних ІС підтримки вивчення англійської мови

Фізичну модель розроблено на основі попередньо розробленої логічної моделі та розширено за допомогою введення обмежень на типи даних атрибутів кожного з об'єктів.

Висновки до розділу 2

Другий розділ роботи містить аналіз предметної області, виділені й затверджені основні вимоги до розроблювальної ІС. Такими є:

- автономність;
- швидкість роботи;
- простота підтримки;
- збір аналітичних даних;
- зручність користування;
- забезпечення роботи в офлайн-режимі;

- перегляд списку слів, збережених у базі даних;
- додавання нових слів;
- відправка слів на перевірку;
- визначення назви реального об'єкта за допомогою камери та машинного зору.

На основі сформованих вимог спроектовано та описано основні діючі об'єкти інформаційної системи, а також представлено діаграми прецедентів та класів.

Визначені класи стали основою для подальшого проектування пакетів та модулів.

Також описано розгортання системи на апаратних засобах.

Проаналізовано та описано усі основні функції ІС. Також на основі результатів досліджень побудовано та представлено діаграми діяльності та послідовностей кожного модуля.

Заключним етапом проектування є етап побудови бази даних. Проведено аналіз предметної області ІС, отримані результати проілюстровано на ER-діаграмі. На основі ER-діаграми створено концептуальну модель, а на її основі – логічну модель з метою виділення основних властивостей, притаманних об'єктам з предметної області. На основі логічної моделі побудовано фізичну модель БД, де вказані всі обмеження для атрибутів кожної з сутностей. Кожну із зазначених моделей зображено у графічному вигляді.

Підсумком проведених робіт у другому розділі стала розробка повної архітектури програмного забезпечення ІС системи підтримки вивчення англійської мови.

РОЗДІЛ 3. РОЗРОБКА І ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ

3.1 Розробка системи

Розробка ІС – заключний етап створення програмного забезпечення і зазвичай вимагає найбільше зусиль та витрат часу, адже саме на цьому етапі відбувається реалізація раніше спроектованих логічних модулів та їх поєднання у готовому продукті.

Розробка будь-якого програмного продукту починається з вибору стеку технологій, які будуть використані при переході від теоретичної моделі системи до реальних прикладних рішень.

Для розроблювальної інформаційної системи підтримки вивчення англійської мови необхідно обрати мову програмування, якою буде реалізований додаток під iOS пристрій та серверна частина ІС, СКБД для мобільного пристрою та хмарного сховища, систему керування сторонніми бібліотеками, систему керування версіями та IDE.

3.1.1 Огляд та вибір мови програмування

Оголошені до розроблювальної ІС підтримки вивчення англійської мови вимоги не містять пункту, який зобов'язує розробити кросплатформний додаток, тому вибір мови програмування зводиться до двох нативних для iOS мов програмування – Objective-C та Swift. Нижче приведено огляд кожної з них.

Objective-C – високорівнева мова програмування, яка виникла на початку 80-х років у вигляді пакета розширення для мови програмування C. До 2014 року активно використовувалась для розробки додатків під операційні системи MacOS, iOS, WatchOS.

До переваг Objective-C відносять наступне:

- проста взаємодії з C, C++ та Objective-C++;
- динамічність;
- method swizzling;
- простота написання бінарних фреймворків.

Недоліками є:

- Objective-C — надбудова мови C, що означає обмеженість простору імен. Усі класи в додатку Objective-C повинні бути унікальними. З метою уникнення колізій створено умову префіксації назв класів;
- строгі (сильні) вказівники;
- здатність надсилати повідомлення неіснуючому об'єкту без збоїв, що приводить до помилок, які важко відслідкувати та виправити;
- складний синтаксис.

Ця мова програмування є морально застарілою, усе рідше використовується у розробці.

Swift — перспективна мова програмування, яка активно рекламується компанією Apple та повсякчас використовується розробниками як альтернатива Objective-C.

До переваг мови належать:

- статична типізація;
- підтримка просторів імен;
- функціональні патерни;
- простий синтаксис;
- простота читання коду;
- можливість розробки серверних рішень;
- швидкість виконання.

На противагу, недоліками використання цієї мови є:

- довготривала компіляція;
- відсутність методу використання напряму C++бібліотек;
- складність створення бінарних фреймворків.

Порівняння описаних мов програмування окреслило особливості, що дозволили зробити вибір на користь Swift для розробки IC, а саме:

1. Відсутність потреби розбиття класів на файли інтерфейсу та імплементації, що значно скорочує кількість коду та витрату часу для реалізації.
2. Простіший синтаксис створення полів та властивостей класу.
3. Підтримка Optional типів.
4. Відсутність потреби явного вказування тип, якщо він зрозумілий з декларації.
5. Наявність оптимальної системи керування пам'яттю.
6. Розширення можливостей для Switch та Enum.
7. Можливість використання функціонального програмування.
8. Підтримка узагальнюючих типів.

Для серверної частини, реалізації потребував лише простий інтерфейс, який відповідає за опрацювання запитів від мобільного додатку, їх перетворення і надсилання до СКБД, формування JSON-об'єктів. Оскільки функціонал доволі простий, для реалізації таких задач використано систему керування контентом Strapi.io [38].

Strapi.io — контент-менеджер система з відкритим вихідним кодом, призначена для швидкого створення API та репозиторію контенту. Особливістю цієї системи є підтримка локального хостингу backend-частини серверного додатку та можливість використання різних СКБД для зберігання вмісту системи. Також підтримується режим адміністрування та графічний інтерфейс, який дозволяє керувати контентом, інтерактивно додавати нові API.

3.1.2 Вибір системи керування базами даних

Вибір БД – важливий крок, оскільки стабільність і швидкодія роботи інформаційної системи в деяких випадках напряму залежить від швидкості обробки запитів до бази даних. Для розроблюваної інформаційної системи

необхідно обрати дві СКБД: для серверної частини та для клієнтської частини – мобільного додатку.

Для хмарних рішень існує доволі широкий вибір СКБД, проте, зважаючи на простоту моделі бази даних, одразу було відкинуто варіанти, які використовуються для реалізації великих і складних баз даних (такі як: MySQL, PostgreSQL). Для реалізації серверної частини розроблюваної інформаційної системи підтримки вивчення англійської мови розглянуто дві системи: SQLite та MongoDB.

SQLite — полегшена реляційна система керування базами даних. Реалізована у вигляді бібліотеки за допомогою мови програмування C. Є найпоширенішою СКБД у світі, вбудовується у всі мобільні пристрої та більшість комп'ютерів, які випускаються нині [39].

До переваг відносять:

- зберігання база даних в одному файлі;
- використання мови SQL;
- відкритість бази для розробки і використання системою.

З недоліків можна виділити наступне:

- необхідність створення додаткової логіки, для перетворення об'єктів бази даних в JSON-об'єкти;
- необхідність опису схеми таблиць.

MongoDB — документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів [40].

Для зазначеної СКБД виділяють наступні переваги:

- документо-орієнтоване сховище;
- зберігання даних у форматі JSON;
- підтримка динамічних запитів;

- ненадійність системи безпеки.

Для реалізації серверної частини інформаційної системи обрано систему керування базами даних SQLite. Ця СКБД може бути встановлена на систему керування контентом без додаткових налаштувань. Перевагою використання SQLite є підтримка її більшістю систем та мов програмування, що, у разі потреби, спрощує перехід backend-додатку на іншу платформу.

Для реалізації клієнтської частини ІС враховано, що клієнтом є мобільний додаток для операційної системи iOS, тому вибір здійснювався між двома системами: Realm та CoreData.

Realm – безкоштовна система керування базою даних, дані у якій зберігаються у вигляді об'єктів, доступна для розробників мобільних та ПК додатків. Є досить популярною СКБД завдяки простоті використання та швидкодії.

Перевагами Realm є:

- збереження даних у вигляді об'єктів;
- збереження даних в одному файлі;
- швидкодія;
- доступна і легка для розуміння документація;
- простота опису прототипів об'єктів.

Серед недоліків можна виділити наступне:

- є сторонньою бібліотекою, яка займає доволі багато дискового простору;
- після будь-якої зміни прототипів об'єктів даних, відбувається перебудова БД під час компіляції проекту, що потребує значних затрат часу.

CoreData – фреймворк від компанії Apple, вбудований в операційну систему iOS та MacOS, який дозволяє розробнику взаємодіяти з базою даних. Дозволяє зберігати дані за структурою «сутність-атрибут-значення» та керувати даними за допомогою маніпуляцій над сутностями та їхніми взаємозв'язками. CoreData не вважається повноцінною СКБД, радше — обгорткою для фізичної реляційної бази даних, яка являє собою сховище типу

SQLite.

Переваги CoreData:

- доступність в iOS SDK;
- зручність опису схеми таблиць за допомогою вбудованого графічного інтерфейсу;
- зберігання даних у вигляді об'єктів;
- підтримка функціональних патернів;
- можливість перегляду даних за допомогою SQL-запитів;
- швидкодія.

Недоліками даного вреймворку є:

- складність налаштування та використання;
- складність пошуку і виправлення помилок.

Для клієнтської частини обрано CoreData, оскільки ця система є нативною для операційної системи iOS, а також вбудована в iOS SDK, що не потребує зайвих витрат пам'яті на запам'ятовуючому пристрої.

3.1.3 Програмне забезпечення використане для розробки інформаційної системи

Операційною системою для розробки серверної частини інформаційної системи та для клієнтської частини обрано систему MacOS 10.15.4 Catalina.

Для розробки мобільних додатків для операційної системи iOS єдиним варіантом IDE є XCode від компанії Apple. Дана IDE має недоліки, проте залишається зручною для написання коду, отримання підказок, перегляду офіційної документації та побудови користувацького інтерфейсу.

Також для розробки мобільного додатку використано наступне прикладне програмне забезпечення:

1. Homebrew – утиліта командного рядка, яка дозволяє встановлювати пакети та програми.
2. Cocoapods – утиліта керування залежностями.

3. Bundler – менеджер для керування gem-залежностями.
4. Fastlane – інструмент автоматизації профілювання та збирання додатків.

3.1.4 Розробка і навчання нейронної мережі для використання в інформаційній системі

З огляду на те, що повна розробка нейронної мережі та її навчання вимагатимуть значних витрат часу та ресурсів, використано готову мережу. Такий підхід не позбавлений недоліків: готові нейронні мережі не відповідають формату, який можна застосовувати у системах для інтеграції в операційну систему iOS. Тому для розробки і впровадження інтелектуальної мережі розпізнавання об'єктів використано наступну стратегію:

- обрано достатньо навчену мережу від компанії Tansorflow [41];
- з метою забезпечення можливості донавчити модель на власних даних і привести її до формату, який використовується у застосунках для iOS «відігнуто» останній рівень нейронів;
- використано Python-скрипт, який виконуватиме перетворення мережі до необхідного формату;
- після приведення мережі до необхідного формату, сформовано тренувальні дані (спеціальні зображення із визначеними межами об'єкта, а також його класом-назвою);
- для донавчання мережі використано вбудовану в XCode утиліту для навчання CoreML моделей.

Після описаних вище дій, отримано нейронну мережу у форматі CoreML, яка може бути використана для визначення об'єктів на картинці. Для визначення об'єктів на відео, медіафайл проділено на окремі кадри, за допомогою попередньо сформованої моделі відбувається визначення об'єктів. Навчання мережі здійснено за допомогою методу зворотного поширення помилки.

3.1.5 Розробка інформаційної системи підтримки вивчення англійської мови

Загальна розробка інформаційної системи поділена на два етапи: розробка серверної частини (API для обробки запитів і відправки їх до СКБД та формування відповідей), розробка клієнтської частини – мобільного додатку.

Для забезпечення зв'язку сервера з клієнтом розроблено API, який перенаправлятиме запити від додатку до СКБД та повертатиме відповіді бази даних до мобільного пристрою. Опрацювання запитів до сервера та перетворення об'єктів бази даних в JSON-об'єкти здійснюються CMS Strapi.

Таке серверне рішення представлено лише з ознайомлювальною метою та потребує подальшого покращення та розвитку, проте виконує сформовані до розроблюваної ІС вимоги. Для реалізації API створено 3 види запитів:

- запит на отримання всіх записів бази даних;
- запит на видалення об'єкту запису за id;
- запит на додавання нового об'єкта.

Запит на отримання всіх записів БД повертає всі JSON-об'єкти, які відповідають за збереження даних про слова у словнику. Вхідних даних не потребує, HTTP метод – GET, endpoint - /words.

Запит на видалення об'єкта видаляє запит із визначеним id. Вхідні дані – id об'єкта, який потребує видалення, HTTP метод – DELETE, відповіддю є статус виконання операції, endpoint – /words.

Запит додавання нового об'єкта зберігає новий об'єкт у базі даних. Як результат виконання повертає статус обробки запиту, вхідні дані – JSON-об'єкт – структура, яка описує слово для словника, HTTP метод – POST, endpoint – /words.

Розробка клієнтської частини вимагає набагато більше зусиль і часу для реалізації, оскільки, окрім логічної частини, потрібно створити зручний та зрозумілий інтерфейс користувача.

Перший і головний екран, який зустрічатиме користувача під час запуску додатку – екран словника. Цей екран відображає список всіх доступних слів, функціонал пошуку слів, перегляду деталей і видалення слова, навігацію до інших екранів та оновлення словника. Загальний вигляд екрана показаний на рисунку 3.1.

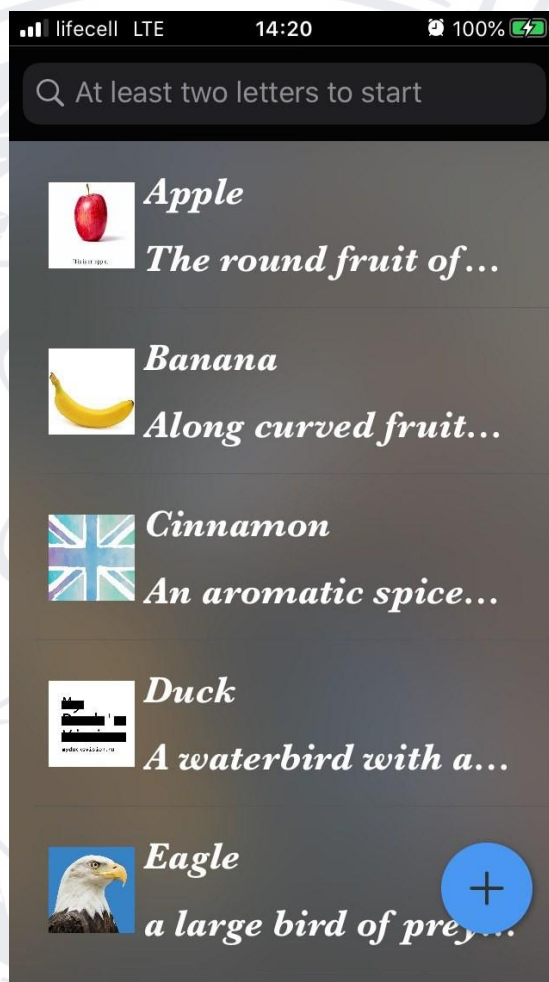


Рисунок 3.1 – Загальний вигляд екрану словника

Для здійснення пошуку слова потрібно вписати хоча б два символи у рядок пошуку. Якщо є записи, назва яких містить введену послідовність букв, словник відобразить ці записи. Окрім того, в рядках назви слів відповідна послідовність буде виділена жирним шрифтом та підкреслена. Приклад результату пошуку зображено на рисунку 3.2.

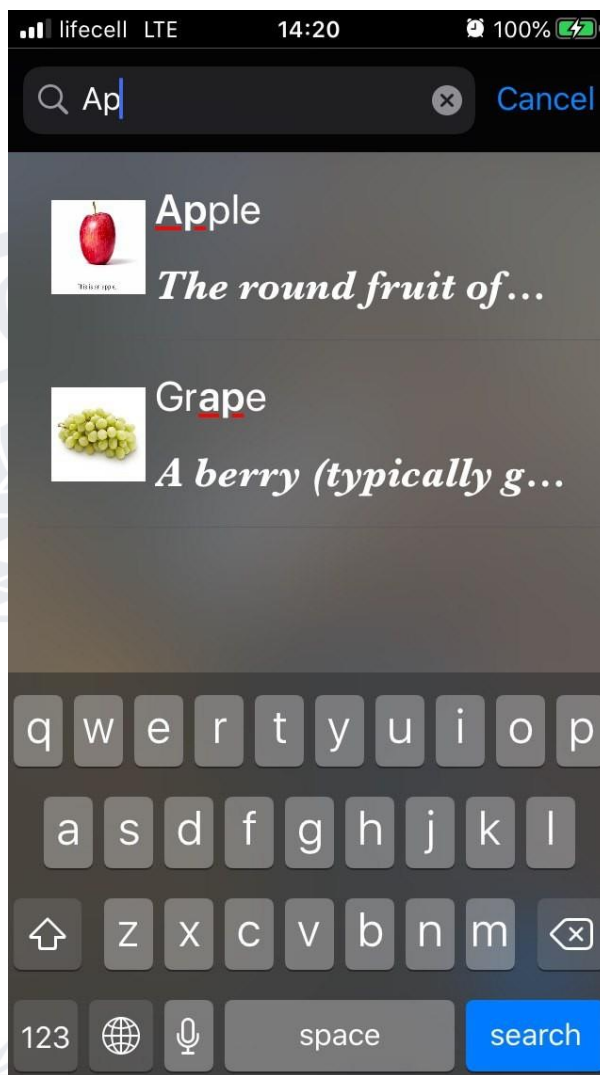


Рисунок 3.2 – Приклад роботи функції пошуку

Для перегляду детальної інформації про слово (транскрипцію, переклад українською мовою, аудіо- та медіафайли) потрібно натиснути на один із рядків списку слів. Приклад інтерфейсу перегляду деталей слова представлено на рисунку 3.3.



Рисунок 3.3. – Інтерфейс перегляду деталей слова.

Для видалення слова зі списку достатньо затиснути рядок зі словом та протягти вліво. Приклад видалення зображено на рисунку 3.4.

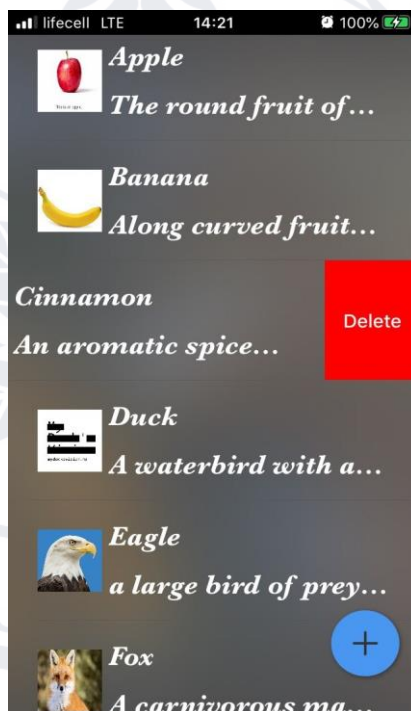


Рисунок 3.4. – Видалення слова зі списку

Для переходу між режимами та екранами користувачу необхідно натиснути на кнопку в правому нижньому куті. При натисканні з'явиться чотири кнопки, які перенаправлятимуть користувача до екрану з власноруч доданими словами, екрану додавання слова, режиму тестування та режиму визначення назви об'єкта за допомогою камери. Приклад навігації зображено на рисунку 3.5.

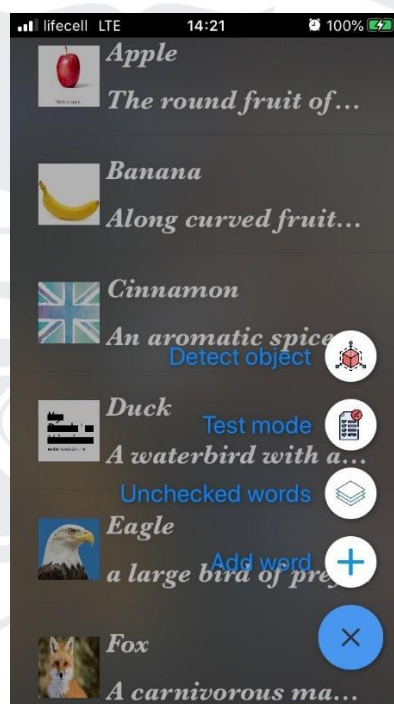


Рисунок 3.5 – Навігація на екрані словника

Екран додавання нового слова являє собою форму, в якій необхідно ввести дані про слово. Обов'язкові поля обведені червоною рамкою, яка змінює свій колір на білий по завершенню заповнення поля. Білими обведено необов'язкові для заповнення поля. Також екран має дві кнопки керування: «скасувати» – закриває екран без збереження та «готово», яка стає активною лише у разі заповнення всіх обов'язкових полів та відповідає за збереження слова в локальній БД і закриття поточного екрану.

Варто зазначити, що додаток розроблюється для операційних систем iOS 15 і вище, що дає змогу закрити будь-який екран (окрім словника) протягуванням екрану вниз (необхідно затиснути строку з назвою екрану і потягнути донизу). Графічний інтерфейс екрана додавання нового слова

представлений на рисунку 3.6.

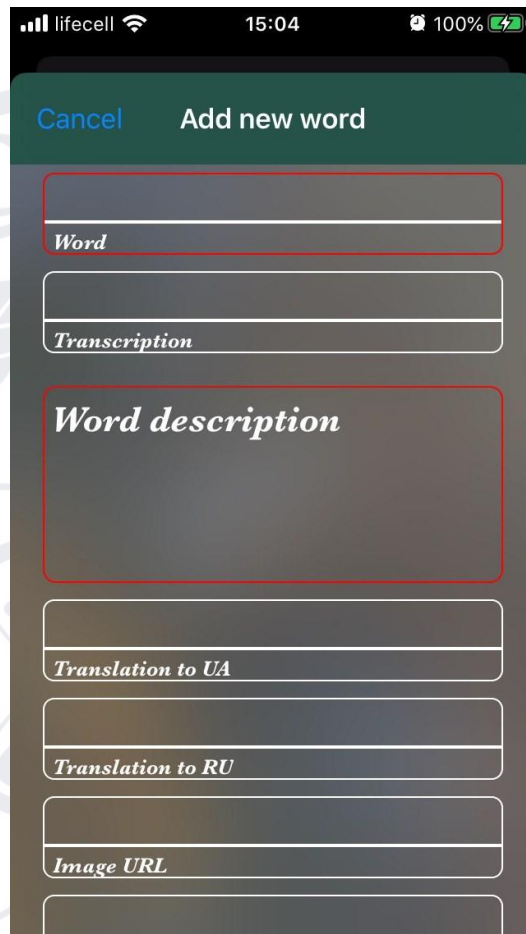


Рисунок 3.6 – Графічний інтерфейс екрану додавання нового слова

Екран власних слів відображає список неперевірених слів. Цей екран містить також функціонал відправки слів на сервер для перевірки, який ініціалізується натисканням на кнопку внизу екрана.

За допомогою вибору рядка зі словом і протягуванням його можна отримати доступ до видалення та редагування. Під час редагування відкривається екран додавання нового слова, проте вже з заповненими полями, які можна редагувати. Також якщо користувач натискає на кнопку «скасувати», яка знаходиться в верхньому лівому куті, але досі є неперевірені слова, додаток запропонує відправити їх на перевірку.

У верхньому правому куті розміщена кнопка з зображенням плюса, що дає можливість перейти до екрану додавання нового слова.

Користувачський інтерфейс описаного екрана представлено на рисунку 3.7.

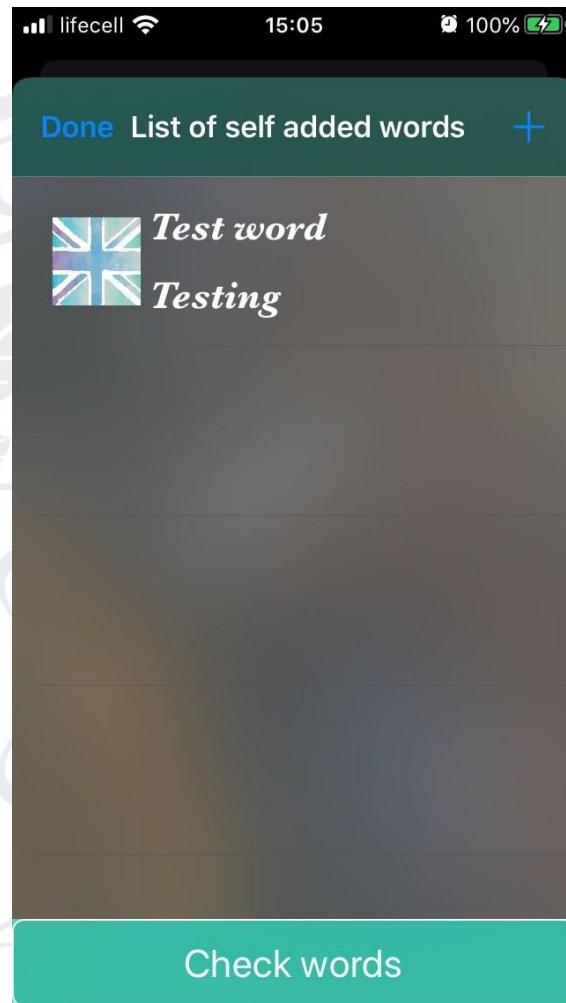


Рисунок 3.7 – Інтерфейс користувача екрану власноруч доданих слів

Режим тестування складається з двох екранів: вибору слів для тестування та екрана складання послідовності букв для формування відповіді. Екран вибору слів містить список, натискання на рядок якого відмічає дане слово як таке, яке буде використане у тестування. Верхній рядок екрана містить дві кнопки: одна для виходу, інша — для переходу до складання букв.

Екран складання слова з послідовності букв має наступну структуру:

- зверху розміщена кнопка виходу;
- нижче розміщується поле для відображення словесного опису слова, яке необхідно скласти з букв;
- посередині розміщене зображення, яке дозволяє зрозуміти чи

правильна відповідь;

- знизу розміщено дві кнопки: для пропуску питання та перевірки.

Після успішного проходження тесту, користувач побачить спливаюче вікно з привітанням, закриття якого приводить до виходу з режиму тестування. Інтерфейс екрана складання слова за допомогою послідовності літер представлено на рисунку 3.8.

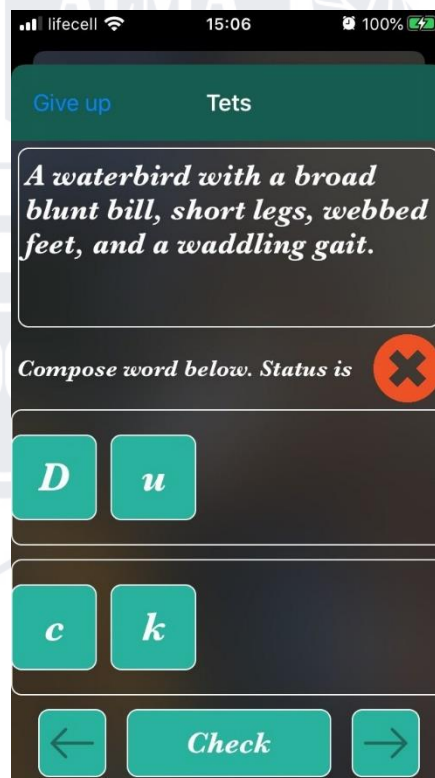


Рисунок 3.8 – Інтерфейс екрана тестування

Останній екран розроблюваної ІС підтримки вивчення англійської мови, який потребує реалізації в інтерфейсі користувача – режим визначення назви об'єкта за допомогою вбудованої в пристрій камери. Цей екран має наступний вигляд: зверху знаходиться кнопка виходу, центральну частину екрана займає контейнер, в якому відображається зображення з камери. Під ним розміщується текстова мітка, яка відображає назву об'єкта, згенеровану нейронною мережею. Нижче знаходиться кнопка пошуку слова у словнику. Якщо натиснути на неї, додаток розпочне пошук запису в локальній базі даних і у разі вдалого пошуку

відобразить екран з детальним описом. Якщо ж такого слова немає у БД – внизу з’явиться ще одна кнопка, яка забезпечує перехід до екрана додавання, із попередньо заповненим полем назви об’єкта. Графічно інтерфейс користувача даного екрану зображено на рисунку 3.9.

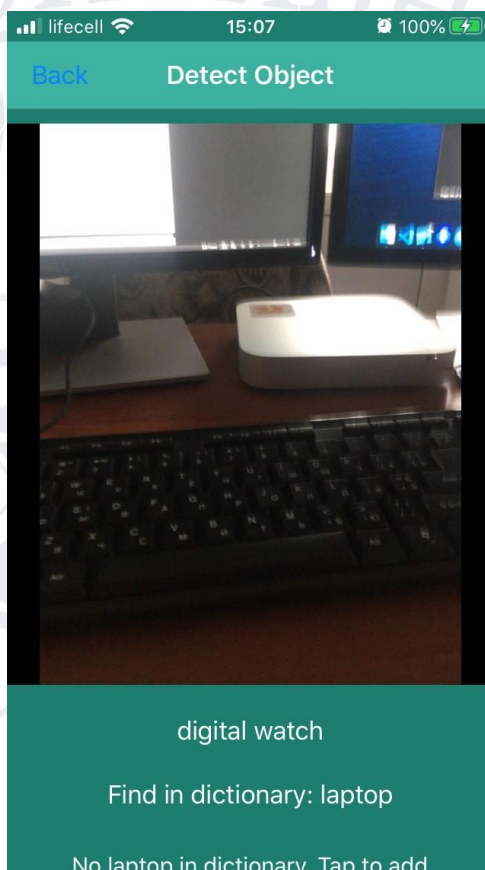


Рисунок 3.9. – Графічний інтерфейс користувача режиму визначення назви об’єкту за допомогою вбудованої камери

3.2 Тестування ІС підтримки вивчення англійської мови

Етап тестування та перевірки – заключний та невід’ємний етап розробки будь-якого програмного забезпечення. Саме на цьому етапі виявляються всі недоліки, які були допущені під час реалізації вимог. Кожен модуль ІС має бути систематично протестовано на логічні помилки та недоліки в користувацькому інтерфейсі для усунення несправностей до реалізації та передачі системи кінцевому користувачу.

Тестування проведено за допомогою методу «чорна скриня». Цей метод являє собою стратегію тестування функціональної поведінки об'єкта з точки зору зовнішнього світу, коли не використовуються знання про внутрішню структуру. Результати тестування системи наведено у таблиці 3.1

Таблиця 3.1 – Результати тестування ІС

| Назва модуля/функціоналу, який перевіряється | Спосіб тестування | Результат тестування |
|---|--|---|
| API | Виклик неіснуючого endpoint'a | Статус код відповіді 404, що є правильним |
| | Звернення під час недоступності сервера | Статус код 500, який може обробити мобільний додаток |
| Модуль словника додатка | Спроба оновлення локальної БД під час недоступності хмарного сховища | Система правильно зреагувала на помилку доступності та вивела на екран повідомлення про помилку |
| Модуль словника додатка | Відмова локальної БД | Система відображає повідомлення про помилку конфігурації БД і пропонує перезавантажити додаток або звернутись до техпідтримки |
| Модуль визначення назви об'єкта за допомогою камери | Заборона додатку використовувати камеру | Система відображає спливаюче вікно, яке попереджує про заборону використання камери і просить надати доступ |
| Модуль перевірки власноруч доданих слів | Відправка слів для перевірки на сервер під час недоступності сервера | Система відображає повідомлення про помилку та пропонує повторити спробу пізніше |
| | Спроба надіслати некоректні дані на сервер | Система відображає помилку та просить спробувати пізніше |

Як видно з таблиці 3.1, розроблена система коректно реагує на будь-які помилки, незалежно від того, стосуються вони помилки на сервері, локальної бази даних або ж заборон системи.

Висновки до розділу 3

У третьому розділі здійснено порівняльний аналіз мов програмування для розробки під операційну систему iOS, обрано оптимальну мову для забезпечення можливості подальшої підтримки розроблюваної інформаційної системи. Мовою розробки інформаційної системи підтримки вивчення англійської мови вибрано мову програмування Swift.

Проаналізовано наявні варіанти СКБД для локальної бази даних. Обрано фреймворк CoreData, який забезпечує швидкість і стабільність запитів до БД, а також надійність та простоту зберігання даних.

За допомогою вибраного стеку технологій та додаткових інструментів розроблено інформаційну систему підтримки вивчення англійської мови.

На етапі тестування проведено тестування за допомогою методу «чорна скриня», результати якого свідчать про відсутність необроблених помилок та небажаних ситуацій. Також перевірка показала, що всі поставлені вимоги виконано якісно і належним чином.

Система у майбутньому може бути покращена через удосконалення серверної частини, додавання режиму адміністрування та збільшення кількості методик, які б допомагали користувачу якісніше і швидше вивчати мову.

ВИСНОВКИ

У дипломній роботі проведено конструктивний аналітичний розбір проблеми складності вивчення іноземної мови, а також спроектовано, розроблено та протестовано інформаційну систему підтримки вивчення англійської мови для операційної системи iOS. Мобільний додаток розроблено для полегшення вивчення нових слів та повторення вже вивчених з використанням ігрової форми навчання.

Розроблена система має наступний функціонал:

- перегляд списку доступних слів у словнику;
- перегляд детального опису слова, що включає в себе відеоопис, словесний опис, переклади українською мовою, зображення, яка допомагає краще зрозуміти сенс поняття;
- додавання слів власноруч та надстлання їх на перевірку;
- наявність мінігри для перевірки вивчених слів;
- визначення назви об'єкта за допомогою вбудованої в пристрій камери та пошук його у словнику.

Інформаційну систему реалізовано мовою програмування Swift у середовищі розробки XCode. Також використано допоміжні утиліти: Homebrew, Fastlane, CocoaPods. Для реалізації локальної бази даних застосовано фреймворк CoreData від компанії Apple, вбудований в операційні системи MacOS та iOS.

Для досягнення мети дипломної роботи виконано наступні кроки:

1. Оглянуто та проаналізовано область застосування інформаційної системи підтримки вивчення англійської мови.
2. Здійснено порівняльний огляд існуючих систем вивчення англійської мови.
3. Сформовано список задач, які має виконувати ІС.
4. Змодельовано предметну область.
5. Сформовано вимоги до розроблюваної інформаційної системи.
6. Спроектовано логічну структуру системи та побудовано діаграми,

які ілюструють логічні процеси між частинами системи.

7. Розроблено архітектуру ІС.
8. Змодельовано поведінку системи у різних сценаріях.
9. Проведено проектування структури бази даних.
10. За допомогою запропонованих засобів реалізовано ІС. Також враховано поставлені вимоги та сформовані діаграми.
11. Виконано тестування готового продукту методом «чорна скриня».
12. Розроблену інформаційну систему можна використовувати як самостійну та повноцінну методику навчання, а також як додатковий спосіб навчання.

Система розроблялась з перспективою майбутнього покращення, тому легка щодо підтримки та масштабування. Покращити розроблену ІС можна за допомогою впровадження режиму адміністрування, удосконалення серверної частини та розширення списку методик навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The most spoken languages worldwide in 2022. Statista. URL: <https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide/> (дата звернення: 11.02.2022).
2. Статистика англійської мови: веб-сайт. URL: <https://teacheng.info/other/english-language-statistics> (дата звернення: 11.02.2022).
3. Coronavirus disease (COVID-19) pandemic. World Health Organization. URL: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019> (дата звернення: 14.02.2022)
4. Software Bugs Don't Shelter in Place: What app usage and error data reveal during COVID-19. Bugsnag. URL: <https://www.bugsnag.com/covid-19-app-usage-error-data-report> (дата звернення: 16.02.2022).
5. Бурік Д. Технотренди-2021: якими сервісами та пристроями будуть користуватися українці. *Економічна правда*. 8.03.2021. URL: <https://www.epravda.com.ua/publications/2021/03/8/671678/> (дата звернення: 16.02.2022).
6. Memrise. Learn to speak like the locals. URL: <https://www.memrise.com/> (дата звернення: 18.02.2022).
7. Неповний огляд програм для розподіленого повторення (Memrise vs Duolingo) URL: <https://bunyk.wordpress.com/2019/07/12/spaced-repetition-software/>
8. Doulingo. The world's #1 way to learn a language. URL: <https://www.duolingo.com/> (дата звернення: 18.02.2022).
9. Грицик Т. Які відомі платформи для вивчення іноземних мов надали українцям безкоштовний доступ – список. *AIN*. URL: <https://ain.ua/2022/04/03/yaki-vidomi-platformy-dlya-vyvchennya-inozemnyh-mov-nadaly-ukrayinczyam-bezkoshtovnyj-dostup-spysok/>
10. Meredithkreisa. The 39 Best Language Learning Apps of 2022 That Make Sure Studying Fun. URL: <https://www.fluentu.com/blog/best-language-learning-apps/> (дата звернення: 19.02.2022).
11. Хосе Мігель. Найефективніші типи навчання. Ресурси для самодопомоги. URL: <https://www.recursosdeautoayuda.com/uk/типи-навчання/>
12. Бобко О. Управління знанням в інформаційних системах: соціально-філософські аспекти. *Науковий вісник Технократ*. 21.05.2010 by technocrat. URL: <http://technocrat.org.ua/upravlinnya-znannjam-v-informatsiynih-s/>
13. Лист міністерства Освіти і науки України від 01. 07. 2019 р. № 1/11-5966. "Методичні рекомендації щодо викладання іноземних мов у 2019/2020 навчальному році" URL: <https://osvitoria.media/metodychni-rekomendatsiyi-shhodo-vykladannya-inozemnyh-mov-u-2019-2020-navchalnomu-rotsi/>

14. Rosetta Stone. 30 years of intuitive language learning. URL: https://www.rosettastone.eu/lp/s3/?cmp=30years&utm_campaign=AW_Brand_NU_EMEA_ENG_Rosetta_Stone&gclid=Cj0KCQJw3v6SBhCsARIsACyrRAmBbvYF0IrvEw-UNKnAmnY8U62x1UqYtgi-FZKz4vDA_x887jTR6YUaAqZuEALw_wcB (дата звернення: 18.02.2022).
15. Обзор Rosetta Stone. Ресурс Rosetta Stone. URL: <https://coba.tools/rosetta-stone>
16. Vikash Gupta. Memrise Review – Is It Worth or Waste of time? Study French Spanish. URL: <https://www.studyfrenchspanish.com/memrise-review/>
17. Review Of The Language Learning App “Duolingo”. Specialist Language Courses. URL: <https://specialistlanguagecourses.com/review-language-learning-app-duolingo/> (дата звернення: 18.02.2022).
18. Benny Lewis. An Honest Rosetta Stone Stine Review: Does Rosetta Stone Really Work? URL: <https://www.fluentin3months.com/reviews/rosetta-stone-review/>
19. Марченко А.В. Проектування інформаційних систем. Навчальний посібник. Київ: Державний Університет Телекомунікацій. 2016. 69 с.
20. Unified Modeling Language – uml.org: веб-сайт. URL: <https://www.uml.org/what-is-uml.htm> (дата звернення: 03.03.2022).
21. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем: навчальний посібник / В.С. Авраменко, А.С. Авраменко. – Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. – 434 с.
22. Гриценко В.Г. UML-моделювання інформаційно-аналітичної системи. URL: https://lib.iitta.gov.ua/706851/1/aisnn_uml.pdf
23. Діаграма прецедентів – Вікіпедія: веб-сайт. URL: https://uk.wikipedia.org/wiki/Діаграма_прецедентів (дата звернення: 03.03.2022).
24. Fast Intuitive Modeling & Design. URL: <https://sparxsystems.com>
25. Діаграма класів – Вікіпедія: веб-сайт. URL: https://uk.wikipedia.org/wiki/Діаграма_класів (дата звернення: 05.03.2022).
26. Елементи UML. Діаграма випадків використання. URL: <https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml-elements.html>
27. Діаграма пакетів – Вікіпедія: веб-сайт. URL: https://uk.wikipedia.org/wiki/Діаграма_пакетів (дата звернення: 05.03.2020).
28. Іванов Л.В. Об'єктно-орієнтоване програмування. URL: http://iwanoff.inf.ua/oop_kn/LabTraining05.html
29. Shaw, Mary; DeLine, Robert; Klein, Daniel V.; Ross, Theodore L.; Young, David M.; Zelesnik, Gregory (1995-04). Abstractions for Software Architecture and Tools to Support Them. URL: <https://www.microsoft.com/en-us/research/people/rdeline/?from=http%3A%2F%2Fresearch.microsoft.com%2Fen-us%2Fum%2Fpeople%2Frdeline%2Fpublications%2Funicorn-tse-95.pdf>

30. Авраменко В. С. Авраменко С. А. Проектування інформаційних систем. Черкаси: Чабаненко Ю. А., 2017. 434 с.
31. Діаграма розгортання – Вікіпедія: веб-сайт. URL: https://uk.wikipedia.org/wiki/Діаграма_розгортання (дата звернення: 09.03.2022).
32. Теорія систем керування: підручник / В.І. Корнієнко, О.Ю. Гусєв, О.В. Герасіна, В.П. Щокін; М-во освіти і науки України, Нац. гірн. ун-т. Дніпро: НГУ, 2017. 497 с.
33. Діаграма діяльності – Вікіпедія: веб-сайт. URL: https://uk.wikipedia.org/wiki/Діаграма_діяльності (дата звернення: 09.03.2022).
34. Діаграма послідовності – Вікіпедія: веб-сайт. URL: https://uk.wikipedia.org/wiki/Діаграма_послідовності (дата звернення: 09.03.2022).
35. Benedikt Bollig (2006). 7. Message Sequence Charts. Formal Models of Communicating Systems. Springer.
36. Логічна модель даних – Вікіпедія: веб-сайт. URL: https://uk.wikipedia.org/wiki/Логічна_модель_даних (дата звернення: 10.03.2022).
37. Фізична модель даних – Вікіпедія: веб-сайт. URL: https://uk.wikipedia.org/wiki/Фізична_модель_даних (дата звернення: 10.03.2022).
38. Strapi — strapi.io: веб-сайт. URL: <https://strapi.io/> (дата звернення :15.03.2022)
39. SQLite – sqlite.org: веб-сайт. URL: <https://www.sqlite.org/about.html> (дата звернення: 15.03.2022).
40. MongoDB – mongodb.com: веб-сайт. URL: <https://www.mongodb.com/what-is-mongodb> (дата звернення 15.03.2022).
41. Tansorflow – tensorflow.com: веб-сайт. URL: <https://www.tensorflow.org/about> (дата звернення 17.03.2022).