

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Штучного інтелекту  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розробка мобільного додатку для вивчення іноземних мов дітьми  
(тема)

Виконав:  
здобувач четвертого року навчання,  
групи ІТШ-21-1

Міхаїл Гунбін  
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-професійна  
Освітня програма Штучний інтелект  
(повна назва освітньої програми)

Керівник доц. Ігор Магдаліна  
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ \_\_\_\_\_  
(підпис)

Олег ЗОЛОТУХІН  
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_

Кафедра \_\_\_\_\_ Штучного інтелекту \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ перший (бакалаврський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 122 Комп'ютерні науки \_\_\_\_\_  
(код і повна назва)

Тип програми \_\_\_\_\_ освітньо-професійна \_\_\_\_\_

Освітня програма \_\_\_\_\_ Штучний інтелект \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві \_\_\_\_\_ Гунбіну Михайлу Юрійовичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка мобільного додатку для вивчення іноземних мов дітьми

затверджена наказом університету від 19 травня 2025 р. № 378Ст

2. Термін подання студентом роботи до екзаменаційної комісії 18 червня 2025 р.

3. Вихідні дані до роботи офіційні документації до UIKit, Swift, SwiftData, AVFoundation, Xcode, а також методичні рекомендації щодо створення користувацьких інтерфейсів, проектування мобільних застосунків та лексичного навчання дітей.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1) Аналіз предметної галузі та постановка задачі

2) Проектування системи

3) Програмна реалізація



## РЕФЕРАТ

Пояснювальна записка: 88 с., 23 рис., 1 табл., 1 дод., 20 джерел.

ДИТИНА, ЛОКАЛІЗАЦІЯ, МОВНЕ НАВЧАННЯ, МОБІЛЬНИЙ ДОДАТОК, СТОХАСТИЧНИЙ РЕЖИМ, AVFOUNDATION, FLASHCARD MODEL, MIXED MODE, SWIFT, SWIFTDATA, UIKIT.

Об'єктом дослідження є процес вивчення іноземної лексики за допомогою цифрових технологій у мобільному середовищі.

Предметом дослідження є методи персоналізованого навчання лексики з використанням мобільних додатків і мультимовного контенту.

Метою роботи є створення мобільного додатку для дітей, що забезпечує вивчення лексики декількома мовами з урахуванням індивідуального прогресу.

У дослідженні використано методи об'єктно-орієнтованого проектування, проектування інтерфейсу користувача, методи локалізації, моделювання даних та використання стохастичного повторення.

У результаті реалізовано повноцінний мобільний додаток із підтримкою мультимовності, статистики навчання, озвучення, темного режиму та інтерфейсу, адаптованого для дітей.

Запропоноване рішення рекомендується для використання в освітніх закладах різного типу, а також у середовищі українських дітей-біженців, які проживають за кордоном та потребують підтримки в збереженні й розвитку мовної компетенції рідною мовою поряд із вивченням мови країни перебування.

## ABSTRACT

Bachelor's thesis contains: 88 pp., 23 fig., 1 tabl., 1 ann., 20 references.

AVFOUNDATION, CHILD, FLASHCARD MODEL, LANGUAGE LEARNING, LOCALIZATION, MIXED MODE, MOBILE APPLICATION, STOCHASTIC MODE, SWIFT, SWIFTDATA, UIKIT.

The object of the research is the process of foreign vocabulary acquisition through digital technologies in a mobile environment.

The subject of the research is the methods of personalized vocabulary learning using mobile applications and multilingual content.

The aim of the work is to develop a mobile application for children that enables vocabulary learning in several languages with personalized progress tracking.

The research applies structural and object-oriented design methods, user interface design, localization techniques, data modeling, and stochastic repetition.

As a result, a full-featured mobile application was implemented, with multilingual support, learning statistics, audio playback, dark mode, and a child-friendly interface.

The proposed solution is recommended for use in various types of educational institutions as well as among Ukrainian refugee children living abroad, who require support in preserving and developing language competence in their native tongue alongside learning the language of the host country.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	8
Вступ .....	9
1 Аналіз предметної галузі та постановка задачі .....	11
1.1 Актуальні тенденції цифрової освіти для дітей дошкільного віку ....	11
1.1.1 Роль мобільних додатків у сучасній освіті .....	11
1.1.2 Особливості навчання дітей дошкільного віку .....	13
1.1.3 Значення мультимодального підходу для вивчення іноземних мов.....	14
1.2 Огляд конкурентних рішень .....	15
1.2.1 Огляд найпопулярніших додатків .....	16
1.2.2 Порівняльний аналіз функціоналу.....	21
1.2.3 Визначення переваг та недоліків існуючих рішень .....	23
1.3 Вимоги цільової аудиторії до функціональності та інтерфейсу мобільного додатку .....	24
1.3.1 Виявлення цільової аудиторії.....	25
1.3.2 Потреби батьків та викладачів .....	25
1.3.3 Адаптація інтерфейсу під дитяче сприйняття .....	26
1.4 Визначення ключових функцій та можливостей.....	27
1.4.1 Тематичні лексичні колекції для різноманітного мовного навчання .....	27
1.4.2 Цифрові флеш-картки як засіб мультимодального засвоєння лексики .....	28
1.4.3 Підтримка багатомовності та прослуховування еталонної вимови носія .....	30
1.4.4 Реалізація стохастичного режиму тренування словникового запасу.....	30
1.5 Постановка задачі .....	31
2 Проектування системи .....	32

2.1	Аналіз вимог до мобільного додатку.....	32
2.1.1	Вимоги до користувацького інтерфейсу .....	33
2.1.2	Функціональні вимоги .....	34
2.2	Опис функціональної моделі.....	35
2.3	Схема використання додатку .....	37
2.4	Архітектура мобільного додатку .....	38
2.4.1	Вибір платформи розробки та обґрунтування.....	39
2.4.2	Обрана структура додатку .....	40
2.4.3	Модулі системи та їх взаємодія .....	42
2.5	Проектування бази даних та структури зберігання даних .....	43
3	Програмна реалізація .....	45
3.1	Технологічна основа мобільного додатку.....	45
3.1.1	Мова програмування.....	46
3.1.2	Фреймворки та бібліотеки .....	47
3.2	Реалізація моделей даних у SwiftData .....	48
3.3	Реалізація функціональності навчання з використанням флеш-карток .....	53
3.4	Механізм озвучення слів за допомогою AVFoundation.....	55
3.5	Організація мультимовності та локалізація ресурсу.....	59
3.6	Локальне зберігання прогресу та конфігурацій користувача .....	62
3.7	Логіка стохастичного режиму повторення .....	66
3.8	Основні екрани користувацького інтерфейсу.....	69
3.8.1	Головний екран.....	70
3.8.2	Екран флеш-картки .....	76
3.8.3	Екран налаштувань.....	81
	Висновки.....	84
	Перелік джерел посилання .....	86
	Додаток А Відомість кваліфікаційної роботи.....	88

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

JSON – JavaScript Object Notation – формат обміну даними;

MVVM – Model-View-ViewModel – модель представлення з проміжним шаром логіки;

UI – User Interface – інтерфейс користувача;

UX – User Experience – користувацький досвід.

## ВСТУП

У сучасних умовах активного розвитку цифрових технологій та глобалізації особливої ваги набуває питання доступності якісної освіти для дітей незалежно від їхнього місця перебування. Одним із ключових напрямів цифрової трансформації в освіті є створення мобільних застосунків, які дозволяють організувати навчання у зручній та індивідуалізованій формі. Мобільні пристрої стають основним інструментом пізнання навколишнього світу для дітей, а тому використання спеціалізованих додатків у ранньому віці вже не є новаторським підходом, а скоріше – необхідністю. Особливо гостро ця потреба постає для сімей, які змушені були покинути територію України через війну, адже діти, опинившись у новому мовному та культурному середовищі, стикаються з подвійним викликом – з одного боку, необхідністю адаптації до нової мови, а з іншого – ризиком втрати рідної мовної ідентичності.

Питання збереження української мови серед дітей-біженців набуває все більшої актуальності. У багатьох країнах, де нині проживають українські діти, провідними у повсякденному житті є інші мови, зокрема польська, німецька, англійська, чеська та інші. У процесі інтеграції у нове середовище українські діти починають активно засвоювати місцеву мову, що з одного боку полегшує процес адаптації, але з іншого – ставить під загрозу підтримку рівня володіння українською мовою. Відсутність системного підходу до збереження рідної мови у цифровому середовищі створює реальну загрозу її витіснення з повсякденного вжитку дитини. З огляду на це, виникає потреба у створенні такого освітнього інструменту, який одночасно забезпечуватиме можливість вивчення іноземних мов та підтримуватиме зв'язок дитини з мовою її Батьківщини.

Одним із найбільш ефективних форматів такого інструменту є мобільний додаток на базі флеш-карток, який поєднує мультимодальний підхід до навчання з ігровою формою взаємодії. У віці 2–7 років діти

сприймають інформацію переважно через зорові та слухові канали, а також активно використовують тактильні дії. Флеш-картки, що містять зображення, текст, транскрипцію та озвучення, ідеально відповідають цим особливостям, сприяючи одночасному формуванню лексичного запасу кількома мовами. За умови реалізації багатомовної підтримки, можливості перемикавання мов, прослуховування правильного звучання слів і системи тематичних колекцій, такий додаток стає універсальним засобом початкового мовного навчання для дітей у різних соціокультурних умовах.

Метою цієї кваліфікаційної роботи є розробка мобільного додатку для вивчення слів іноземною мовою дітьми дошкільного віку з урахуванням потреб українських дітей-біженців, які перебувають у багатомовному середовищі. Запропонований застосунок має забезпечити зручне вивчення базової лексики через мультимодальні флеш-картки, підтримку української мови як основної, можливість перемикавання мов відповідно до середовища проживання, просту навігацію та інтуїтивно зрозумілий інтерфейс. Окрему увагу буде приділено адаптації візуального стилю, елементів взаємодії та навчального контенту до вікових особливостей користувачів.

Результати даної роботи можуть бути використані для створення повноцінного освітнього інструменту у вигляді мобільного додатку, призначеного як для індивідуального навчання вдома, так і для занять у дошкільних навчальних закладах та освітніх центрах за кордоном. Крім того, розробка має потенціал для масштабування – додавання нових мов, розширення словникових тем, інтеграції системи мотивації та статистики для батьків. Таким чином, створення цього застосунку є не лише технічним завданням, а й важливим соціально-комунікативним проектом, спрямованим на підтримку мовної культури та освітнього розвитку українських дітей у глобальному контексті.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1 Актуальні тенденції цифрової освіти для дітей дошкільного віку

У сучасному світі цифрові технології стали невід'ємною частиною навчального процесу навіть у ранньому дитинстві. Зростаюча кількість мобільних додатків, спрямованих на розвиток мовних, математичних та когнітивних навичок, свідчить про високу зацікавленість як батьків, так і педагогів у використанні цифрових інструментів для формування базових знань у дітей дошкільного віку. Важливо зазначити, що такі додатки часто комбінують ігрову форму з навчальним змістом, що позитивно впливає на мотивацію дитини до навчання.

Однією з ключових тенденцій цифрової освіти є зміщення акценту з пасивного сприйняття інформації на активну взаємодію. Завдяки сенсорному управлінню, мультимедійним елементам і адаптивності інтерфейсу, мобільні застосунки забезпечують індивідуалізований підхід до навчання кожної дитини. Це особливо важливо для дітей віком 2–7 років, коли рівень абстрактного мислення ще формується, а наочно-дійове сприйняття домінує.

Сучасні дослідження вказують на ефективність використання мультимодальних методів у цифровій освіті. Поєднання зорових, слухових та тактильних каналів сприйняття дозволяє досягти глибшого закріплення нових знань, особливо у процесі вивчення іноземних слів. Додатки, які реалізують подібний підхід, не лише збагачують словниковий запас дитини, а й сприяють розвитку фонематичного слуху, пам'яті та концентрації уваги.

#### 1.1.1 Роль мобільних додатків у сучасній освіті

Мобільні додатки у сфері освіти сьогодні розглядаються як ефективний інструмент індивідуалізації навчального процесу. Їх

використання дозволяє адаптувати освітній контент до рівня знань, вікових та психологічних особливостей користувача. Особливо важливим це є у контексті дошкільної освіти, де кожна дитина навчається у власному темпі і потребує персоналізованого підходу. Візуальні елементи, озвучування, анімації та інтерактивна взаємодія створюють умови для гнучкого формування навчального середовища.

Окрім персоналізації, мобільні освітні рішення забезпечують високу залученість дитини у процес навчання. Завдяки гейміфікованому підходу, навчання сприймається як гра, що сприяє формуванню позитивної мотивації до вивчення нового матеріалу. Можливість швидко змінювати завдання, отримувати зворотний зв'язок і досягати результатів у вигляді візуальних досягнень підвищує інтерес дитини до повторного використання додатку.

Основні переваги мобільного навчання (рисунок 1.1), серед яких персоналізоване навчання, систематичний підхід, актуальність знань, ефективне використання часу, зростання залученості користувача та загальна перевага додатків серед інших форм навчального контенту. Ці характеристики формують підґрунтя для подальшого дослідження.

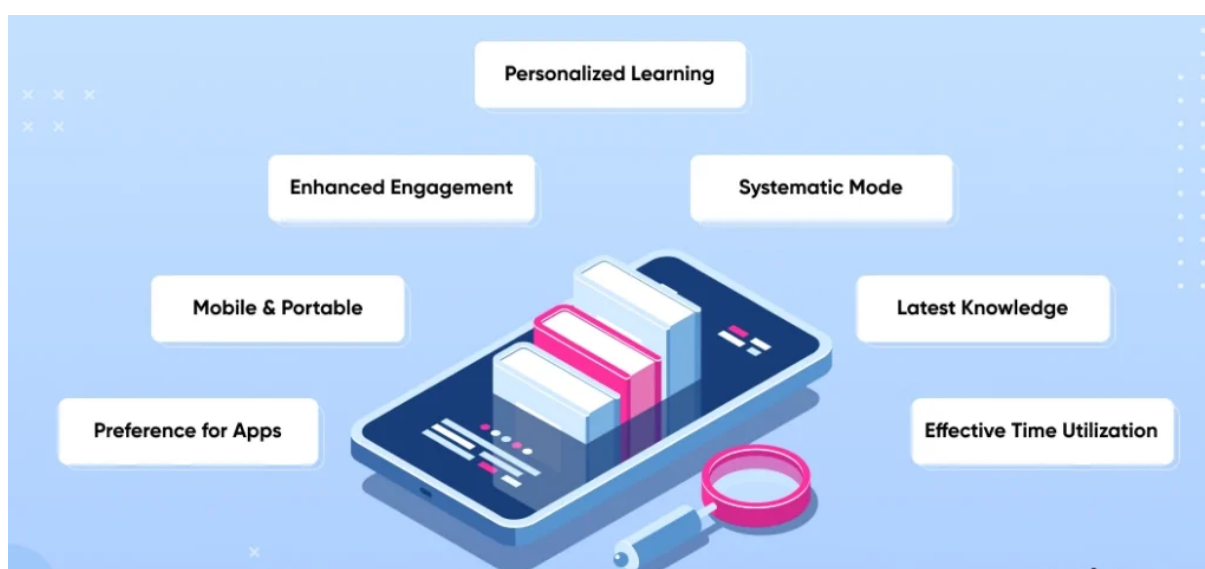


Рисунок 1.1 – Переваги мобільних додатків для освіти

Суттєвою перевагою мобільних рішень є їхня доступність та мобільність. Діти можуть навчатися у будь-який час і в будь-якому місці – у транспорті, вдома, на прогулянці з батьками. Це дозволяє ефективно використовувати вільний час і підтримувати регулярність навчання, що особливо актуально в умовах швидкого ритму сучасного життя. Багато додатків також працюють в офлайн-режимі, що дозволяє зменшити залежність від підключення до Інтернету.

### 1.1.2 Особливості навчання дітей дошкільного віку

Навчання дітей дошкільного віку має низку специфічних особливостей, пов'язаних із віковими психологічними та когнітивними характеристиками. У цьому віці домінує наочно-образне мислення, тому нова інформація найкраще сприймається через зорові образи, дії та ігрові ситуації. Діти вчаться через повторення, асоціації та емоційне підкріплення, що вимагає відповідного підходу до організації навчального процесу – з акцентом на візуалізацію, динамічність і короткі навчальні епізоди.

Важливим чинником ефективного навчання дошкільнят є обмежений обсяг уваги та низька стійкість до монотонної діяльності. Це означає, що навчальні матеріали повинні бути насичені різними типами стимулів: зміною кольорів, звуками, короткими анімаціями, інтерактивними елементами. Зміна активностей та чергування вправ є необхідною умовою підтримання зацікавленості дитини. Також критично важливо, щоб навчання відбувалося в атмосфері позитивного емоційного тону, без примусу, а як природне продовження ігрової діяльності.

Окрім того, діти цього віку не мають сформованої навички читання, тому інформація має бути представлена у вигляді простих візуальних символів, малюнків, озвучених інструкцій та прикладів. Взаємодія з додатком повинна бути максимально інтуїтивною – без складних меню, переходів чи необхідності текстового введення. Таким чином, успішне

навчання дошкільнят можливе лише за умови адаптації змісту, форми та темпу подання матеріалу до їхніх пізнавальних можливостей.

### 1.1.3 Значення мультимодального підходу для вивчення іноземних мов

Мультимодальний підхід у навчанні іноземним мовам передбачає використання кількох каналів сприйняття інформації одночасно – зорового, слухового, тактильного та, за можливості, мовленнєвого. Такий підхід особливо ефективний у роботі з дітьми дошкільного віку, оскільки відповідає природним механізмам засвоєння знань у ранньому віці. Поєднання візуального образу слова, його озвучування та інтерактивної взаємодії значно підвищує ймовірність запам'ятовування та розуміння лексичних одиниць.

У цифрових навчальних додатках мультимодальність реалізується через флеш-картки із зображенням, написанням слова, фонетичною транскрипцією та озвучкою. Діти мають змогу не лише побачити слово, а й почути його правильну вимову, а також взаємодіяти з ним через дотик – натискання на картку, перемикання мов тощо. Така комбінація стимулів забезпечує більш глибоку когнітивну обробку інформації, активуючи різні ділянки мозку.

Приклади навчальних підходів, що базуються на мультимодальності: гейміфіковане навчання, занурення в мовне середовище, вивчення словника, мовний обмін і комплексне опрацювання мови (рисунок 1.2). Вони ілюструють ключові напрямки, які успішно застосовуються у сучасних мовних застосунках і можуть слугувати основою для побудови ефективної освітньої платформи для дітей дошкільного віку.



Рисунок 1.2 – Ключові особливості додатків для вивчення мов

Важливим елементом мультимодального навчання є ігровий компонент, який сприяє емоційному залученню дитини. Гейміфіковане середовище не лише підтримує увагу, а й формує асоціативні зв'язки між поняттями. Це створює додаткові контексти для засвоєння лексики та забезпечує її довготривале збереження у пам'яті. Ігрові завдання, в яких необхідно зіставити зображення зі звуком або обрати правильну мову, також стимулюють розвиток логічного мислення.

## 1.2 Огляд конкурентних рішень

Для обґрунтування необхідності розробки нового мобільного додатку для вивчення іноземної лексики дітьми дошкільного віку доцільним є проведення аналізу існуючих конкурентних рішень. Такий аналіз дозволяє виявити найкращі практики, визначити функціональні та візуальні особливості популярних застосунків, а також оцінити їхню відповідність потребам цільової аудиторії. Порівняння можливостей, що вже реалізовані на ринку, дає змогу сформулювати уявлення про очікування користувачів і знайти потенційні напрями для вдосконалення майбутнього продукту.

Ринок дитячих освітніх додатків є динамічним та насиченим, проте значна частина рішень поєднує мовне навчання з іншими напрямками, такими як математика, логіка чи загальний розвиток. У цьому розділі увага зосереджена виключно на тих додатках, які спеціалізуються на засвоєнні іноземної мови в її базовому вигляді – тобто зосереджені на лексичних одиницях, їх візуалізації, озвученні та мовному контексті. Такий підхід дозволяє провести більш точне й обґрунтоване порівняння.

Нижче буде наведено короткий огляд найпопулярніших додатків, які безпосередньо конкурують між собою у сфері мобільного мовного навчання для дошкільнят. Йтиметься про ключові функціональні особливості, принципи взаємодії з користувачем, підходи до подачі лексики та використання мультимодальних елементів. Також буде проведено порівняльний аналіз основних характеристик додатків із виявленням сильних і слабких сторін кожного з них. Це дозволить зробити обґрунтовані висновки щодо можливостей поліпшення досвіду користувача. Завершальний підрозділ присвячено виділенню переваг та недоліків існуючих рішень у контексті їх подальшого врахування під час проектування власного додатку.

Загальний огляд конкурентного середовища є важливим аналітичним етапом, що допомагає визначити нішу для нового продукту, уникнути дублювання вже реалізованих рішень та сформулювати чіткі вимоги до функціоналу, інтерфейсу й методичного підходу мобільного додатку, що розробляється.

### 1.2.1 Огляд найпопулярніших додатків

У сфері мобільних додатків для мовного навчання дітей дошкільного віку вже існує низка перевірених рішень, що займають провідні позиції на ринку. Ці застосунки орієнтовані на базовий рівень володіння мовою та адаптовані під вікові особливості дітей 2–7 років. Їхній функціонал часто

включає ілюстровані флеш-картки, озвучення носієм мови, ігрову взаємодію та інтуїтивно зрозумілий інтерфейс. У цьому підпункті представлено огляд найбільш відомих мовних додатків, які можна вважати безпосередніми або функціонально близькими конкурентами.

Одним із найвідоміших є додаток FunEasyLearn (рисунок 1.3). Він пропонує багатомовне середовище з підтримкою понад 30 мов, серед яких англійська, іспанська, французька, німецька, китайська та інші. Усі слова супроводжуються яскравими зображеннями, фонетичною транскрипцією і професійною озвучкою носієм мови. В додатку передбачена можливість проходження тем у вигляді наборів, згрупованих за категоріями: тварини, їжа, транспорт, кольори тощо. Для дітей дошкільного віку додаток пропонує спрощений режим навчання, де виключено зайві функції, що можуть перевантажити сприйняття. Основна мета – допомогти дитині поступово накопичувати базову лексику без акценту на граматику чи складні мовні конструкції.



Рисунок 1.3 – Іконка додатку FunEasyLearn

Ще одним важливим конкурентом є Duolingo ABC (рисунок 1.4) – версія популярного додатку Duolingo, орієнтована саме на молодших користувачів. Додаток фокусується на вивченні англійської мови, особливо навичках читання та словниковому запасі. Він побудований на коротких вправоорієнтованих етапах з яскравими візуальними підказками, озвученням і мотивувальними реакціями. Слова та фрази подаються у вигляді флеш-карток, у поєднанні з інтерактивними елементами: треба провести пальцем по букві, обрати відповідний об'єкт або повторити почуте. Цей додаток має високу педагогічну якість і повністю безкоштовний, що робить його особливо популярним серед батьків та освітніх закладів.

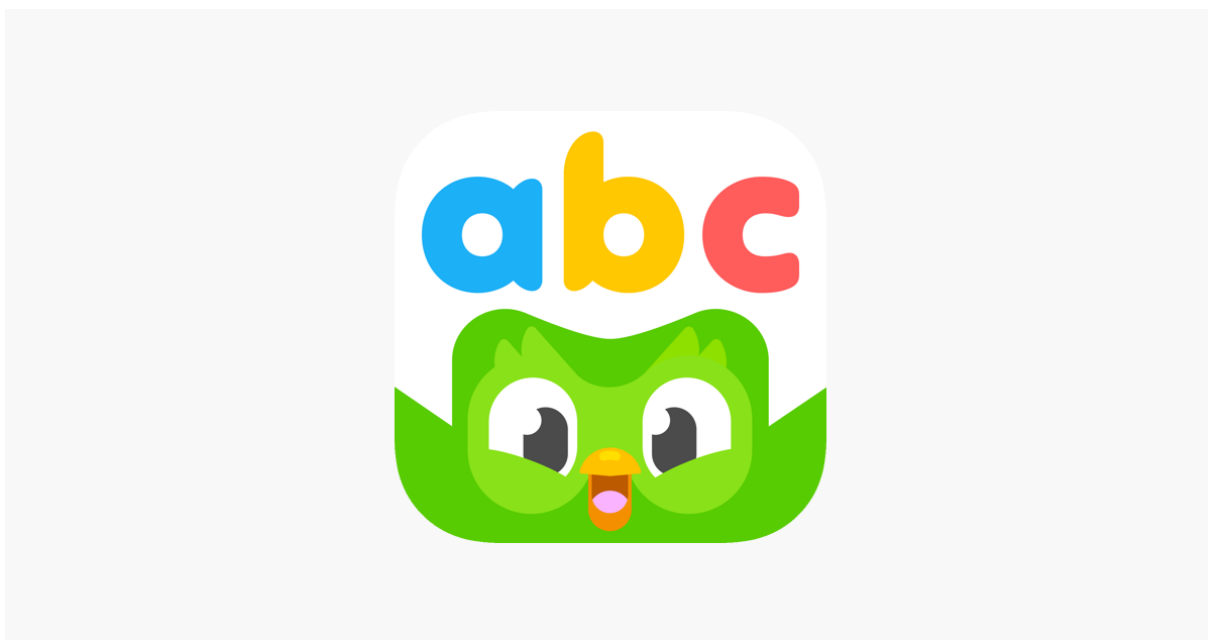


Рисунок 1.4 – Іконка додатку Duolingo ABC

Серед інших додатків, що зосереджені винятково на мовному навчанні, можна відзначити First Words for Kids (рисунок 1.5). Його основна функція – вивчення перших слів іноземною мовою на основі карток із малюнками та озвученням. Додаток підтримує кілька мов, серед яких англійська, іспанська, французька, німецька та інші. Однак, на відміну від

попередніх прикладів, цей застосунок має більш статичну побудову й обмежену інтерактивність, що може знизити рівень залученості дитини. Проте він зручний для пасивного повторення та слухового тренування у простому режимі.



Рисунок 1.5 – Іконка додатку First Words for Kids

До списку конкурентів також можна додати додаток Learn First Words (рисунок 1.6). Він орієнтований на базове знайомство з лексикою у формі тематичних наборів. Особливістю додатку є наявність чіткої транскрипції,

високоякісного аудіо та адаптація для офлайн-режиму. Хоча він не має глибокої системи прогресу або гейміфікації, простота інтерфейсу робить його придатним для дітей, які лише починають знайомство з іноземною мовою.



Рисунок 1.6 – Іконка додатку FunEasyLearn

Таким чином, можна відзначити, що провідні мобільні додатки для мовного навчання дітей націлені на формування початкового словникового запасу, використовуючи прості слова, барвисті зображення та озвучення носієм мови. Усі вони реалізують певний ступінь мультимодальності, однак відрізняються за рівнем персоналізації, адаптивності та інтерактивності. Ці відмінності мають бути враховані при проектуванні нового мобільного додатку, який повинен поєднати кращі практики з урахуванням актуальних дидактичних вимог та вікових особливостей дошкільників.

## 1.2.2 Порівняльний аналіз функціоналу

Порівняльний аналіз функціоналу мобільних додатків для вивчення іноземної мови дошкільнятами дає змогу детально оцінити рівень відповідності кожного рішення основним дидактичним, технічним і ергономічним вимогам. Для такого аналізу були обрані додатки, що спеціалізуються саме на вивченні лексики, а не поєднують її з іншими сферами розвитку, наприклад, математикою або логікою. У цьому підпункті буде здійснено порівняння ключових аспектів, таких як структура навчального контенту, способи подання інформації, наявність озвучення, кількість підтримуваних мов, інтерфейсні особливості та засоби адаптації до потреб дитини.

Один із найпотужніших з точки зору функціональності додатків – це FunEasyLearn. Він має розгалужену систему тематичних лексичних наборів, які поділені на теми (тварини, їжа, транспорт тощо), та підтримує більше ніж 30 мов. Важливою особливістю є можливість озвучування слів носіями мови, що забезпечує фонетичну точність. Окрім того, додаток використовує мультимодальність: кожне слово супроводжується зображенням, текстом, транскрипцією та звуком. Проте його інтерфейс орієнтований на широку вікову аудиторію, що може створювати надмірну навігаційну складність для дітей 2–4 років.

Duolingo ABC, навпаки, розроблений спеціально для молодших користувачів. Його функціонал зосереджено на розвитку навичок читання та розширенні словникового запасу. Подання інформації здійснюється через вправи з дотиком, жестами та аудіоінструкціями. Основним недоліком цього рішення є обмеження щодо кількості мов – додаток підтримує лише англійську мову. Крім того, відсутня можливість налаштування тематики слів, що знижує індивідуалізацію навчального контенту. Тим не менш, високий рівень інтерактивності, адаптивний дизайн та систематичність

подачі матеріалу роблять цей додаток одним із найуспішніших у сфері навчання дітей.

Додаток First Words for Kids, хоча й менш комплексний, забезпечує базову функціональність флеш-карток з підтримкою кількох мов. Його сильна сторона – простота: мінімалістичний інтерфейс, чітка структура тем, швидке перемикання між словами. Кожна картка містить зображення, озвучене слово та, у деяких версіях, текстове написання. Однак відсутність інтерактивності, механізмів зворотного зв'язку та ігрових елементів робить навчання більш пасивним, що може знизити залучення дитини до процесу.

Ще одним прикладом є First Words for Kids. Його функціонал майже ідентичний попередньому, однак інтерфейс дещо менш адаптований до наймолодших користувачів. У додатку використовується стандартна структура: перелік тематик, перелік слів, натискання на кожен картку для прослуховування. Відсутня варіативність режимів, можливість перемішування карток або обмеження часу для повторення, що є недоліками з точки зору дидактичного підходу до повторення та автоматизації лексичних одиниць.

Порівняльна таблиця функціональних характеристик додатків, представлена нижче у таблиці 1.1, дозволяє наочно оцінити сильні та слабкі сторони кожного з них:

Таблиця 1.1 – Порівняння функціональних характеристик додатків

Характеристика	FunEasyLearn	Duolingo ABC	Learn First Words	First Words for Kids
Підтримка мов	30+	1	5-10	5-10
Флеш-карти	є	ні	є	є
Набори карт	є	ні	є	є
Ігрові елементи	ні	є	ні	ні

На основі проведеного порівняння можна зробити висновок, що жоден із наявних додатків не поєднує в собі повний набір функціональних, педагогічних і UX-рішень, необхідних для ефективного, зручного та захопливого навчання дітей 2–7 років. Найбільш збалансованим виглядає Duolingo ABC, але його мовна обмеженість та відсутність тематичного структурування звужують сферу застосування. Це відкриває перспективу для створення нового додатку, який поєднає розширену багатомовну підтримку, чітку тематичну організацію, повну мультимодальність, адаптацію до вікових особливостей та можливості інтерактивної взаємодії, що задовольнятимуть як дидактичні вимоги, так і очікування дітей та їх батьків.

### 1.2.3 Визначення переваг та недоліків існуючих рішень

Аналіз функціоналу найпопулярніших мобільних додатків для вивчення іноземної лексики дошкільнятами дозволив виявити як ефективні підходи, так і поширені обмеження. До переваг існуючих рішень слід віднести підтримку мультимодального навчання, використання озвучень носіями мови, наявність ілюстрованих флеш-карток, гейміфікацію та інтуїтивно зрозумілий інтерфейс. Деякі додатки, зокрема Duolingo ABC, демонструють високу адаптованість до вікових особливостей дітей та систематичність подачі матеріалу. Інші, як-от FunEasyLearn, забезпечують розширену мовну підтримку та поділ лексики за темами, що полегшує орієнтацію в навчальному матеріалі.

Разом з тим, більшість розглянутих додатків мають суттєві недоліки. Зокрема, у багатьох з них відсутній режим перемішування слів або варіативність навчальних сценаріїв, що знижує ефективність повторення. Часто також не передбачено можливості гнучкого вибору мови інтерфейсу та навчання, що обмежує доступність для багатомовних користувачів. У деяких випадках інтерфейс містить надмірну кількість елементів, що

ускладнює взаємодію для наймолодших дітей. Ці недоліки свідчать про потребу у створенні нового додатку, який би враховував найкращі практики та усував виявлені обмеження в існуючих рішеннях.

### 1.3 Вимоги цільової аудиторії до функціональності та інтерфейсу мобільного додатку

Ефективність мобільного додатку для навчання значною мірою визначається не лише його функціональністю, а й ступенем відповідності потребам цільової аудиторії. Особливо це стосується додатків, орієнтованих на дітей дошкільного віку, де кожен елемент інтерфейсу, кожна функція та візуальне рішення мають враховувати вікові, психологічні та когнітивні особливості користувача. У контексті цього дослідження цільовою аудиторією є діти з України, які тимчасово або постійно проживають за кордоном. Тому додаток має не лише сприяти вивченню нових мов, а й виконувати важливу культурну місію – підтримувати знання рідної мови, зберігати мовну ідентичність та забезпечувати емоційний зв'язок із Батьківщиною.

У цьому підрозділі буде розглянуто ключові аспекти, пов'язані з виявленням потреб та очікувань цільової аудиторії. Він зосереджується на визначенні основних характеристик користувачів додатку – віку, контексту проживання, мовного середовища. Також будуть проаналізовані очікування та потреби дорослих – батьків і викладачів, які опосередковано впливають на вибір та використання додатку. Нарешті, у кінці буде розглянуто вимоги до інтерфейсу з урахуванням дитячого сприйняття – кольорова гама, розмір елементів, спрощеність взаємодії та уникнення перевантаження. Такий підхід дозволяє забезпечити цілісне бачення потреб користувача на всіх рівнях і закласти основу для обґрунтованих дизайнерських та функціональних рішень.

### 1.3.1 Виявлення цільової аудиторії

Основною цільовою аудиторією мобільного додатку є діти дошкільного віку з України, які внаслідок воєнних подій були вимушені залишити територію країни та зараз проживають у різних країнах світу. Це переважно діти віком від 2 до 7 років, які перебувають у багатомовному середовищі та часто піддаються ризику втрати рідної мови. У таких умовах важливо створити інструмент, який не лише сприятиме засвоєнню лексики нової для дитини мови, але й одночасно підтримуватиме знання української мови як основи національної ідентичності.

Аудиторія також характеризується високою залежністю від дорослих у питаннях доступу до цифрових технологій, тому функціональність додатку має враховувати як потреби дитини, так і зручність для батьків. Оскільки географія проживання українських біженців охоплює десятки країн, мовна палітра додатку повинна включати українську як основну мову, а також найбільш поширені мови країн перебування – зокрема польську, німецьку, англійську, іспанську, чеську тощо. Таким чином, ідентифікація цільової аудиторії дозволяє сформулювати базові вимоги до структури контенту, мовної підтримки та формату взаємодії користувача з додатком.

### 1.3.2 Потреби батьків та викладачів

Окрім безпосередніх користувачів додатку – дітей дошкільного віку – важливу роль у його використанні відіграють дорослі: батьки, опікуни та викладачі. Саме вони приймають рішення про встановлення додатку, контролюють час користування та оцінюють ефективність навчання. Для них важливими є такі аспекти, як безпека контенту, відсутність реклами, можливість перемикання мов, простота у використанні та наявність навчального ефекту. Також значення має офлайн-доступ до контенту, щоб дитина могла навчатися без постійного підключення до мережі.

Викладачі, логопеди або працівники освітніх центрів, які працюють з дітьми-біженцями, очікують від додатку чіткої структури матеріалу, тематичної організації слів та адаптації контенту до різних рівнів підготовки. Важливо, щоб додаток був гнучким – дозволяв обирати потрібну мову, тему чи тип вправи. Також цінується наявність зворотного зв'язку у формі звукових або візуальних підказок, що допомагають дитині самостійно навчатися. Врахування потреб дорослих користувачів сприяє підвищенню довіри до додатку та забезпечує його регулярне і цілеспрямоване використання.

### 1.3.3 Адаптація інтерфейсу під дитяче сприйняття

Інтерфейс мобільного додатку, орієнтованого на дітей дошкільного віку, має бути максимально простим, інтуїтивним і візуально зрозумілим. Діти цього віку ще не володіють навичками читання або лише починають їх формувати, тому основна взаємодія повинна відбуватися через іконки, зображення, кольори та озвучення. Усі елементи інтерфейсу мають бути великими, добре помітними, з мінімальною кількістю тексту. Кольорова гама повинна бути яскравою, але не надмірно контрастною, щоб не перевантажувати зорове сприйняття.

Особливу увагу слід приділити однорідності структури екранів, передбачуваності дій та уникненню складних навігаційних схем. Важливо, щоб дитина могла з легкістю повернутися до попередньої дії або повторити слово без сторонньої допомоги. Озвучення має автоматично супроводжувати кожну нову картку, а також бути доступним за натисканням. Крім того, в інтерфейсі не повинно бути зайвих елементів або текстових меню, які можуть викликати розгубленість. Простота, емоційна привабливість та стабільність поведінки інтерфейсу є ключовими умовами його ефективного сприйняття дитиною.

## 1.4 Визначення ключових функцій та можливостей

Функціональність мобільного додатку є ключовим чинником, що визначає його ефективність, зручність використання та відповідність потребам користувача. У випадку навчального застосунку для дітей дошкільного віку особливу увагу слід приділити як структурі навчального контенту, так і формату його подання. Важливо не лише забезпечити доступ до лексичних одиниць різними мовами, а й створити таку систему взаємодії, яка буде цікавою, інтуїтивною та пізнавальною для дитини. У цьому розділі буде розглянуто основні функції, які реалізовані або плануються до реалізації в рамках додатку.

У підрозділі описано тематичні лексичні колекції, що забезпечують навчання на основі смислових блоків. Також присвячено цифровим флеш-карткам як основному засобу мультимодального засвоєння матеріалу. Звісно розглядається підтримка багатомовності та механізми аудіовідтворення еталонної вимови. Нарешті, текст висвітлює реалізацію стохастичного режиму – функціональності, що дозволяє вивчати слова у випадковому порядку для кращого закріплення знань. Така структура дозволяє комплексно охопити як дидактичні, так і технічні аспекти реалізації навчального процесу в додатку.

### 1.4.1 Тематичні лексичні колекції для різноманітного мовного навчання

Використання тематичних лексичних колекцій є основою систематизованого вивчення словникового запасу у дітей дошкільного віку. Такий підхід дозволяє структурувати мовний матеріал у логічні блоки, що відповідають реальному життєвому досвіду дитини: «тварини», «фрукти», «транспорт», «одяг», «сім'я», «кольори» тощо. Завдяки цьому дитина не просто вивчає слова, а формує між ними смислові зв'язки, що сприяє

глибшому засвоєнню та перенесенню знань у повсякденне мовлення. Тематичне групування також полегшує навчання для батьків і викладачів, дозволяючи цілеспрямовано обирати необхідний набір для повторення або введення нової лексики.

Кожна колекція у додатку подається у вигляді набору флеш-карток, які супроводжуються зображенням, написанням слова, транскрипцією та озвученням. Завдяки простій навігації між наборами дитина або дорослий може легко переходити від однієї теми до іншої, що підтримує інтерес і запобігає втомі від одноманітності. Крім того, тематичний підхід дозволяє адаптувати навчальний контент до різного віку і рівня підготовки дитини, забезпечуючи поступовий розвиток мовних навичок у багатомовному середовищі.

#### 1.4.2 Цифрові флеш-картки як засіб мультимодального засвоєння лексики

Цифрові флеш-картки є ключовим засобом подачі мовного матеріалу в мобільному додатку, орієнтованому на дітей дошкільного віку. Їх перевага полягає в поєднанні кількох каналів сприйняття: дитина одночасно бачить слово, чує його правильну вимову, а також може співвіднести його із зображенням. Такий підхід забезпечує мультимодальність навчання та активізує когнітивні процеси, що сприяють кращому засвоєнню нової лексики. Принцип простоти й повторення, властивий флеш-карткам, цілком відповідає можливостям сприйняття дітей цього віку.

У цифровому форматі картки подаються в інтерактивному вигляді з можливістю переходу між ними, повторного прослуховування слова, зміни мови або тематики. Окремі елементи можуть змінюватися в залежності від налаштувань: наприклад, транскрипція може бути увімкнена або прихована, якщо користувач ще не вміє читати. Зручність використання карток полягає

також у можливості перемикання між режимами навчання, які чергують ознайомлення, повторення та перевірку знань.

Зображено приклади реалізації інтерфейсу цифрових флеш-карток у мобільному додатку (рисунок 1.7). Картка містить зображення об'єкта, написання слова, транскрипцію та кнопки для голосового озвучення, підтвердження або повтору. Такий підхід дозволяє дитині взаємодіяти з кожним словом у кількох форматах, підкріплюючи сенсорний і мовний досвід. Також на екрані відображається прогрес опрацювання теми, що мотивує до подальшого навчання.

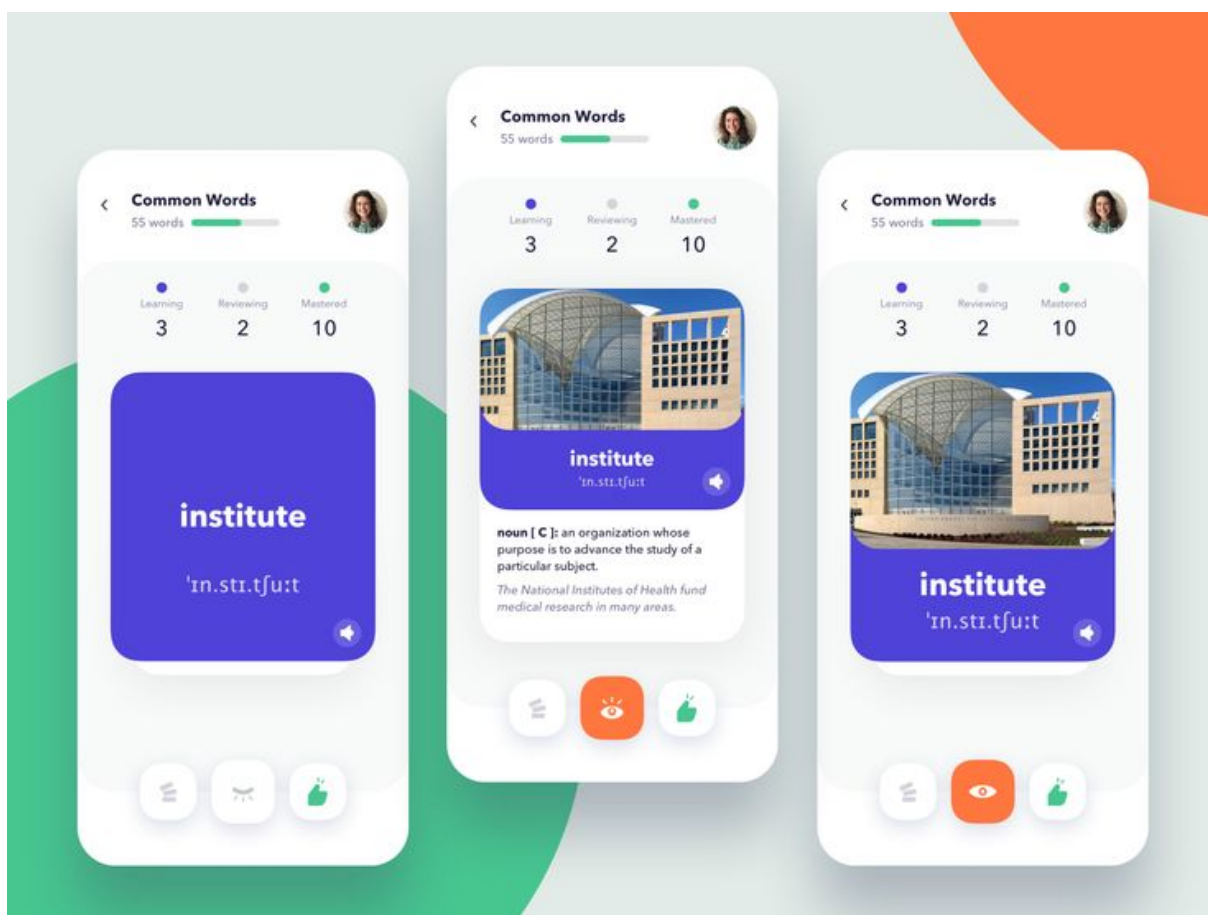


Рисунок 1.7 – Приклад дизайну флеш-картки

Додатковою перевагою цифрових флеш-карток є їхня адаптивність до індивідуального темпу навчання. Дитина може самостійно або з допомогою дорослого зосередитися на конкретній темі або повернутися до складних

слів. Завдяки цьому флеш-картки можуть бути використані як у самотньому навчанні, так і в освітньому середовищі, наприклад, у мовних групах для біженців. Простота структури, зручна візуалізація та можливість багатомовної підтримки роблять цифрові картки універсальним інструментом для початкового мовного навчання.

#### 1.4.3 Підтримка багатомовності та прослуховування еталонної вимови носія

Підтримка багатомовності у мобільному додатку є однією з ключових функцій, яка дозволяє адаптувати навчання до індивідуального мовного середовища дитини.

Оскільки цільова аудиторія охоплює дітей з України, які проживають у різних країнах, важливо надати можливість швидкого перемикання між українською мовою та мовами країни перебування – наприклад, польською, німецькою, англійською, чеською. Кожне слово у додатку супроводжується озвученням носія мови, що забезпечує правильну фонетичну модель для наслідування.

Такий підхід сприяє не лише накопиченню словникового запасу, а й формуванню навичок правильної вимови та розпізнавання звучання на слух. Це особливо важливо в умовах багатомовного середовища, де дитина одночасно чує декілька мов і потребує чітких зразків для наслідування.

#### 1.4.4 Реалізація стохастичного режиму тренування словникового запасу

Стохастичний режим тренування словникового запасу передбачає випадкове відтворення карток без фіксованої послідовності, що підвищує ефективність засвоєння лексики та запобігає ефекту запам'ятовування порядку, а не змісту. Такий режим є особливо корисним на етапі

повторення, коли дитина вже ознайомила з темою і готова до активного відтворення вивчених слів. Випадковість у подачі матеріалу активізує увагу, дозволяє уникнути звикання до структури та створює умови, наближені до реального мовного середовища, де слова зустрічаються в різному контексті.

У додатку передбачено можливість активувати режим «вчити в перемішку» як окрему опцію, що дозволяє урізноманітнити навчальний процес і підтримувати інтерес дитини.

### 1.5 Постановка задачі

У контексті розробки мобільного додатку для вивчення слів іноземною мовою дітьми дошкільного віку постає необхідність чітко сформулювати задачі, які має вирішити даний програмний продукт. Актуальність проекту зумовлена потребою у збереженні мовної ідентичності українських дітей-біженців, які перебувають у багатомовному середовищі за кордоном. Водночас додаток має виконувати подвійне завдання: підтримувати знання української мови як рідної та сприяти адаптації до мовного середовища країни проживання.

Виходячи з цього, основними задачами дослідження є: проаналізувати сучасні тенденції цифрової освіти для дітей дошкільного віку та визначити вимоги до інтерфейсу та функціональності навчального застосунку; провести огляд конкурентних мовних додатків і виявити їхні сильні та слабкі сторони; сформулювати перелік ключових функцій для реалізації ефективної мовної практики; спроектувати структуру додатку з урахуванням мультимодального підходу; реалізувати тематичні набори слів з підтримкою української та інших мов; забезпечити інтуїтивний, безпечний та емоційно привабливий інтерфейс для дошкільнят.

## 2 ПРОЕКТУВАННЯ СИСТЕМИ

### 2.1 Аналіз вимог до мобільного додатку

Етап аналізу вимог є одним з найважливіших у процесі розробки мобільного застосунку, оскільки саме на цьому етапі формується загальне бачення функціональності, інтерфейсу та технічних характеристик майбутньої системи. Ретельно визначені вимоги дозволяють уникнути непорозумінь під час реалізації, скоротити час розробки, зменшити витрати на виправлення помилок та забезпечити відповідність очікуванням користувача. Особливої актуальності набуває це у випадку освітнього застосунку для дітей, де необхідно враховувати вікові, когнітивні та мовні особливості цільової аудиторії.

Аналіз вимог поділяється на дві основні категорії – функціональні та нефункціональні. Функціональні вимоги визначають, що саме має робити застосунок: які дії повинен виконувати користувач, які сценарії взаємодії підтримуються, яка логіка навчання реалізується. Натомість нефункціональні вимоги стосуються загальних характеристик системи, таких як швидкодія, адаптивність, зручність використання, вимоги до пристрою та безпеки зберігання даних. У сукупності ці аспекти утворюють базу для формування чітких цілей розробки та критеріїв прийнятності готового продукту.

В окрему категорію варто винести вимоги до користувацького інтерфейсу. Для мобільного застосунку, орієнтованого на дошкільнят, інтерфейс є не лише засобом управління, а й ключовим елементом навчального середовища. Він повинен бути простим, інтуїтивно зрозумілим, візуально привабливим і не перевантаженим зайвими елементами. Крім того, необхідно враховувати роль дорослих користувачів, які допомагають дитині взаємодіяти із системою – це вимагає додаткових функціональних зон або режимів доступу.

У межах цього підрозділу розглядатимуться загальні підходи до формування вимог до мобільного застосунку, які далі будуть деталізовані в окремих пунктах. Зокрема, буде подано вимоги до інтерфейсу, визначено основні функціональні сценарії, описано очікувану поведінку додатку у процесі навчання, а також вказано вимоги до надійності, масштабованості та мовної адаптації. Такий поділ дозволяє більш чітко організувати процес подальшого проектування системи.

### 2.1.1 Вимоги до користувацького інтерфейсу

Користувацький інтерфейс є одним з основних елементів, що визначають зручність, ефективність та привабливість мобільного застосунку. У випадку цільової аудиторії, яка представлена дітьми дошкільного віку, інтерфейс має бути максимально простим, візуально зрозумілим та адаптованим до рівня розвитку сприйняття. Інтуїтивна взаємодія, велика графіка, яскраві кольори та відсутність зайвих текстових блоків дозволяють дитині швидко зрозуміти принцип роботи застосунку без необхідності у читанні або складних діях.

Особливої уваги потребує поділ інтерфейсу на дві умовні зони – для дитини та для дорослого. У дитячій частині всі елементи повинні бути великими, статичними або з плавною анімацією, а дії – передбачуваними. Необхідно уникати подвійних натискань, прихованих функцій або складної навігації. Водночас доросла зона, наприклад, меню налаштувань, може містити більш складні елементи управління, але бути захищеною додатковим підтвердженням або блокуванням.

Інтерфейс повинен забезпечувати миттєвий зворотний зв'язок: натискання кнопки, перемикання мов або програвання звуку має супроводжуватися візуальними й аудіоіндикаторами. Для дітей важливо не лише бачити зміни, а й чути результат своїх дій. Усі анімації повинні бути

повільними, щоб не перевантажувати когнітивне сприйняття, і мати логічне завершення, наприклад – підтвердження переходу на нову картку.

Вимоги до інтерфейсу також передбачають відповідність сучасним принципам UX-дизайну для мобільних застосунків. Система має адаптуватися під різні розміри екранів, зберігати пропорції візуальних елементів та забезпечувати стабільну взаємодію з дотиком. Крім того, інтерфейс має враховувати мультикультурність середовища, у якому перебуває дитина, – зокрема, через використання нейтральної іконографіки та уникнення культурно обумовлених символів. Усі ці фактори формують вимоги до проектування комфортного, безпечного й ефективного середовища навчання.

### 2.1.2 Функціональні вимоги

Функціональні вимоги визначають набір можливостей, які має реалізовувати мобільний застосунок для досягнення своєї навчальної мети. Основною функцією системи є ознайомлення дитини з базовим лексичним запасом кількома мовами через інтерактивні флеш-картки. Кожна картка повинна містити ілюстрацію, написання слова, транскрипцію та озвучення носієм мови. Користувач повинен мати можливість переходити між картками, обирати теми та перемикає мову навчання.

Застосунок має підтримувати тематичні колекції, які згруповані за змістовними категоріями, зрозумілими дітям дошкільного віку – тварини, фрукти, транспорт, кольори, сім'я тощо. Користувач повинен мати змогу самостійно або за допомогою дорослого обрати тему, переглянути кількість слів у наборі, а також розпочати навчання. Після перегляду всіх слів система має зберігати прогрес, щоби при наступному запуску можна було повернутися до вивченого матеріалу або повторити складні слова.

Особливе місце займає функція перемикає мов. Кожне слово повинно бути доступне українською та однією з іноземних мов,

англійською, польською, німецькою тощо. Перемикання має здійснюватися однією дією, із миттєвим оновленням усіх елементів на екрані. Крім того, повинна бути реалізована функція програвання еталонної вимови натисканням або автоматично при переході до нової картки. Це дозволить дитині сформувати правильні звукові асоціації та підвищити рівень аудіювання.

Додатково система повинна підтримувати режим стохастичного навчання, коли слова відтворюються у випадковому порядку. Такий режим особливо корисний на етапі повторення, оскільки допомагає уникнути запам'ятовування послідовності, а не змісту. Функціональність має також включати мінімальну статистику або позначки опрацьованих слів, що дозволяє батькам відстежувати прогрес. Загалом, функціональні можливості мають бути спрямовані на досягнення простого, ефективного й гнучкого мовного навчання в умовах мобільного середовища.

## 2.2 Опис функціональної моделі

Функціональна модель мобільного застосунку є основою для розуміння логіки взаємодії користувача з системою. Вона описує, які дії може виконати користувач, у якій послідовності ці дії виконуються та які результати очікуються від кожного кроку. У межах цього застосунку визначено кілька основних ролей взаємодії – дитина як основний користувач, а також батьки або вихователь як допоміжні користувачі. Для кожної з цих ролей передбачено власні сценарії взаємодії, які реалізуються через окремі елементи інтерфейсу.

Основним функціональним сценарієм є проходження дитиною навчального блоку на основі флеш-карток. Користувач обирає тему з каталогу доступних колекцій, після чого на екрані з'являється перша картка. На картці зображено предмет, що озвучується при завантаженні. Також користувач може повторно натиснути кнопку для прослуховування або

переглянути транскрипцію. Після вивчення поточного слова дитина натискає на стрілку переходу до наступної картки. Після завершення теми застосунок фіксує прогрес.

Окремим сценарієм є повторення вже вивченого матеріалу. Після проходження теми або кількох тем дитина може активувати режим «вчити в перемішку», який реалізує стохастичне відображення слів. У цьому режимі картки з'являються у випадковому порядку, що дозволяє краще закріпити вивчену лексику. За бажанням батьки можуть обрати одну або кілька тем для повторення, або налаштувати кількість слів, які з'являтимуться під час сесії.

Ще один функціональний сценарій пов'язаний із перемиканням мов. На будь-якому етапі взаємодії користувач або дорослий може змінити мову вивчення. Після цього автоматично оновлюється як написання слова, так і аудіоозвучення. Усі функції працюють аналогічно, але мовний контент подається в новій вибраній мові. Це дозволяє дитині знайомитися з одним і тим самим поняттям у різних мовних контекстах, розвиваючи багатомовну компетентність.

Функціональна модель також передбачає сценарій доступу до налаштувань. Ця зона є обмеженою для дитини й передбачає захист (наприклад, спрощений код або тривале натискання). У налаштуваннях дорослий користувач може змінити мову інтерфейсу, переглянути прогрес, вибрати активні теми або очистити дані. У подальших версіях застосунку може бути реалізовано збереження статистики через хмарне сховище або прив'язку до профілю.

Кожен із наведених сценаріїв має чітко визначений початок і завершення, а також логіку переходу між станами. Таке моделювання дозволяє сформулювати основу для створення діаграм діяльності, схем переходів і модульної архітектури застосунку. На базі функціональної моделі буде реалізовано як візуальні елементи інтерфейсу, так і бізнес-

логіку, що забезпечує послідовність і передбачуваність взаємодії з системою.

### 2.3 Схема використання додатку

У межах побудови загальної логіки використання мобільного застосунку доцільно представити послідовність взаємодії користувача з основними модулями системи. Така послідовність дозволяє чітко визначити етапи користувацької активності та відповідні дії інтерфейсу, починаючи від запуску застосунку й закінчуючи завершенням сеансу навчання. Сценарій взаємодії (рисунок 2.1), що охоплює вибір теми, ознайомлення з картками, перемикання мов та фіксацію прогресу, подано у вигляді діаграми з урахуванням внутрішніх викликів системних компонентів.

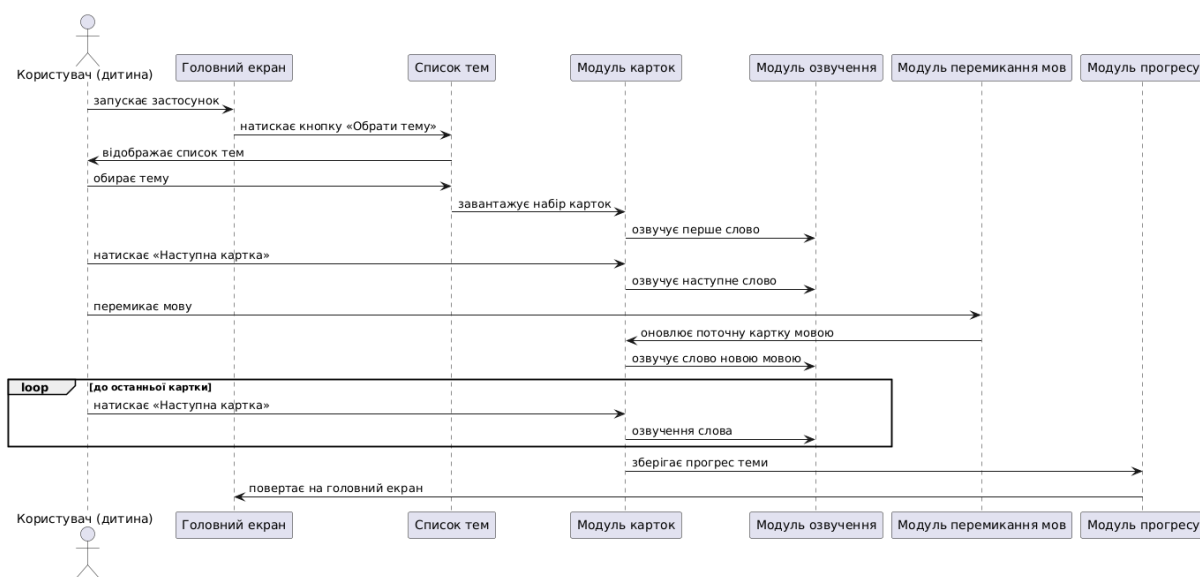


Рисунок 2.1 – Типовий сценарій використання мобільного застосунку

Основні учасники процесу взаємодії включають головний екран, список тем, модуль карток, модуль озвучення, перемикання мов та збереження прогресу. Початковий етап починається з запуску застосунку й обрання теми, після чого завантажується відповідна колекція карток. Кожне

слово озвучується системою одразу після його появи, а користувач може переходити між картками, активуючи нову озвучку.

Користувач має змогу перемикати мову вивчення без необхідності повертатися до початку сесії. Ця дія автоматично оновлює як написання, так і озвучення слова. Ключовим моментом є також стохастичний режим, при якому слова подаються у випадковому порядку, що моделюється у вигляді повторюваного циклу на діаграмі. Після завершення навчальної сесії система фіксує прогрес, і користувач повертається на головний екран.

Завдяки цій схемі можна чітко побачити структуру користувацького сценарію та виявити всі необхідні компоненти, які потрібно реалізувати на рівні архітектури та інтерфейсу. Послідовна побудова етапів взаємодії є критично важливою для забезпечення інтуїтивної логіки та високої ефективності навчального процесу. Вона також слугує основою для майбутньої реалізації тестування та валідації поведінки застосунку.

## 2.4 Архітектура мобільного додатку

Архітектура мобільного застосунку є ключовим елементом під час розробки, оскільки саме вона визначає структуру компонентів, логіку їх взаємодії, спосіб обробки даних та розширюваність системи. На цьому етапі формуються основи для ефективного функціонування всіх модулів – від користувацького інтерфейсу до збереження даних і відтворення аудіо. Обрана архітектура має враховувати особливості роботи на мобільних пристроях, обмеженість ресурсів і потребу у швидкій реакції на дії користувача.

У межах цієї частини буде розглянуто платформу розробки, яку обрано для реалізації застосунку, з урахуванням вимог до підтримки та доступності сучасних інструментів. Особливу увагу буде приділено архітектурному підходу, що дозволяє розділити бізнес-логіку, управління станом і рендеринг інтерфейсу на окремі рівні. Це забезпечує гнучкість,

тестованість і простоту у підтримці застосунку на різних етапах його розвитку.

Також буде виконано загальний опис основних модулів системи, таких як модуль карток, модуль тем, аудіомодуль, менеджер перемикання мов та сховище даних. Для кожного з модулів буде подано логіку взаємодії з іншими частинами системи, а також принципи, за якими вони реалізуються. У результаті буде створено архітектурну основу, що дозволяє ефективно реалізувати функціональну модель, описану на попередніх етапах.

#### 2.4.1 Вибір платформи розробки та обґрунтування

Вибір платформи розробки для мобільного застосунку є одним з визначальних етапів технічного проектування, оскільки він безпосередньо впливає на сумісність, продуктивність, масштабованість та подальшу підтримку системи. З урахуванням того, що цільова аудиторія застосунку – діти-біженці з України, які переважно використовують пристрої Apple, було прийнято рішення про створення застосунку під операційну систему iOS. Це забезпечує максимальну якість користувацького досвіду, високу продуктивність на цільових пристроях і доступ до розвиненої екосистеми інструментів.

Серед інструментів розробки для iOS було свідомо обрано UIKit замість SwiftUI. Попри популярність SwiftUI як сучасного фреймворку, він усе ще має певні обмеження при реалізації складного контролю над UI-елементами, що є критично важливим у застосунках для дітей. UIKit, як зріле та стабільне середовище розробки, дозволяє створити надійний та гнучкий інтерфейс з чіткою відповідністю до специфікацій поведінки, що формуються під потреби користувачів дошкільного віку.

Обрана платформа також дає можливість точно налаштовувати анімації, сенсорні жести та інтеграцію мультимедійного контенту, що має

вирішальне значення для ефективного мультимодального навчання. Наприклад, відтворення еталонної вимови носіями мов або швидке перемикання мовних режимів реалізуються через вбудовані аудіофреймворки iOS з мінімальними затримками, що позитивно впливає на досвід користувача.

Крім технічних переваг, UIKit має потужну підтримку з боку інструментів розробки – Xcode, Instruments, Interface Builder – які дозволяють здійснювати швидку перевірку, оптимізацію та профілювання застосунку на різних пристроях. Це забезпечує відповідність застосунку усім вимогам Apple щодо продуктивності та стабільності, що є передумовою для успішного розміщення в App Store.

Ще одним важливим фактором вибору є підтримка локалізації. Інтерфейс, створений на базі UIKit, легко адаптується до багатомовного середовища через використання стандартних підходів до локалізації рядків, форматів дат, мовної структури меню тощо. Це дозволяє реалізувати підтримку кількох мов інтерфейсу – української, польської, англійської, німецької та інших – без дублювання логіки або ручного перепроєктування.

У результаті обрана платформа забезпечує гнучке, стабільне та масштабоване середовище для створення освітнього застосунку, орієнтованого на дітей. UIKit дозволяє реалізувати складну функціональність, що поєднує навчання, гру та інтерактивність, з точним контролем над кожним елементом UI. Це відкриває шлях до створення якісного цифрового продукту з потенціалом подальшого розвитку.

#### 2.4.2 Обрана структура додатку

Під час проектування структури мобільного застосунку особливу увагу приділено вибору архітектурного шаблону, що забезпечує модульність, повторне використання коду та легкість тестування. У нашому випадку застосовується шаблон MVVM (рисунок 2.2) або інакше кажучи

Model–View–ViewModel, що добре зарекомендував себе в розробці сучасних iOS-застосунків, зокрема для освітніх систем. Його основна ідея полягає в розділенні логіки відображення, бізнес-логіки та взаємодії з даними.

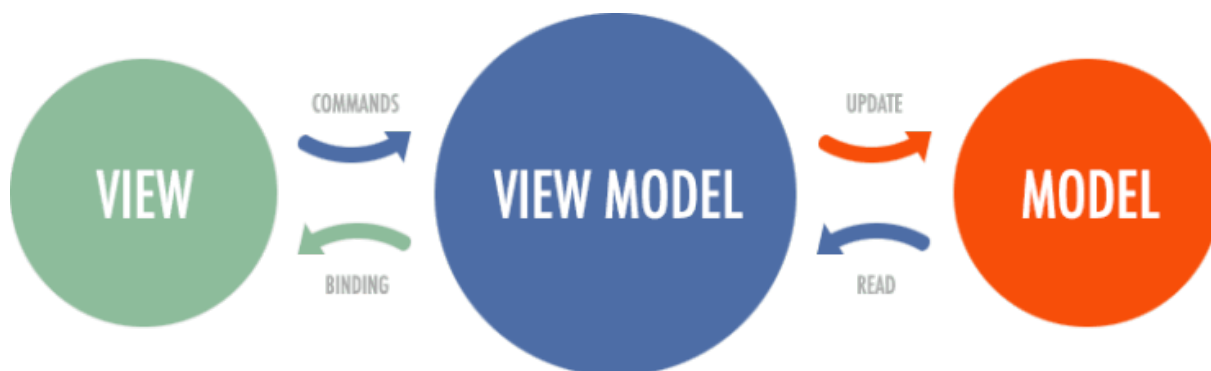


Рисунок 2.2 – Архітектурний шаблон Model–View–ViewModel

Компонент View відповідає за побудову користувацького інтерфейсу, тобто саме те, що бачить і з чим безпосередньо взаємодіє користувач. У межах застосунку це екрани з картками, списком тем, кнопками навігації та іншими елементами. View не має власної логіки обробки даних, а лише відображає ту інформацію, яку отримує від ViewModel, і реагує на дії користувача через підписки або делегати.

ViewModel виконує роль посередника між View та Model. Вона не знає нічого про візуальні компоненти, натомість формує всі дані у зручному для відображення вигляді. Вона також обробляє взаємодії користувача – натискання кнопок, перемикання мов, перехід між картками – і змінює стан програми відповідно до логіки. У цьому компоненті також реалізуються всі обчислення, сортування, перевірки та інші операції, що не пов'язані безпосередньо з відображенням.

Model містить лише дані – словники, список тем, результати користувача, налаштування локалізації. Вона не має знань про те, як ці дані будуть використані. Цей компонент відповідає за взаємодію з базою даних,

завантаження аудіофайлів або ресурсів з пам'яті пристрою. Таким чином, зміна формату зберігання або джерела даних не потребує зміни View або ViewModel.

Завдяки архітектурі MVVM мобільний застосунок стає легко підтримуваним. Наприклад, можна реалізувати новий вигляд для карток без змін у логіці або додати новий тип даних у Model без впливу на відображення. Цей підхід дозволяє масштабувати застосунок, забезпечує незалежне тестування кожного з компонентів та спрощує налагодження. У контексті дитячого освітнього продукту це критично важливо для швидкого оновлення контенту та адаптації до нових мов або сценаріїв навчання.

### 2.4.3 Модулі системи та їх взаємодія

Модульна структура мобільного застосунку забезпечує чіткий розподіл функціональності між окремими компонентами системи, що дозволяє уникати дублювання логіки, полегшує супровід проекту та створює передумови для гнучкого масштабування. У межах цього підходу кожен модуль виконує окреме завдання: від обробки даних до взаємодії з користувачем, зберігаючи єдину архітектурну модель застосунку. Подібне проектування знижує ризик помилок при модифікаціях та дозволяє ефективно розширювати функціональність у майбутньому.

Схематичне представлення логіки взаємодії модулів між собою показано у вигляді діаграми компонентів (рисунок 2.3), що відображає структуру додатку. З головного екрану користувач переходить до модуля тем, де може вибрати одну з лексичних колекцій. Далі застосунок передає управління модулю карток, який є центральним елементом навчального процесу. Саме тут активується робота таких модулів, як озвучення, перемикання мов, збереження прогресу та стохастичного формування порядку слів.

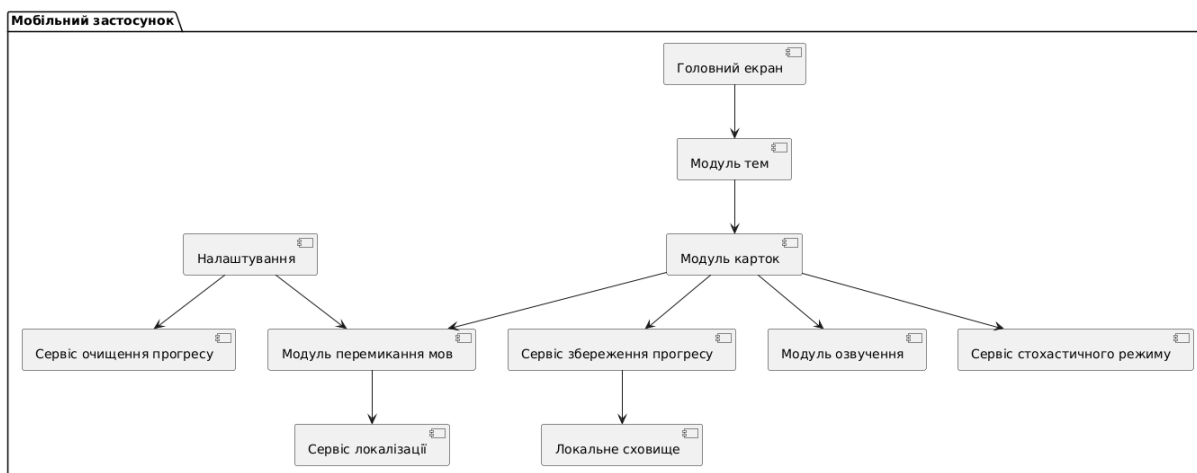


Рисунок 2.3 – Основні модулі мобільного застосунку

Сервіс збереження прогресу забезпечує постійне оновлення даних щодо вивчених слів, їх поточного статусу та кількості повторень. Ці дані зберігаються у локальному сховищі, що дозволяє зберігати навчальний прогрес навіть при перезапуску застосунку. Сервіс стохастичного режиму працює паралельно з модулем карток і відповідає за випадкове перемішування слів, що запобігає ефекту механічного запам'ятовування та стимулює довготривалу пам'ять.

У структурі застосунку також передбачено окремий екран налаштувань, через який користувач може змінити мову інтерфейсу або скинути прогрес вивчення. Цей екран взаємодіє з модулем перемикання мов і сервісом очищення прогресу. Таким чином, уся система побудована навколо централізованого керування потоком даних між модулями, зберігаючи логіку контролю у відповідності до принципів модульності та інкапсуляції.

## 2.5 Проектування бази даних та структури зберігання даних

У мобільному застосунку для вивчення іноземних слів зберігання всіх навчальних та користувацьких даних реалізується за допомогою технології

SwiftData – сучасного фреймворку Apple, призначеного для створення локальних баз даних у додатках під iOS. SwiftData забезпечує зручний декларативний підхід до опису моделей за допомогою атрибутів та автоматично створює схему зберігання. Перевагою цього інструменту є тісна інтеграція з середовищем Swift та можливість роботи з реактивними оновленнями інтерфейсу, що особливо важливо для динамічного освітнього застосунку.

Структура бази даних включає декілька основних сутностей: «Topic», «FlashCard», «UserProgress» та «UserSettings». Кожна тема «Topic» об'єднує набір карток, що належать до однієї лексичної категорії. Картка «FlashCard» містить слово, переклад, транскрипцію, зображення та посилання на аудіофайл з вимовою. Модель «UserProgress» фіксує, які картки вже вивчені, скільки разів вони переглядались, і в якому вони стані. Також окрема сутність «UserSettings» відповідає за збереження мови інтерфейсу, обраної теми та режиму навчання, включно з параметрами стохастичного перемішування.

Завдяки використанню SwiftData забезпечується висока швидкодія при доступі до даних, стабільне зберігання прогресу навіть після перезапуску застосунку та просте оновлення інформації в реальному часі. Усі дані зберігаються локально на пристрої, що дозволяє користувачу продовжувати навчання у будь-який момент без потреби в інтернет-з'єднанні. Такий підхід гарантує автономність і надійність, що є критично важливим для додатків, орієнтованих на використання дітьми в різних умовах проживання.

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

### 3.1 Технологічна основа мобільного додатку

Реалізація мобільного застосунку для вивчення слів іноземною мовою вимагала чіткого визначення технологічної основи, яка забезпечить стабільну роботу, швидкодію, підтримку мультимовності та зручну взаємодію з користувачем. Застосунок розроблявся спеціально для платформи iOS з орієнтацією на використання сучасних інструментів та рішень, які є частиною екосистеми Apple. Під час розробки враховувались вимоги до автономності, простоти оновлення контенту та адаптації під дитячу аудиторію.

Усі компоненти застосунку створені на основі перевірених технологій, що забезпечують високу продуктивність, інтеграцію з інтерфейсом і можливість масштабування. Технологічна основа включає як засоби реалізації інтерфейсу та навігації, так і інструменти збереження даних, опрацювання мультимедійного контенту та локалізації. Основна увага була зосереджена на побудові зрозумілої архітектури, яка дозволяє чітко відокремити логіку навчання, відображення, зберігання та користувацьких налаштувань.

У цьому підрозділі розглядається реалізований стек технологій, що охоплює вибір мови програмування, фреймворків та бібліотек. Також буде наведено опис способу організації модулів, створення моделей даних, логіки переходів між екранами та роботи з мультимовним ресурсом. Завдяки ретельному підбору інструментів вдалося забезпечити збалансовану систему, що поєднує в собі простоту використання та гнучкість структури. Подальші підрозділи розкривають усі аспекти реалізації: від базових рішень до конкретних механізмів, які забезпечують озвучення, зберігання прогресу, локалізацію та виведення контенту на екран.

### 3.1.1 Мова програмування

У межах реалізації мобільного застосунку було використано мову програмування Swift – основну мову для розробки під платформу iOS, рекомендовану компанією Apple. Swift поєднує високу продуктивність, лаконічний синтаксис і сучасні концепції, які дозволяють створювати безпечний, стабільний та легко підтримуваний код. Мова активно розвивається, має велику спільноту, глибоку інтеграцію з Xcode і надає широкі можливості для роботи з графікою, мультимедіа та локальними базами даних.

Під час розробки застосунку Swift продемонстрував себе як зручний інструмент для побудови логіки навігації, структурування моделей даних і організації взаємодії з інтерфейсом користувача. Завдяки використанню сильного статичного типування та механізмів обробки помилок, вдалося зменшити кількість неочікуваних виключень і забезпечити надійну логіку обробки дій користувача. Також важливою перевагою Swift є його підтримка асинхронного виконання, що використовується для завантаження ресурсів без блокування основного потоку.

Swift також забезпечує зручну роботу з локальними структурами даних, що було ключовим під час реалізації логіки навчання, де важливою є реакція на дії користувача в реальному часі. Зокрема, мова дозволила гнучко організувати відображення карток, перемикання між мовами та взаємодію з аудіо через системні фреймворки. Усі моделі були реалізовані як типи з явно визначеними властивостями, що дозволило легко адаптувати їх до роботи з базою даних SwiftData.

Використання Swift у цьому проєкті повністю відповідає сучасним підходам до розробки мобільних застосунків для дітей. Мова надала змогу реалізувати логіку з чітким поділом на модулі, ефективно керувати станом інтерфейсу, а також підтримувати масштабованість системи.

### 3.1.2 Фреймворки та бібліотеки

У процесі реалізації мобільного застосунку було використано низку фреймворків і бібліотек, які забезпечили необхідний функціонал, ефективну обробку мультимедійних даних, роботу з локальним зберіганням, мультимовність і побудову інтуїтивного інтерфейсу. Усі інструменти є частиною офіційної екосистеми Apple, що гарантує їхню стабільну підтримку, сумісність з останніми версіями iOS та відповідність вимогам до продуктивності і безпеки.

Ключовим фреймворком для побудови інтерфейсу став UIKit. Саме через нього реалізовано всі основні екрани застосунку: головне меню, перегляд тем, відображення карток, налаштування та елементи навігації. UIKit надає повний контроль над зовнішнім виглядом і логікою інтерфейсу, що дозволило адаптувати дизайн під потреби дітей дошкільного віку. Використання класичних компонентів iOS разом із кастомними елементами дало змогу реалізувати візуально просту, але функціонально гнучку взаємодію з користувачем.

Для організації зберігання навчальних даних застосовано фреймворк SwiftData. Цей інструмент дозволяє описувати структуру моделі у вигляді сутностей, з якими зручно працювати безпосередньо з коду. За його допомогою реалізовано чотири основні моделі: теми, картки, прогрес користувача та налаштування. SwiftData автоматично керує зв'язками між об'єктами, оновлює інформацію в базі, фільтрує, сортує і дозволяє здійснювати вибірку даних відповідно до потреб інтерфейсу. Це забезпечило високу швидкість відгуку та гнучкість логіки навчання.

Для відтворення аудіо було використано фреймворк AVFoundation. Його інтеграція дала змогу реалізувати функціональність програвання озвучення для кожної картки. Кожен запис зберігається локально, тому доступний без підключення до інтернету. Аудіофайли ініціалізуються при запуску відповідної теми або картки, після чого програватимуться без затримок.

Реакція на дії користувача, як-от повторне натискання або автоматичне озвучення при переході до наступної картки, також реалізована через цей фреймворк.

Фреймворки Foundation і FileManager були використані для обробки текстових ресурсів, локалізації та збереження файлів. Локалізація інтерфейсу реалізована через систему Localizable.strings, що дозволило зручно додавати нові мови і підтримувати багатомовне середовище. Крім того, словникові набори організовані у вигляді локалізованих json-файлів, які легко імпортуються в систему та синхронізуються з обраною мовою інтерфейсу. Це забезпечує масштабованість контенту та простоту оновлення даних без необхідності зміни програмного коду.

Використані фреймворки та бібліотеки працюють узгоджено в межах архітектурної структури застосунку, реалізованої за принципами MVVM. Кожен компонент відповідає за свою зону відповідальності: інтерфейс, логіку, дані, ресурси або медіа. Такий підхід дозволив побудувати стабільну, продуктивну та гнучку систему, яка легко адаптується до нових функцій або змін у контенті. Обрані рішення забезпечили повну автономність роботи застосунку та позитивний досвід для цільової аудиторії.

### 3.2 Реалізація моделей даних у SwiftData

У процесі реалізації мобільного додатку важливим етапом стала побудова ефективної структури даних для локального зберігання. Зважаючи на необхідність повної автономності застосунку, зберігання користувацького прогресу та мовного контенту реалізовано за допомогою SwiftData – фреймворку від Apple для роботи з об'єктною моделлю. Було створено низку моделей, які відображають основні сутності предметної галузі: теми, картки, налаштування користувача, прогрес тощо. Кожна з

моделей має зв'язки з іншими об'єктами, що дозволяє легко здійснювати фільтрацію, сортування та оновлення інформації.

У якості базового прикладу розглянемо модель теми, яка подана у лістингу 3.1, яка є контейнером для пов'язаних словникових одиниць. Вона зберігає унікальний ідентифікатор, назву теми, код мови, що використовується, а також ім'я ілюстративного зображення, яке відображається у списку тем. Окрім того, передбачено відношення до флеш-карток з автоматичним каскадним видаленням, що дозволяє зберігати цілісність структури даних у разі видалення теми. Така структура забезпечує зручну організацію мовного контенту та дозволяє гнучко працювати з навчальними наборами.

Лістинг 3.1 – Програмний код, що описує модель теми Topic для зберігання назви, мови, іконки та пов'язаних флеш-карток у локальній базі даних SwiftData

```
@Model
class Topic {
    @Attribute(.unique) var id: UUID
    var title: String
    var imageName: String
    var languageCode: String
    @Relationship(deleteRule: .cascade) var cards:
[FlashCard]
    init(title: String, imageName: String, languageCode:
String) {
        self.id = UUID()
        self.title = title
        self.imageName = imageName
        self.languageCode = languageCode
        self.cards = []
    }
}
```

Ключовою складовою навчального процесу в додатку є об'єкт флеш-картки, який містить усю необхідну інформацію для засвоєння нової лексичної одиниці. Кожна картка містить слово іноземною мовою, його переклад українською, транскрипцію для правильного читання, зображення для візуального сприйняття та назву відповідного аудіофайлу з вимовою. Вказана структура дозволяє ефективно зберігати дані про картки у `SwiftData`, що і демонструє код, поданий у лістингу 3.2.

Лістинг 3.2 – Програмний код, що описує модель `FlashCard` для зберігання слів, перекладів, транскрипцій, зображень та аудіофайлів у контексті певної теми в `SwiftData`

```
@Model
class FlashCard {
    @Attribute(.unique) var id: UUID
    var word: String
    var translation: String
    var transcription: String
    var imageName: String
    var audioFileName: String
    var languageCode: String
    @Relationship var topic: Topic?
    init(word: String, translation: String, transcription:
String, imageName: String, audioFileName: String, languageCode:
String) {
        self.id = UUID()
        self.word = word
        self.translation = translation
        self.transcription = transcription
        self.imageName = imageName
        self.audioFileName = audioFileName
        self.languageCode = languageCode
    }
}
```

Для відстеження навчального прогресу користувача в системі реалізовано модель `UserProgress`, яка зберігає індивідуальну інформацію про кожну картку. У структурі моделі передбачено збереження унікального ідентифікатора картки, кількості її переглядів, дати останньої взаємодії та булевого індикатора, що вказує на факт засвоєння слова.

Такий підхід дозволяє адаптувати логіку навчання до реальної поведінки користувача, зокрема фільтрувати вивчені та невивчені слова, вести статистику та реалізовувати стохастичне повторення. Модель дозволяє формувати персоналізований освітній досвід, який реагує на рівень активності та прогрес дитини, як це реалізовано в коді, що поданий у лістингу 3.3.

Лістинг 3.3 – Програмний код, що описує модель `UserProgress` для фіксації індивідуального прогресу користувача по кожній картці: кількість переглядів, дата останньої взаємодії та статус засвоєння

```
@Model
class UserProgress {
    @Attribute(.unique) var id: UUID
    var cardID: UUID
    var timesSeen: Int
    var isLearned: Bool
    var lastSeen: Date

    init(cardID: UUID, timesSeen: Int = 0, isLearned: Bool
= false, lastSeen: Date = .now) {
        self.id = UUID()
        self.cardID = cardID
        self.timesSeen = timesSeen
        self.isLearned = isLearned
        self.lastSeen = lastSeen
    }
}
```

Далі було реалізовано модель `UserSettings`, код поданий у лістингу 3.4. Вона містить дані про вибрану мову інтерфейсу, ідентифікатор останньої активної теми, стан режиму перемішування карток, озвучення, а також інформацію про обраний голос для синтезу мови. Окремим параметром враховується перемикач між світлою та темною темами, що сприяє комфортному візуальному сприйняттю у різних умовах.

Лістинг 3.4 – Програмний код, що описує модель `UserSettings`, яка враховує вибір голосу, перемикач темної теми, стан озвучення, активну тему та мову інтерфейсу

```
@Model
class UserSettings {
    @Attribute(.unique) var id: UUID
    var selectedLanguageCode: String
    var activeTopicID: UUID?
    var isShufflingEnabled: Bool
    var isAudioEnabled: Bool
    var selectedVoice: String
    var isDarkModeEnabled: Bool

    init(selectedLanguageCode: String, activeTopicID:
        UUID? = nil, isShufflingEnabled: Bool = true, isAudioEnabled:
        Bool = true, selectedVoice: String = "Female",
        isDarkModeEnabled: Bool = false) {
        self.id = UUID()
        self.selectedLanguageCode = selectedLanguageCode
        self.activeTopicID = activeTopicID
        self.isShufflingEnabled = isShufflingEnabled
        self.isAudioEnabled = isAudioEnabled
        self.selectedVoice = selectedVoice
        self.isDarkModeEnabled = isDarkModeEnabled
    }
}
```

### 3.3 Реалізація функціональності навчання з використанням флеш-карток

Функціональність навчання у мобільному застосунку реалізована через взаємодію користувача з набором флеш-карток, які відображаються на екрані послідовно одна за одною. Основна мета цього механізму – забезпечити інтуїтивну і стабільну логіку перегляду слів у межах обраної теми. Навчання базується на поступовому ознайомленні з лексичними одиницями, кожна з яких має візуальний, текстовий та, за потреби, звуковий супровід. Для цього передбачено окремі функції для завантаження відповідних даних, відображення картки та навігації між ними.

Першим етапом реалізації є отримання списку карток, що належать до конкретної теми. Для цього використовується звернення до локальної бази SwiftData з використанням предиката за ідентифікатором теми. Це дозволяє вибірково витягувати лише ті картки, які безпосередньо пов'язані з обраною темою, що забезпечує ефективну фільтрацію контенту та швидке завантаження необхідних даних. Реалізація цієї логіки представлена у лістингу 3.5.

Лістинг 3.5 – Програмний код, що завантажує всі картки, пов'язані з конкретною темою, використовуючи SwiftData і предикат за ідентифікатором теми

```
func loadFlashCards(for topic: Topic) -> [FlashCard] {
    let descriptor = FetchDescriptor<FlashCard>(
        predicate: #Predicate { $0.topic?.id == topic.id }
    )
    return (try? context.fetch(descriptor)) ?? []
}
```

Після отримання необхідного набору флеш-карток із локального сховища наступним кроком є їх відображення у відповідному інтерфейсі.

Для цього реалізовано функцію, яка передає дані з об'єкта картки до візуальних компонентів екрану. Такий підхід дозволяє централізовано оновлювати зміст картки при кожному перемиканні: користувач бачить нове слово, його переклад, транскрипцію та відповідне зображення, що підтримує асоціативне сприйняття лексики. Конкретна реалізація цієї логіки подана у лістингу 3.6.

Лістинг 3.6 – Програмний код, що оновлює елементи інтерфейсу відповідно до вмісту поточної флеш-картки: слово, переклад, транскрипція та зображення

```
func configureCardView(with card: FlashCard) {  
    wordLabel.text = card.word  
    translationLabel.text = card.translation  
    transcriptionLabel.text = card.transcription  
    imageView.image = UIImage(named: card.imageName)  
}
```

Щоб забезпечити повноцінну взаємодію користувача з набором карток, було реалізовано механізм переходу між ними. Користувач має можливість переглядати слова у межах теми, перемикаючись вперед або назад. Така логіка забезпечує контрольований, поступовий доступ до навчального матеріалу, дозволяючи повернутись до попередньої картки у разі необхідності.

Реалізація перемикання між картками ґрунтується на індексації масиву отриманих карток. При кожному натисканні на відповідну кнопку викликається функція, яка змінює індекс `currentIndex`, а також оновлює вміст інтерфейсу через виклик `configureCardView`. Такий підхід дозволяє легко масштабувати логіку, додаючи, наприклад, анімації переходів або умови завершення теми. Вказаний механізм реалізовано у кодї, поданому в лістингу 3.7.

Лістинг 3.7 – Програмний код, що реалізує логіку перемикання між картками вперед і назад, оновлюючи інтерфейс за допомогою функції `configureCardView`

```
var currentIndex = 0
var flashCards: [FlashCard] = []

func showNextCard() {
    guard currentIndex < flashCards.count - 1 else { return }

    currentIndex += 1
    configureCardView(with: flashCards[currentIndex])
}

func showPreviousCard() {
    guard currentIndex > 0 else { return }
    currentIndex -= 1
    configureCardView(with: flashCards[currentIndex])
}
```

### 3.4 Механізм озвучення слів за допомогою AVFoundation

Для підвищення ефективності засвоєння нових слів у застосунку передбачено можливість прослуховування правильної вимови кожної лексичної одиниці. Такий механізм озвучення дозволяє залучити аудіальний канал сприйняття, що є важливим елементом мультимодального навчання. Озвучення реалізовано на основі локальних аудіофайлів, які асоційовані з кожною картою та підтримують кілька варіантів голосів. Технічно для цього використовується фреймворк AVFoundation, зокрема компонент AVAudioPlayer, що забезпечує гнучке управління відтворенням звукових ресурсів.

Першим етапом роботи з аудіо є його підготовка до відтворення. З цією метою реалізовано функцію, яка формує назву аудіофайлу,

враховуючи обраний користувачем голос, здійснює пошук відповідного ресурсу у файловому пакунку застосунку та ініціалізує AVAudioPlayer. Після успішної підготовки звуковий файл готовий до відтворення без затримок, що забезпечує швидкий відгук системи на дії користувача. Цей процес проілюстровано у лістингу 3.8.

Лістинг 3.8 – Програмний код, що ініціалізує AVAudioPlayer для відповідного слова, враховуючи вибраний користувачем голос

```
var audioPlayer: AVAudioPlayer?
func prepareAudio(for card: FlashCard, voice: String) {
    let fileName = "\(card.audioFileName)_\(voice)"
    guard let url = Bundle.main.url(forResource: fileName,
withExtension: "mp3") else { return }
    audioPlayer = try? AVAudioPlayer(contentsOf: url)
    audioPlayer?.prepareToPlay()
}
```

Озвучення картки має відбуватись одразу при її завантаженні, без необхідності додаткових дій з боку користувача. Для цього було реалізовано функцію, яка поєднує всі необхідні кроки: перевірку наявності активного плеєра, зупинку попереднього відтворення, пошук нового аудіофайлу відповідно до вибраного голосу та подальше його програвання. Такий підхід забезпечує плавний перехід між картками й гарантує, що озвучення буде відповідати актуальному слову, яке бачить користувач на екрані.

Окрему увагу приділено обробці можливих помилок: у разі відсутності аудіофайлу або виникнення проблем під час ініціалізації плеєра система виводить повідомлення в консоль, не перериваючи загальну логіку навчання. Це дозволяє підвищити стабільність застосунку та запобігти аварійному завершенню сесії. Увесь процес повного відтворення звукового супроводу при завантаженні картки представлено у лістингу 3.9.

Лістинг 3.9 – Програмний код, що виконує повне відтворення озвучення слова при завантаженні нової картки: зупиняє попередній звук, готує новий файл і програвє його з урахуванням обраного голосу

```

func playCardAudio(for card: FlashCard, voice: String) {
    if let currentPlayer = audioPlayer,
currentPlayer.isPlaying {
        currentPlayer.stop()
    }

    let fileName = "\(card.audioFileName)_\(voice)"

    guard let url = Bundle.main.url(forResource: fileName,
withExtension: "mp3") else {
        print("Аудіофайл не знайдено: \(fileName).mp3")
        return
    }

    do {
        audioPlayer = try AVAudioPlayer(contentsOf: url)
        audioPlayer?.prepareToPlay()
        audioPlayer?.play()
    } catch {
        print("Не вдалося відтворити аудіофайл:
\(error.localizedDescription)")
    }
}

```

Щоб надати користувачу можливість ще раз прослухати озвучення поточного слова, у застосунку передбачено кнопку повторного програвання. Ця функціональність особливо важлива для закріплення правильної вимови, а також у ситуаціях, коли користувач не встиг почути слово з першого разу. При натисканні на кнопку аудіофайл заново завантажується, готується до відтворення і програвється, незалежно від того, чи відтворювався він щойно чи ні.

У реалізації враховано перевірку на наявність вже активного плеєра, який необхідно зупинити перед запуском нового. Крім того, у функції реалізовано пошук відповідного аудіофайлу в ресурсах застосунку з урахуванням обраного користувачем голосу. У разі виникнення помилки, наприклад через відсутній файл або проблему ініціалізації плеєра, виводиться діагностичне повідомлення, що підвищує стійкість до збоїв. Увесь механізм повторного озвучення показано у лістингу 3.10.

Лістинг 3.10 – Програмний код, що дозволяє користувачу повторно прослухати озвучення поточної картки шляхом натискання на кнопку

```
func repeatButtonTapped(for card: FlashCard, voice: String)
{
    let fileName = "\(card.audioFileName)_\(voice)"

    if let player = audioPlayer, player.isPlaying {
        player.stop()
    }

    guard let url = Bundle.main.url(forResource: fileName,
withExtension: "mp3") else {
        print("Файл не знайдено: \(fileName).mp3")
        return
    }

    do {
        audioPlayer = try AVAudioPlayer(contentsOf: url)
        audioPlayer?.prepareToPlay()
        audioPlayer?.play()
    } catch {
        print("Помилка при відтворенні аудіо:
\(error.localizedDescription)")
    }
}
```

### 3.5 Організація мультимовності та локалізація ресурсу

У межах мобільного застосунку, орієнтованого на багатомовне навчання, важливу роль відіграє не лише локалізація навчального контенту, але й адаптація самого інтерфейсу до обраної мови. Для цього було реалізовано підтримку локалізації з використанням системного механізму Apple на основі файлів `Localizable.strings`. Такий підхід дозволяє виводити текстові елементи у відповідному мовному варіанті, не змінюючи логіки інтерфейсу та не перевантажуючи користувача зайвими налаштуваннями.

У реалізації локалізації інтерфейсу застосунок звертається до функції `NSLocalizedString`, яка за ключем витягує відповідний переклад з мовного файлу. Усі ключові текстові елементи винесені до окремих ресурсів, що дозволяє централізовано оновлювати мовні версії та додавати нові без зміни програмного коду. Функція `updateLocalizedString` викликається при зміні мови або при ініціалізації відповідного екрану, гарантуючи коректне відображення текстів згідно з вибраною локалізацією. Вказана логіка представлена у лістингу 3.11.

Лістинг 3.11 – Програмний код, що забезпечує локалізацію інтерфейсу на основі ключів, збережених у файлах `Localizable.strings`

```
func updateLocalizedString() {
    titleLabel.text = NSLocalizedString("main_title",
    comment: "Заголовок головного екрана")
    startButton.setTitle(NSLocalizedString("start_button",
    comment: "Кнопка початку"), for: .normal)
    settingsButton.setTitle(NSLocalizedString("settings_button",
    comment: "Кнопка налаштувань"), for: .normal)
    languageLabel.text =
    NSLocalizedString("language_label", comment: "Підпис для вибору
    мови")
}
```

На відміну від системної локалізації, яка вимагає перезапуску застосунку для зміни мови, у даному випадку реалізовано повноцінну підтримку динамічної зміни мови інтерфейсу в реальному часі. Для цього створено окремий менеджер локалізації `LocalizationManager`, який зберігає обраний користувачем мовний код у `UserDefaults`, а також завантажує відповідний ресурсний пакет (`.lproj`). Це дозволяє оновлювати текстові компоненти інтерфейсу без необхідності перезавантаження всього застосунку, що особливо важливо для зручності користувача в мультимовному навчальному середовищі.

Ключовою функцією цієї реалізації є `reloadBundle`, яка динамічно змінює поточний `Bundle` відповідно до новообраної мови. Завдяки цьому всі виклики `localizedString` у програмі звертаються до актуального мовного ресурсу. Це дає змогу організувати гнучку локалізацію інтерфейсу, адаптуючи його під різні мовні середовища за вибором користувача. Повна реалізація цієї логіки подана у лістингу 3.12.

Лістинг 3.12 – Програмний код, що реалізує зміну мови інтерфейсу в реальному часі без перезапуску застосунку

```
class LocalizationManager {
    static let shared = LocalizationManager()
    private init() {}

    var currentLanguage: String {
        get {
            UserDefaults.standard.string(forKey:
"AppLanguage") ?? Locale.current.languageCode ?? "en"
        }
        set {
            UserDefaults.standard.set(newValue, forKey:
"AppLanguage")
            reloadBundle(for: newValue)}}

    private(set) var bundle: Bundle = .main
```

## Продовження лістингу 3.12

```

private func reloadBundle(for languageCode: String)
{
    guard let path = Bundle.main.path(forResource:
languageCode, ofType: "lproj"),
        let newBundle = Bundle(path: path) else {
        bundle = .main
        return
    }
    bundle = newBundle
}

func localizedString(for key: String) -> String {
    bundle.localizedString(forKey: key, value: nil,
table: nil)
}
}

```

Оскільки кожна тема в застосунку підтримує кілька мовних варіантів, картки реалізовано як окремі мовні сутності, пов'язані з темою через спільний ідентифікатор. Це означає, що одна й та сама тема може містити картки з текстами різними мовами, кожна з яких має власне слово, переклад, транскрипцію, зображення та аудіо. Для забезпечення коректної вибірки реалізовано функцію фільтрації карток за мовним кодом, що дозволяє завантажити лише ті записи, які відповідають вибраній мові навчання.

Коли користувач змінює мову під час навчання, система автоматично оновлює параметр `selectedLanguageCode` у налаштуваннях, виконує повторне завантаження карток теми та відображає першу з них. Такий підхід забезпечує безперервність навчального процесу при зміні мовного середовища та дозволяє миттєво перемикатись між мовами без втрати контексту. Ця логіка реалізована у лістингу 3.13.

Лістинг 3.13 – Програмний код, що реалізує логіку зберігання карток з окремими мовними версіями, їх фільтрацію при відображенні та перемикання мови в режимі реального часу

```
func getCards(for topic: Topic, languageCode: String) ->
[FlashCard] {
    let descriptor = FetchDescriptor<FlashCard>(
        predicate: #Predicate {
            $0.topic?.id == topic.id && $0.languageCode ==
languageCode
        }
    )
    return (try? context.fetch(descriptor)) ?? []
}

func didChangeLanguage(to code: String) {
    userSettings.selectedLanguageCode = code
    try? context.save()
    flashCards = getCards(for: currentTopic, languageCode:
code)
    currentIndex = 0
    configureCardView(with: flashCards[currentIndex])
}
```

### 3.6 Локальне зберігання прогресу та конфігурацій користувача

Для забезпечення персоналізованого навчального досвіду мобільний застосунок реалізує механізм локального зберігання даних про прогрес користувача та його індивідуальні налаштування. Вся інформація зберігається у форматі моделей SwiftData, що дозволяє ефективно керувати даними, навіть без підключення до мережі. Прогрес зберігається на рівні кожної окремої картки, а налаштування – на рівні користувача. Такий підхід забезпечує гнучкість, автономність та можливість реалізації адаптивної логіки навчання, орієнтованої на реальні дії користувача.

Кожного разу, коли користувач переглядає нову картку, система перевіряє, чи вже є запис у таблиці прогресу. Якщо запис існує, відбувається інкремент лічильника переглядів та оновлення дати останньої активності. Якщо ж картка переглядається вперше, створюється новий запис. Після трьох переглядів слово автоматично позначається як вивчене. Така логіка дозволяє без участі користувача фіксувати поступ у навчанні, що реалізовано у лістингу 3.14.

Лістинг 3.14 – Програмний код, що при кожному перегляді картки автоматично оновлює кількість переглядів, дату останнього перегляду, а також відмічає слово як вивчене після третього перегляду

```
func updateProgress(for card: FlashCard) {
    let request = FetchDescriptor<UserProgress>(
        predicate: #Predicate { $0.cardID == card.id }
    )

    if var progress = try? context.fetch(request).first {
        progress.timesSeen += 1
        progress.lastSeen = .now
        if progress.timesSeen >= 3 {
            progress.isLearned = true
        }
    } else
    {
        let newProgress = UserProgress(cardID: card.id,
timesSeen: 1, isLearned: false, lastSeen: .now)
        context.insert(newProgress)
    }

    try? context.save()
}
```

Окрім загальних параметрів, таких як мова інтерфейсу, активна тема чи стан озвучення, модель `UserSettings` була розширена полями, що дозволяють зберігати статистику щоденної навчальної активності користувача. Це дало змогу реалізувати функціональність, пов'язану з обліком вивчених слів за день, підрахунком кількості днів безперервного навчання (`streak`), а також збереженням інформації про те, у які дні тижня користувач працював із застосунком. Таке розширення моделі суттєво підвищує інформативність і адаптивність освітнього процесу.

Наявність полів `lastActiveDate`, `currentStreak`, `wordsLearnedToday` та `weekdayActivity` дає змогу не лише відображати статистику на екрані, а й формувати логіку її оновлення без потреби в зовнішніх сервісах чи аналітиці. У майбутньому це може бути використано для побудови системи мотиваційних нагород, візуалізації навчального прогресу або адаптації складності контенту до поведінки користувача. Структура оновленої моделі `UserSettings` подана у лістингу 3.15.

Лістинг 3.15 – Програмний код, що розширює модель `UserSettings`, додаючи змінні для зберігання дати останньої активності, тривалості серії днів навчання, кількості вивчених слів за день та міток активності

```
@Model
class UserSettings {
    @Attribute(.unique) var id: UUID
    var selectedLanguageCode: String
    var activeTopicID: UUID?
    var isShufflingEnabled: Bool
    var isAudioEnabled: Bool
    var selectedVoice: String
    var isDarkModeEnabled: Bool
    var lastActiveDate: Date
    var currentStreak: Int
    var wordsLearnedToday: Int
    var weekdayActivity: [String: Bool]
```

## Продовження лістингу 3.15

```
    init(  
        selectedLanguageCode: String,  
        activeTopicID: UUID? = nil,  
        isShufflingEnabled: Bool = true,  
        isAudioEnabled: Bool = true,  
        selectedVoice: String = "Female",  
        isDarkModeEnabled: Bool = false  
        self.id = UUID()  
        self.selectedLanguageCode = selectedLanguageCode  
        self.activeTopicID = activeTopicID  
        self.isShufflingEnabled = isShufflingEnabled  
        self.isAudioEnabled = isAudioEnabled  
        self.selectedVoice = selectedVoice  
        self.isDarkModeEnabled = isDarkModeEnabled  
        self.lastActiveDate = .distantPast  
        self.currentStreak = 0  
        self.wordsLearnedToday = 0  
        self.weekdayActivity = [:]  
    }  
}
```

Оновлення щоденної статистики навчання відбувається автоматично при кожному перегляді нової картки. У реалізованій логіці спочатку перевіряється, чи відповідає поточна дата останній зафіксованій активності. Якщо користувач відкрив застосунок уперше за день, система визначає, чи був учора також активний день, і залежно від цього оновлює лічильник streak. Крім того, лічильник слів, вивчених за поточну дату, обнуляється, після чого збільшується на одиницю для нової картки. Водночас оновлюється дата останньої активності та мітка відповідного дня тижня. Цей механізм повністю автоматизує ведення навчальної статистики та реалізується через код, поданий у лістингу 3.16.

## Лістинг 3.16 – Програмний код, що новлює щоденну статистику

## навчання

```

func updateDailyStatistics(for card: FlashCard) {
    let calendar = Calendar.current
    let today = calendar.startOfDay(for: Date())
    let lastActivity = calendar.startOfDay(for:
userSettings.lastActiveDate)
    if today != lastActivity {
        if calendar.date(byAdding: .day, value: -1, to:
today) == lastActivity {
            userSettings.currentStreak += 1
        } else {
            userSettings.currentStreak = 1
        }
        userSettings.wordsLearnedToday = 0
    }
    userSettings.wordsLearnedToday += 1
    userSettings.lastActiveDate = today
    let weekdaySymbol =
calendar.shortWeekdaySymbols[calendar.component(.weekday, from:
today) - 1]
    userSettings.weekdayActivity[weekdaySymbol] = true

    try? context.save()
}

```

## 3.7 Логіка стохастичного режиму повторення

Для підтримки гнучкого стилю навчання у застосунку реалізовано стохастичний режим повторення, який дозволяє подавати флеш-картки у випадковому порядку. Такий режим активується через відповідний перемикач у налаштуваннях користувача та є особливо корисним на етапах повторення вивченого матеріалу. На відміну від фіксованої послідовності, перемішування запобігає механічному запам'ятовуванню порядку слів і

сприяє глибшому засвоєнню лексики. Стохастичне подання карток також стимулює концентрацію уваги та знижує ефект адаптації до шаблонів.

Реалізація стохастичного режиму здійснюється на етапі завантаження карток теми. Після вибірки карток за відповідною мовою та ідентифікатором теми перевіряється значення прапорця `isShufflingEnabled`. Якщо він активний, масив карток додатково перемішується за допомогою функції `shuffle`. Таким чином, підготовлений список карток динамічно змінюється, забезпечуючи кожного разу нову послідовність для користувача. Ця логіка подана у лістингу 3.17.

Лістинг 3.17 – Програмний код, що завантажує картки заданої теми та мови, і, залежно від прапорця `isShufflingEnabled` у налаштуваннях користувача, перемішує їх у випадковому порядку

```
func loadCards(for topic: Topic, languageCode: String) ->
[FlashCard]
{
    let descriptor = FetchDescriptor<FlashCard>(
        predicate: #Predicate {
            $0.topic?.id == topic.id && $0.languageCode ==
languageCode
        }
    )

    var cards = (try? context.fetch(descriptor)) ?? []

    if userSettings.isShufflingEnabled {
        cards.shuffle()
    }

    return cards
}
```

Для підвищення ефективності стохастичного режиму було додатково реалізовано пріоритезацію невивчених карток. Така логіка дозволяє показувати спочатку ті слова, які ще не були достатньо засвоєні, зберігаючи при цьому випадковий порядок їх подання. На основі збереженого прогресу кожної картки формується словник відповідностей між ідентифікатором картки та її статусом засвоєння. Далі масив карток сортується таким чином, щоб невивчені мали вищий пріоритет, а картки з однаковим статусом перемішувались випадковим чином. Це дозволяє адаптувати навчання до рівня користувача, не втрачаючи елементу випадковості. Детальна реалізація цього алгоритму представлена у лістингу 3.18.

Лістинг 3.18 – Програмний код, що сортує картки теми так, щоб невивчені слова з'являлись раніше, навіть у випадковому порядку.

```
func loadPrioritizedCards(for topic: Topic, languageCode:
String) -> [FlashCard] {
    let descriptor = FetchDescriptor<FlashCard>(
        predicate: #Predicate {
            $0.topic?.id == topic.id && $0.languageCode ==
languageCode
        }
    )

    let allCards = (try? context.fetch(descriptor)) ?? []

    let                progressDescriptor                =
FetchDescriptor<UserProgress>()
    let                allProgress                =                (try?
context.fetch(progressDescriptor)) ?? []

    let learnedMap = Dictionary(uniqueKeysWithValues:
allProgress.map { ($0.cardID, $0.isLearned) })

    let sortedCards = allCards.sorted {
```

### Продовження лістингу 3.18

```

    let firstLearned = learnedMap[$0.id] ?? false
    let secondLearned = learnedMap[$1.id] ?? false

    if firstLearned == secondLearned {
        return Bool.random()
    } else {
        return firstLearned == false
    }
}

return userSettings.isShufflingEnabled ? sortedCards :
allCards
}

```

Таким чином, логіка стохастичного режиму повторення у мобільному застосунку поєднує просту реалізацію перемішування з адаптивною стратегією відбору карток. Такий підхід дозволяє ефективно тренувати словниковий запас, не повторюючи вивчені слова надмірно часто, водночас зберігаючи достатню варіативність послідовності. Завдяки цьому користувач отримує динамічний і гнучкий досвід навчання, який реагує на його реальний прогрес.

### 3.8 Основні екрани користувацького інтерфейсу

Візуальна частина мобільного додатку є ключовим чинником, що забезпечує комфортне навчання та інтуїтивну навігацію для користувача. Оскільки основна аудиторія – це діти молодшого віку, для кожного екрану було обрано максимально просту й доступну структуру. Інтерфейс побудований за принципом прямолінійної взаємодії, де кожен елемент виконує конкретну функцію і не перевантажує користувача зайвою

інформацією. Колірна гама, типографіка та іконографіка підбирались з урахуванням вікових особливостей цільової аудиторії.

Для реалізації основного сценарію взаємодії передбачено декілька ключових екранів, які охоплюють увесь навчальний цикл: вибір теми, робота з флеш-картками та доступ до налаштувань. Вони поєднують у собі як навчальний, так і адміністративний функціонал, дозволяючи користувачеві не лише проходити тренування, а й керувати параметрами навчання. Особлива увага приділяється мінімізації дій, необхідних для досягнення цілі, що дозволяє зосередитись саме на навчальному процесі.

У межах цього підрозділу буде наведено приклади основних екранів інтерфейсу з коротким описом їхнього функціонального призначення. Усі елементи організовані за принципом тематичних блоків, кожен з яких відповідає окремому етапу взаємодії користувача з додатком. Такий підхід дозволяє структурувати інтерфейс у логічні частини та полегшує аналіз реалізованої візуальної логіки.

### 3.8.1 Головний екран

Головний екран мобільного додатку відіграє роль центрального навігаційного хабу, з якого користувач починає свою взаємодію із системою. У верхній частині розміщено вітання з ім'ям користувача, що створює ефект персоналізації і підтримує емоційний контакт із дитиною. Нижче знаходиться основна зона взаємодії – блок запуску навчання, який дозволяє обрати одну з тем або перейти до змішаного режиму навчання, що підвищує гнучкість роботи з лексичним матеріалом.

Особливу увагу приділено мотиваційним елементам. Візуалізація статистики навчання дозволяє дитині слідкувати за кількістю вивчених слів та серією активних днів. Такий підхід заохочує до регулярного повторення, формує позитивну звичку та підтримує інтерес до навчання. Внизу передбачено фільтрацію тем за мовами, що дає змогу легко перемикатися

між мовними середовищами. Усі теми представлені у вигляді плиток із кольоровими іконками, які полегшують розпізнавання і роблять інтерфейс більш дружнім до дітей.

Далі наведено зображення, яке демонструє описану структуру головного екрана мобільного додатку (рисунок 3.1), включаючи блок вітання, статистику прогресу, елементи запуску навчання та добірку останніх тем. Такий візуальний підхід дозволяє підтримувати зосередженість користувача, спрощує навігацію між функціональними модулями та забезпечує швидкий доступ до основних можливостей застосунку.



Рисунок 3.1 – Головний екран мобільного додатку

На головному екрані реалізовано зручний механізм перемикання між доступними мовами навчання. Користувач має змогу одним натисканням

обрати мовне середовище, в якому він бажає переглядати тематичні набори та працювати з лексикою. Такий підхід не лише підвищує гнучкість використання, а й сприяє створенню інклюзивного середовища для дітей-біженців, які можуть вивчати нову мову, не втрачаючи контакт з рідною. Перелік підтримуваних мов, зокрема української, англійської, польської та німецької, наведено у вигляді горизонтального списку кнопок (рисунок 3.2), що дозволяє миттєво оновити відображення відповідно до обраної мови.



Рисунок 3.2 – Перелік мов, які підтримує додаток

Ключовою частиною навчального процесу у мобільному додатку є тематичні набори, які охоплюють базову лексику, актуальну для дітей дошкільного віку. Теми підібрані таким чином, щоб охопити різні аспекти повсякденного життя дитини – від тварин і їжі до шкільних предметів, емоцій, кольорів та транспортних засобів. Це дозволяє формувати збалансований словниковий запас.

Кожна картка теми відображає кількість вивчених слів із загального числа, що забезпечує візуальний контроль за прогресом. Наприклад, у темі Colors користувач опанував 24 слова з 25, тоді як у темі Food поки що не вивчено жодного. Такий підхід стимулює інтерес до подальшого навчання та створює ефект досягнення.

Варто зазначити, що усі теми адаптовані до вибраної мови інтерфейсу, що забезпечує одномовний простір під час проходження конкретного набору. Відповідно, якщо активною мовою встановлено українську, теми відображають слова та лічильники саме цією мовою. Це видно на екрані з темами (рисунок 3.3).

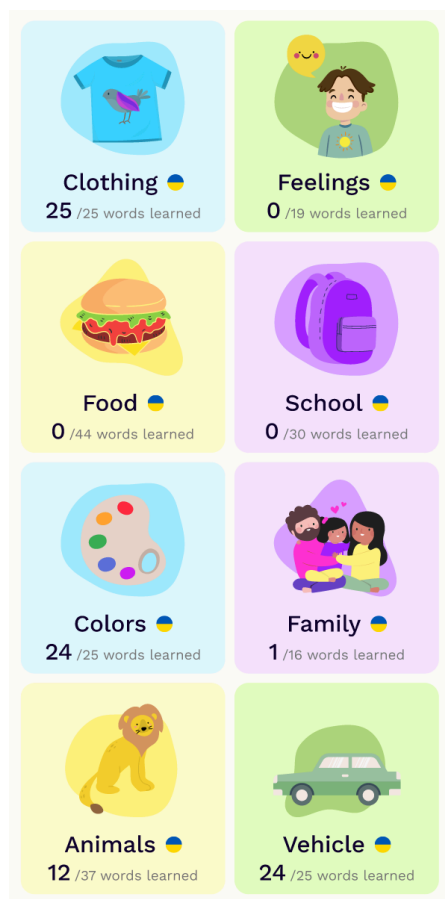


Рисунок 3.3 – Перелік тем, які має додаток

Інтерфейс головного екрана мобільного додатку включає дві основні кнопки для початку навчального процесу – «Select topic» та «Mixed Mode», що дозволяють користувачу обрати відповідний режим вивчення слів. Ці елементи розміщено на помітному місці у верхній частині екрана, що спрощує навігацію та мінімізує кількість дій для старту. Кожна кнопка оформлена контрастним кольором, а також має піктограму, яка підсилює візуальне сприйняття функціоналу (рисунок 3.4).

Кнопка «Select topic» відкриває екран із доступними темами, що дозволяє зосередитись на вивченні конкретного лексичного набору. У свою чергу, натискання «Mixed Mode» активує стохастичний режим, у якому картки з різних тем відображаються в довільному порядку. Такий режим особливо корисний для закріплення вже вивченого матеріалу.

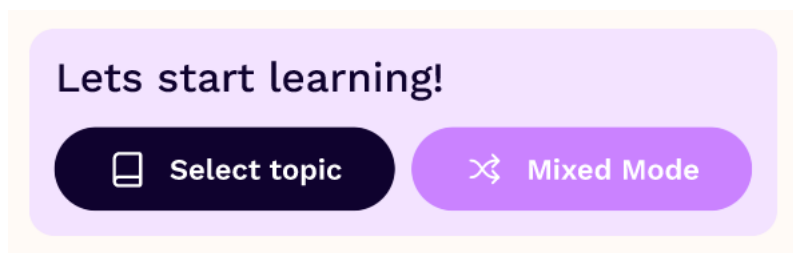


Рисунок 3.4 – Головні кнопки мобільного додатку

Панель щоденної активності та прогресу забезпечує користувача візуальним зворотним зв'язком щодо регулярності навчання та кількості вивчених слів у поточний день. Центральним елементом є індикатор серії днів (*learning streak*), що демонструє кількість днів безперервної активності та мотивує користувача зберігати навчальний темп. Цифровий лічильник поєднано з іконкою полум'я, яка традиційно асоціюється з активністю й досягненням цілей (рисунок 3.5).

Праворуч від лічильника розташовано календар тижня з позначками про активність за кожен день. Кольорові індикатори сигналізують, у які дні користувач вивчав нові слова. Така візуалізація не лише формує звичку до щоденного навчання, а й надає користувачу змогу відслідковувати свої досягнення у розрізі тижня, оцінюючи власний прогрес у зручному й зрозумілому форматі.

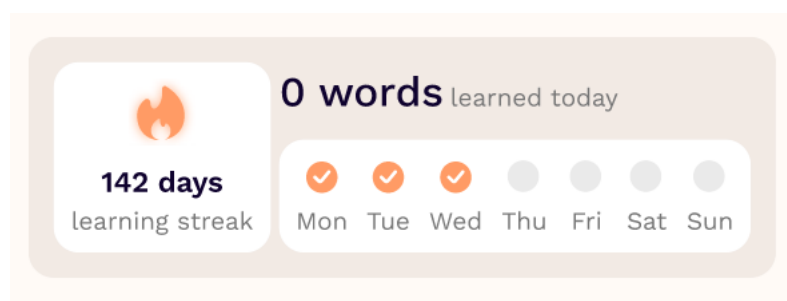


Рисунок 3.5 – Панель щоденної активності та прогресу

У мобільному додатку реалізовано інтуїтивно зрозуміле спливаюче меню для вибору теми навчання, що дозволяє швидко переходити до необхідного набору слів (рисунок 3.6). Кожна картка теми містить назву, іконку та лічильник прогресу, який показує співвідношення між уже вивченими словами та загальною кількістю у темі. Це не лише спрощує орієнтацію у навчальному контенті, а й мотивує користувача продовжувати роботу над ще не завершеними темами. Застосоване кольорове кодування та ілюстрації покращують сприйняття, особливо для цільової аудиторії з числа дітей дошкільного віку.

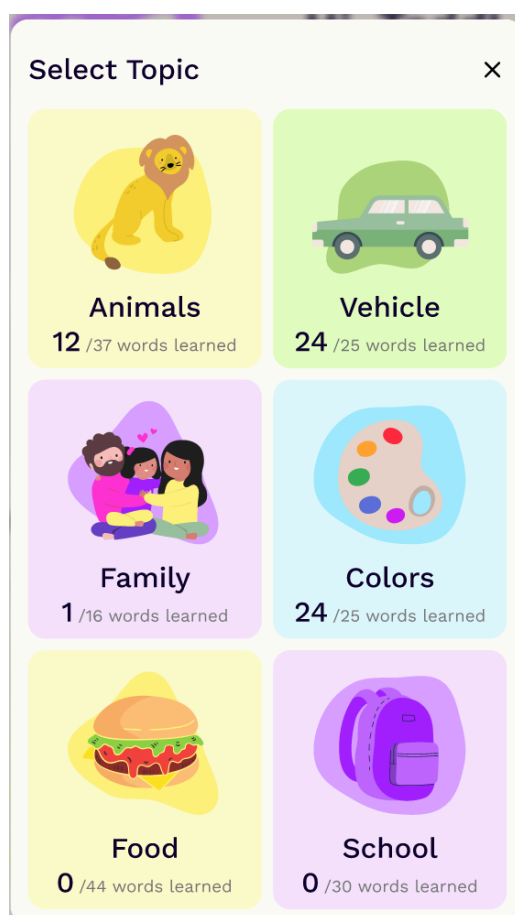


Рисунок 3.6 – Вибір теми для навчання у спливаючому меню

У застосунку реалізовано зручний механізм вибору мови навчання, що представлений у вигляді спливаючого меню з чотирма варіантами: українська, англійська, польська та німецька (рисунок 3.7). Перемикання

мови здійснюється динамічно, що дозволяє користувачеві миттєво бачити відповідні мовні картки без потреби перезавантаження інтерфейсу. Такий підхід забезпечує адаптивність застосунку до індивідуальних потреб користувача, дозволяючи одночасно підтримувати рідну мову та вивчати іноземну.



Рисунок 3.7 – Вибір мови навчання у спливаючому меню

### 3.8.2 Екран флеш-картки

Екран флеш-картки є центральним елементом у процесі навчання, оскільки саме тут дитина безпосередньо взаємодіє зі словниковим матеріалом. В інтерфейсі цього екрану відображається зображення, що ілюструє значення слова, його написання у вибраній мові та транскрипція. Така структура сприяє асоціативному запам'ятовуванню лексики, поєднуючи візуальну, текстову та аудіальну інформацію.

На екрані також розміщено елементи навігації, які дозволяють переходити між картками у межах вибраної теми. Знизу розміщені перемикачі мов, які дають змогу миттєво змінити мовне середовище відображення без повернення на головний екран. Це особливо зручно для дітей-білінгвів або тих, хто вивчає декілько мов одночасно (рисунок 3.8).

Окрім цього, важливу роль виконує кнопка програвання звуку, що активує озвучення слова. Озвучення базується на попередньо збережених

аудіофайлах із голосами, адаптованими під дитячу аудиторію. Такий функціонал не лише підсилює ефект занурення, але й формує правильну вимову, що є критично важливим у ранньому віці.



Рисунок 3.8 – Екран флеш-картки

Однією з ключових функцій мобільного додатку є підтримка багатомовного режиму навчання. На прикладі зображено флеш-картка, що подає слово німецькою мовою (рисунок 3.9), доповнене транскрипцією. Така реалізація дозволяє користувачу не лише ознайомитися з новими словами, але й закріпити правильну вимову завдяки інтеграції аудіофайлів.

Флеш-картка забезпечує інтуїтивне візуальне сприйняття: центральне зображення предмета або явища, назва німецькою мовою та її фонетична транскрипція. Окрему роль відіграє нижній блок інтерфейсу, де можна змінити мову навчання одним натисканням, що активує відповідну версію

картки з тією самою ілюстрацією. Це особливо корисно для порівняння та асоціативного закріплення матеріалу.

Кнопка відтворення звуку, розташована під словом, відповідає за запуск аудіофайлу з правильною вимовою відповідного слова. Таким чином, дитина не тільки читає та бачить нове слово, але й чує його вимову у відповідному мовному середовищі, що створює повноцінну мультимодальну модель засвоєння лексики.



Рисунок 3.9 – Флеш-картка з лексикою німецькою мовою

Далі можна побачити реалізацію англословної флеш-картки, де відображено зображення мотоцикла, його назву англійською мовою та транскрипцію (рисунок 3.10). Такий підхід дозволяє забезпечити послідовне засвоєння лексики, поєднуючи зорове, мовне та слухове сприйняття. Картка

містить всю необхідну інформацію для вивчення нового слова у конкретному контексті.

У нижній частині екрану реалізовано мовний перемикач, де активна англійська мова, що дозволяє змінити мовну локалізацію флеш-картки у реальному часі. Це забезпечує користувачу можливість одночасного порівняння варіантів написання та звучання одного й того ж поняття різними мовами, що суттєво підвищує ефективність навчання.



Рисунок 3.10 – Флеш-картка з лексикою англійською мовою

Флеш-картка, яка демонструє слово польською мовою, органічно поєднує текстову, візуальну та аудіальну інформацію для забезпечення повноцінного сприйняття нового слова (рисунок 3.11). У прикладі з лексемою «Małpa» подано ілюстрацію тварини, транскрипцію слова міжнародною фонетичною абеткою та кнопку для прослуховування

звукової доріжки. Такий підхід дозволяє не лише зорово асоціювати значення, але й закріплювати правильну вимову.

Важливим елементом є візуальна навігація мовного середовища, що знаходиться внизу екрана. Користувач має змогу швидко перемикатись між українською, англійською, польською та німецькою мовами, змінюючи відповідно й вміст флеш-картки. Таким чином реалізується багатомовна підтримка, яка надає змогу вивчати один і той самий контент у різних мовних контекстах.



Рисунок 3.11 – Флеш-картка з лексикою польською мовою

Екран флеш-картки є центральним елементом інтерфейсу мобільного додатку, що забезпечує безпосереднє занурення користувача у процес навчання. Завдяки поєднанню ілюстрацій, перекладу, транскрипції, озвучення та навігаційних елементів, реалізовано зручний і інтуїтивний

механізм засвоєння лексики. Гнучка підтримка мов, адаптивна подача інформації та інтеграція зі статистикою прогресу роблять даний екран ключовим інструментом для ефективного запам'ятовування нових слів.

### 3.8.3 Екран налаштувань

На екрані налаштувань у світлій темі користувач отримує доступ до ключових параметрів персоналізації мобільного застосунку. Цей екран реалізований таким чином, щоб забезпечити максимальну зручність при зміні індивідуальних вподобань. Його компоненти згруповані у зрозумілу вертикальну структуру, де кожен параметр супроводжується відповідною іконкою, назвою опції та перемикачем або списком.

Першим налаштуванням є керування звуковим супроводом. Користувач може за допомогою перемикача увімкнути або вимкнути озвучення флеш-карток. Ця опція особливо корисна для тих, хто бажає зосередитися лише на візуальному або текстовому сприйнятті інформації, або навпаки – активувати аудіоканал для покращення вимови.

Наступним пунктом є вибір голосу для озвучення слів. Доступна можливість обрати між варіантами, наприклад, жіночим або чоловічим голосом. Це дозволяє адаптувати звучання до особистих уподобань користувача, а також створює додатковий ефект реалістичності, оскільки озвучення максимально наближене до природного мовлення.

Особливу увагу приділено підтримці темного режиму. Відповідне налаштування дає змогу перемикати між світлим і темним візуальним оформленням інтерфейсу. Це покращує ергономіку роботи в умовах поганого освітлення та зменшує навантаження на зір, особливо при тривалому користуванні у вечірній час.

У верхній частині екрана розміщено інформаційний блок про користувача. Він містить ім'я користувача, тип облікового запису та кнопку

для редагування профілю. Це забезпечує швидкий доступ до персональних даних і надає відчуття персоналізованої взаємодії з додатком.

Загалом екран налаштувань (рисунок 3.12) демонструє приклад ефективного поєднання простоти, логічності та естетичної виразності. Він дозволяє не лише налаштувати зовнішній вигляд, а й глибоко впливати на функціональну поведінку застосунку, що відповідає потребам широкого кола користувачів різного віку.

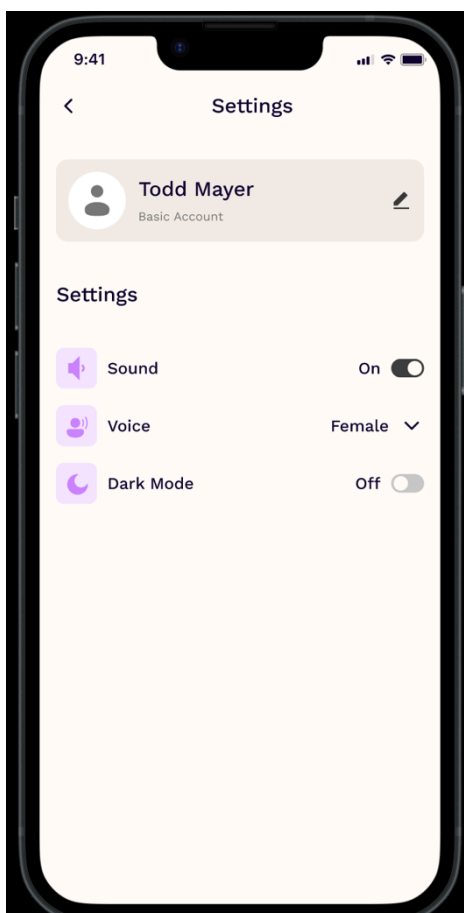


Рисунок 3.12 – Екран налаштувань у світлій темі

Підтримка темної теми інтерфейсу є важливою складовою сучасного користувацького досвіду, особливо з огляду на зростаючі вимоги до комфортного та безпечного використання мобільних застосунків у нічний час або при слабкому освітленні. Нижче видно, як змінюється візуальне оформлення застосунку при активованому темному режимі (рисунок 3.13),

що дозволяє зменшити навантаження на зір, зберігаючи при цьому естетичну цілісність інтерфейсу. Усі елементи залишаються інтуїтивно зрозумілими, завдяки збереженню контрастності, чітких іконок та незмінній логіці розташування компонентів.

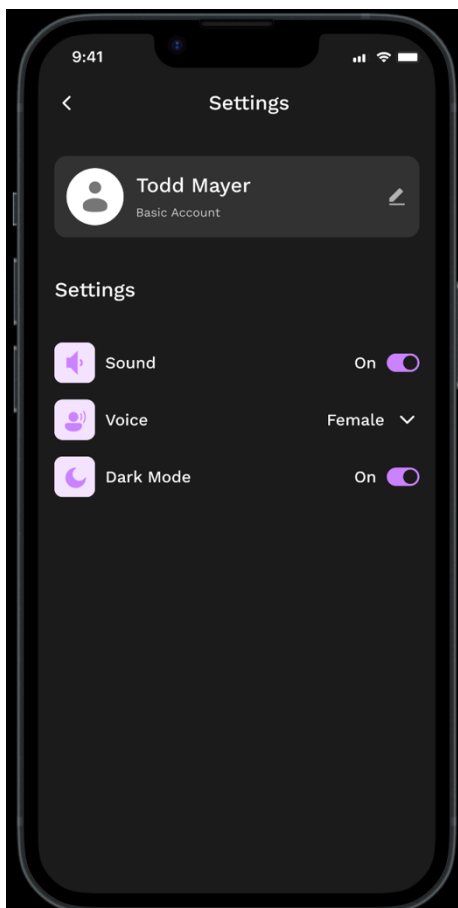


Рисунок 3.13 – Інтерфейс у темній темі

Розроблений інтерфейс мобільного додатку демонструє послідовну реалізацію принципів зручності, адаптивності та візуальної привабливості. Усі екрани, починаючи з головного та завершуючи картками навчання і налаштуваннями, створено з урахуванням потреб цільової аудиторії – дітей, які вивчають іноземну лексику. Кожен екран містить чітко структуровані елементи, інтуїтивно зрозумілі візуальні сигнали та динамічну підтримку різних мов.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи повною мірою реалізовано поставлене завдання – розробка мобільного додатку для вивчення лексики на основі флеш-карток з підтримкою мультимовності та персоналізованого навчання. Було проведено повний цикл проектування програмної системи, починаючи з аналізу предметної галузі й вимог користувачів, завершуючи технічною реалізацією інтерфейсу, моделі даних і функціональної логіки в екосистемі iOS. Усі ключові аспекти, визначені на етапі постановки задачі, реалізовано: підтримка чотирьох мов – української, англійської, німецької, польської, зберігання персонального прогресу користувача, можливість перемикання голосів озвучення, підтримка темної теми, а також стохастичний режим повторення з урахуванням вивченості карток. Розроблений додаток надає зручний механізм фіксації щоденної активності та трекінгу навчальної серії, що сприяє формуванню сталої мотивації до навчання. Якісні показники ефективності системи, що проявляються у швидкості завантаження тем, точності фільтрації карток за мовами, відгуку на взаємодію користувача та адаптації контенту до вибраних налаштувань, підтверджують відповідність реалізації сучасним стандартам мобільної розробки.

Порівняння реалізованого рішення з аналогічними додатками на ринку свідчить про досягнення певної конкурентної переваги у вузькому сегменті – персоналізованого лексичного навчання дітей з підтримкою мультимовності та локального зберігання прогресу. У той час як більшість популярних сервісів зосереджуються на інтерактивних курсах або гейміфікованих рівнях, представлений мобільний додаток вирізняється простою, але гнучкою моделлю навчання, що не вимагає підключення до зовнішніх серверів чи авторизації. Завдяки використанню SwiftData реалізовано ефективне зберігання всіх даних локально, що робить застосунок придатним для використання в умовах обмеженого або

відсутнього доступу до інтернету. Особливістю реалізації є підтримка зміни мови в режимі реального часу, що дає змогу не лише вивчати окремі слова, а й формувати зв'язки між мовними відповідниками в різних мовах, чого зазвичай не передбачають стандартні додатки. Більш того, гнучка архітектура на основі MVVM дозволяє легко розширювати систему новими мовами, темами та механіками без необхідності кардинальних змін у структурі коду.

У подальшій роботі рекомендується зосередити зусилля на кількох напрямках. По-перше, варто реалізувати хмарну синхронізацію прогресу з можливістю входу до облікового запису, що дозволить зберігати результати на декількох пристроях і забезпечити відновлення після втрати даних. По-друге, перспективним є впровадження мікрогейміфікації у вигляді простих досягнень, нагород і адаптивного рівневого зростання. Це підвищить мотивацію користувача, особливо у випадку роботи з дітьми. По-третє, доцільним є доповнення системи статистикою складності слів і частоти їхнього повторення, що дозволить будувати інтелектуальну траєкторію навчання з урахуванням сильних і слабких сторін користувача. Крім того, варто дослідити можливість інтеграції з технологіями розпізнавання мовлення, що відкриє шлях до реалізації функції вимови та інтерактивної перевірки. Зрештою, масштабування додатку до Android-платформи забезпечить ширше охоплення аудиторії й уможливить подальше тестування на більшій кількості пристроїв.

Таким чином, реалізований мобільний додаток повністю відповідає визначеним технічним і функціональним вимогам, демонструє сучасний рівень технічної реалізації на базі UIKit і SwiftData, має потенціал для подальшого розвитку в комерційний або освітній продукт і може використовуватися як приклад якісного застосування концепцій персоналізованого навчання в умовах автономного мобільного середовища.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Apple Inc. UIKit Framework Reference.  
URL: <https://developer.apple.com/documentation/uikit> (date of access: 16.01.2025).
2. Barcroft J. Vocabulary in Language Teaching. Routledge, 2015.
3. Chou P. A. The effects of flashcard-based instruction on vocabulary learning. *System*. 2014. Vol. 42. P. 314–322.
4. Freepik. Vector graphics for educational interfaces.  
URL: <https://www.freepik.com> (date of access: 12.02.2025).
5. Inc A. AVFoundation Framework Reference.  
URL: <https://developer.apple.com/documentation/avfoundation> (date of access: 16.01.2025).
6. Inc A. SwiftData – Persisting model data in SwiftUI.  
URL: <https://developer.apple.com/xcode/swiftdata/> (date of access: 22.01.2025).
7. Inc A. Swift Programming Language (Swift 5.9).  
URL: <https://developer.apple.com/documentation/swift> (date of access: 25.01.2025).
8. Inc A. SwiftUI Essentials.  
URL: <https://developer.apple.com/documentation/swiftui> (date of access: 22.01.2025).
9. ISO/IEC. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. 2011.
10. Krashen S. The Input Hypothesis: Issues and Implications. Longman, 1985.
11. Lin H., Chen T. Designing an interactive vocabulary learning system on mobile devices. *Journal of Educational Technology*. 2019. Vol. 16, no. 4. P. 55–64.

12. Nation P. *Learning Vocabulary in Another Language*. Cambridge University Press, 2013.
13. Nielsen J. *Usability Engineering*. Academic Press, 1993.
14. Oxford R. *Language Learning Strategies: What Every Teacher Should Know*. Heinle & Heinle, 1990.
15. Pappas I., Pateli A. Mobile apps as vocabulary learning tools: affordances and limitations. *Education and Information Technologies*. 2021. Vol. 26. P. 1311–1330.
16. Pilnyak P. Building multilingual apps with SwiftUI and localization. *Medium*. 2023. URL: <https://medium.com/@pilnyak/multilingual-swiftui-app> (date of access: 28.02.2025).
17. Reinders M. Developing a flashcard-based vocabulary learning app for iOS using Swift and Core Data. 2021.
18. Richards J., Rodgers T. *Approaches and Methods in Language Teaching*. Cambridge University Press, 2001.
19. Sundell J. Data modeling in Swift using SwiftData. *Swift by Sundell*. 2023. URL: <https://www.swiftbysundell.com/articles/data-modeling-using-swiftdata/> (date of access: 27.02.2025).
20. Tangney B. An Interactive Learning System to Support Vocabulary Acquisition in Children. *Computer & Education*. 2005.