

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Навчально-науковий інститут кібернетики, інформаційних технологій та
інженерії
Кафедра комп'ютерних технологій та економічної кібернетики

Допущено до захисту:

Завідувач кафедри

_____ д. е. н., проф. П. М. Грицюк

« _____ » _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня «магістр»

за освітньо-професійною програмою «Інформаційні технології в бізнесі»
спеціальності 126 «Інформаційні системи та технології»

на тему: **«Мобільний застосунок для словника з англійської мови»**

Виконав:

здобувач вищої освіти 2 курсу, групи ІСТ-61м
Сех Олександр Богданович

Керівник:

канд. техн. наук, доцент Гладка О. М.

Рецензент:

канд. техн. наук, доцент Джоші О. І.

Рівне – 2024

Національний університет водного господарства та природокористування
ННІ кібернетики, інформаційних технологій та інженерії
Кафедра комп'ютерних технологій та економічної кібернетики

Освітньо-кваліфікаційний рівень – **магістр**
за освітньо-професійною програмою **«Інформаційні системи і технології»**
спеціальність **126 «Інформаційні системи та технології»**

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ д. е. н., проф. П. М. Грицюк

« _____ » _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачу _____ ***Сеху Олександрю Богдановичу***
(прізвище, ім'я, по-батькові)

1. Тема роботи ***Мобільний застосунок для словника з англійської мови***

керівник роботи: ***Гладка Олена Миколаївна, канд. техн. наук, доцент***
(прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)

затверджена наказом по університету від “ ***26*** ” ***вересня*** 2024 р. С № ***1320***

2. Термін здачі здобувачем закінченої роботи “ ***29*** ” ***листопада*** 2024 р.

3. Вихідні дані до роботи ***розробити мобільний застосунок для словника з англійської мови, що передбачає проєктування та створення бази даних, розробку відповідних інтерфейсів і функціональних модулів застосунку засобами Android Studio і Kotlin.***

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) ***вступ, обґрунтування актуальності роботи; аналіз предметної області та аналогічних розробок; вибір і обґрунтування засобів розробки програмного продукту; проєктування, розробка і опис створеного мобільного застосунку.***

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1.	Гладка О. М., доцент		
2.	Гладка О. М., доцент		
3.	Гладка О. М., доцент		

7. Дата видачі завдання “ 27 ” вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз об'єкта дослідження, виявлення існуючих проблем	27.09.24 – 10.10.24	
2.	Аналіз існуючих інформаційних методів (технологій)	10.10.24 – 20.10.24	
3.	Вибір та обґрунтування засобів розробки мобільного застосунку	30.10.24 – 01.11.24	
4.	Проектування, розробка та реалізація мобільного застосунку	02.11.24 – 12.11.24	
5.	Підготовка тексту кваліфікаційної роботи	15.11.24 – 29.11.24	
6.	Підготовка презентації роботи	02.12.24 – 13.12.24	
7.	Відгук керівника, рецензування роботи, перевірка на плагіат	02.12.24 – 23.12.24	

Здобувач _____ **О. Б. Сех**
(підпис) (прізвище і ініціали)

Керівник кваліфікаційної роботи

О. М. Гладка
(підпис) (прізвище і ініціали)

АНОТАЦІЯ

Сех О. Б. Мобільний застосунок для словника з англійської мови. Кваліфікаційна робота на здобуття ступеня "магістр" за освітньо-професійною програмою «Інформаційні технології в бізнесі» спеціальності 126 «Інформаційні системи та технології»: 57 с., 22 рис., 1 додаток, 13 літературних джерел.

Метою роботи є розробка мобільного застосунку для підтримки процесу вивчення, актуалізації чи кращого опанування навичками володіння англійською мовою. Об'єктом дослідження є можливості використання електронного словника в процесі вивчення англійської мови за допомогою мобільних застосунків. Предмет дослідження – проектування та розробка мобільного застосунку для словника з англійської мови.

У процесі дослідження були виконані такі кроки: аналіз предметної області та систем-аналогів, проектування бази даних для зберігання словникових записів, розробка користувацького інтерфейсу, реалізація функціоналу пошуку та додавання нового контенту.

Для розробки мобільного застосунку використано платформу Android Studio та мову програмування Kotlin. База даних реалізована з використанням Realtime Database Firebase.

Результатом роботи є функціональний мобільний застосунок, який дозволяє користувачам швидко знаходити значення слів, додавати нові записи, створювати персоналізовані списки слів для вивчення, а також отримувати аналітику щодо прогресу у вивченні лексики.

Ключові слова: мобільний застосунок, словник, англійська мова, автоматизація, Android Studio, Kotlin, база даних Firebase.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. Аналіз предметної області та існуючих аналогів	8
1.1. Описання та аналіз об'єкта дослідження	8
1.2. Аналіз програм-аналогів	11
РОЗДІЛ 2. Опис моделі та методів її реалізації	22
2.1. Вимоги до мобільного застосунку словника	22
2.2. Опис моделі мобільного застосунку словника	23
2.3. Проектування мобільного застосунку за допомогою UML-діаграм	25
2.4. Вибір технологій розробки	31
РОЗДІЛ 3. Опис програмної реалізації мобільного застосунку	36
3.1. Створення бази даних	36
3.2. Кодування в Android Studio	38
3.3. Опис інтерфейсу мобільного додатку	45
3.4. Опис застосунку	52
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТОК А	57

ВСТУП

Сучасні технології значно змінюють підходи до організації процесів навчання та управління інформацією. Мобільні пристрої та застосунки відкрили нові можливості для оптимізації освітніх процесів, зробивши навчання доступнішим, інтерактивним і персоналізованим. Серед багатьох напрямів використання мобільних застосунків особливу увагу привертають інструменти для вивчення мов.

Вивчення англійської мови, яка є однією з найпоширеніших у світі, стає важливим елементом особистісного та професійного розвитку. Сучасні методи навчання потребують використання інноваційних технологій, що спрощують доступ до знань, підтримують індивідуальні темпи навчання та сприяють підвищенню мотивації. Однак традиційні інструменти, такі як друковані словники, вже не задовольняють потреб сучасного користувача, адже вони є малоефективними в умовах теперішнього життя.

Основною проблемою є необхідність створення універсального рішення, яке поєднувало б зручність використання, швидкість пошуку інформації та інтерактивність. Розробка мобільного застосунку для словника з англійської мови дозволить підвищити ефективність вивчення мови, забезпечити швидкий доступ до необхідної лексики, полегшити процес запам'ятовування слів і відстеження прогресу у навчанні.

Метою даної роботи є розробка мобільного застосунку для підтримки процесу вивчення, актуалізації чи кращого опанування навичками володіння англійською мовою. Для досягнення цієї мети були поставлені наступні завдання:

1. Аналіз предметної області та систем-аналогів.
2. Проектування бази даних для зберігання словникових записів.

3. Розробка користувацького інтерфейсу, який буде зручним у використанні.
4. Реалізація функціоналу системи з використанням платформи Android Studio та мови програмування Kotlin.

Об'єктом дослідження є можливості використання електронного словника в процесі вивчення англійської мови за допомогою мобільних застосунків. Предмет дослідження – проектування та розробка мобільного застосунку для словника з англійської мови.

Кваліфікаційна робота складається з трьох розділів, які описують різні аспекти об'єкта дослідження. Перший розділ присвячений аналізу предметної області та існуючих рішень. Другий розділ розглядає вибір технологій та описує проектування мобільного застосунку. Третій розділ присвячений програмній реалізації застосунку.

РОЗДІЛ 1. Аналіз предметної області та існуючих аналогів.

1.1. Описання та аналіз об'єкта дослідження

Вивчення англійської мови є однією з основних потреб сучасного глобалізованого суспільства, де знання іноземних мов стало важливим фактором успіху в професійній діяльності, навчанні та міжособистісному спілкуванні [3]. У світі, де комунікація між людьми різних культур і країн відбувається на щоденній основі, здатність вільно спілкуватися англійською мовою відкриває нові можливості для розвитку особистості та кар'єри. Це особливо важливо у контексті зростання значення англійської мови як глобальної мови бізнесу, науки та технологій. У зв'язку з цим, питання ефективного та зручного вивчення англійської мови набуває все більшої актуальності, особливо в умовах швидких змін на ринку освітніх послуг та технологій.

Мобільні застосунки для вивчення іноземних мов, зокрема англійської, є одними з найпопулярніших і доступних інструментів, які дозволяють користувачам в будь-який час та в будь-якому місці покращувати свої мовні навички [1-2]. Завдяки мобільним технологіям, навчання стало значно доступнішим та ефективнішим, оскільки дозволяє інтегрувати навчальний процес у повсякденне життя, не вимагаючи постійної прив'язки до часу і місця[4]. Мобільні додатки дозволяють зручно організувати процес вивчення, забезпечуючи доступ до різноманітних ресурсів та функцій, таких як інтерактивні вправи, словники, відео та аудіоматеріали.

Об'єктом дослідження цієї роботи є мобільний застосунок для вивчення англійської мови, а саме електронний словник, який повинний не лише забезпечити користувачам зручний доступ до лексичних одиниць, але й підвищити ефективність процесу навчання завдяки розширеним функціям. У

сучасному світі інформаційних технологій існує безліч інструментів для вивчення мов, але саме мобільні додатки стали одним із найбільш популярних і зручних варіантів.

Мобільні словники є особливо корисними завдяки їх доступності, швидкості і зручності. Вони дозволяють користувачам швидко отримати необхідну інформацію про слова, їхні значення, вимову та приклади вживання, що значно полегшує процес навчання.

Ринок мобільних додатків для вивчення мов сьогодні є дуже динамічним і різноманітним. Багато компаній та стартапів розробляють нові продукти, які обіцяють зробити процес вивчення мови швидким і ефективним. Однак, незважаючи на широкий асортимент, більшість існуючих рішень не здатні повністю задовольнити потреби користувачів, оскільки вони мають кілька суттєвих недоліків, які можуть ускладнювати процес навчання.

Першою значною проблемою є недостатня інтерактивність традиційних мобільних словників. Багато електронних словників обмежуються лише базовими функціями перекладу і пояснення значень слів, не надаючи достатньої кількості прикладів їх використання в реальних життєвих ситуаціях. Наприклад, традиційні словники надають лише суху інформацію про те, що означає те чи інше слово, не враховуючи контексту, в якому це слово може бути використане. Це створює труднощі для користувачів, які хочуть краще розуміти, як і коли використовувати певне слово.

Ще однією проблемою є недоступність додаткових функцій для моніторингу прогресу в навчанні. Багато мобільних додатків не забезпечують можливість відстеження успіхів користувачів, що призводить до зниження мотивації. Без системи оцінювання та відстеження навчальних результатів користувачі не можуть адекватно оцінити свій рівень володіння мовою, а

також не мають змоги коригувати стратегію навчання. Користувач може вивчати слова, не знаючи, на якому етапі знаходиться, або витратити надмірно багато часу на вивчення вже знайомих слів.

Крім того, багато мобільних додатків мають проблеми з інтерфейсом, що також ускладнює їх використання. Важливо, щоб користувач міг легко орієнтуватися в додатку та швидко знаходити потрібну інформацію. Якщо інтерфейс є надто складним або непривабливим, це може відлякати користувача і призвести до зниження ефективності навчання.

Мобільний застосунок для вивчення англійської мови повинен враховувати всі проблеми і забезпечити користувачам доступ до ефективного та зручного інструменту для вивчення мови. Зокрема, застосунок повинен бути простим у використанні і мати інтуїтивно зрозумілий інтерфейс. Користувачі повинні мати змогу швидко знаходити потрібні слова та отримувати всю необхідну інформацію про них. Застосунок має також надавати розширену інформацію про кожне слово, включаючи приклади використання, вимову та інформацію про його вживання в різних контекстах. Це дозволить користувачам краще розуміти, коли і як використовувати це слово в реальному житті.

Важливим аспектом є персоналізація навчання. Мобільний застосунок повинен адаптуватися до рівня знань користувача та його інтересів. Наприклад, якщо користувач вивчає англійську для професійних цілей, він повинен мати можливість отримати доступ до спеціалізованої лексики та виразів, пов'язаних з його сферою діяльності. Таким чином, кожен користувач зможе отримувати той навчальний матеріал, який найбільше відповідає його потребам.

Не менш важливою є можливість відстежувати прогрес користувача. Це дозволить не лише оцінювати результати навчання, але й мотивувати користувачів, показуючи їм, як вони покращують свої знання з часом. Вбудовані інструменти для моніторингу прогресу допоможуть користувачам краще організувати свій процес навчання, відстежувати слабкі місця та робити акцент на їх усуненні.

Проблеми, що виникають при вивченні слів, свідчить про необхідність створення сучасних мобільних застосунків. Це полегшує вивчення мови та підвищує ефективність навчання.

1.2. Аналіз програм-аналогів

Сфера мобільних додатків для вивчення англійської мови є важливим напрямком у розвитку інформаційних технологій, оскільки вони сприяють покращенню процесу навчання, забезпечують доступність матеріалів та інтерактивний підхід до вивчення. Такі додатки дозволяють користувачам зосереджуватися на ключових аспектах мови, включаючи лексику, граматику, вимову та контекстне використання. Особливою популярністю користуються програми Oxford Dictionary, Merriam-Webster Dictionary, Advanced Learner's Dictionary та Duolingo.

«Oxford Dictionary» – це один із найвідоміших та найбільш авторитетних словників англійської мови, який став незамінним інструментом для мовознавців, студентів, викладачів та професіоналів у різних галузях. Його мобільна версія поєднує традиційні можливості друкованого видання із сучасними технологіями, роблячи доступ до мовних ресурсів ще простішим та зручнішим.

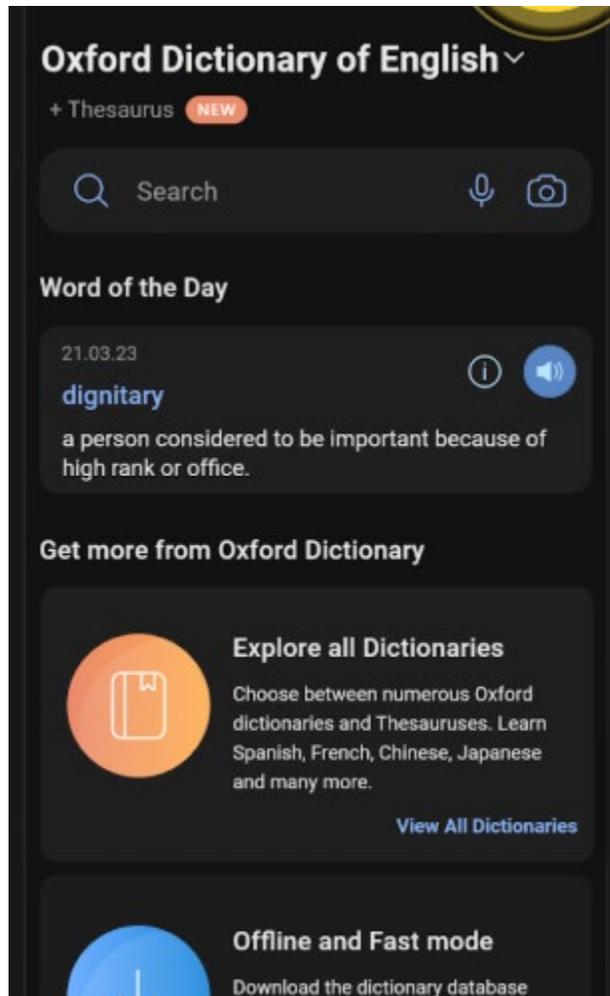


Рис.1.1. Головний екран «Oxford Dictionary»

Основні характеристики Oxford Dictionary:

- Розширена база даних.

Програма містить десятки тисяч слів, фраз, ідіом та термінів. Визначення представлені з детальними поясненнями, синонімами, антонімами та додатковою інформацією про походження слова.

- Приклади речень.

Для кожного слова наводяться приклади використання в реальних контекстах, що допомагає краще розуміти стилістичні та семантичні особливості.

- Аудіовимова.

Користувачі можуть прослуховувати слова як у британській, так і в американській вимові. Це особливо корисно для тих, хто прагне покращити свої навички говоріння чи аудіювання.

- Функція "Слово дня".

Щоденні нові слова сприяють поступовому збільшенню словникового запасу без перевантаження користувача.

- Пошук та фільтри.

Застосунок дозволяє швидко знаходити слова за ключовими словами, використовуючи фільтри для уточнення результатів, таких як частини мови чи тематичні групи.

- Офлайн-доступ.

Словник можна використовувати без підключення до інтернету, що є великою перевагою для подорожей або роботи в зонах з обмеженим доступом до мережі.

Переваги:

- Авторитетність джерела: Oxford Dictionary визнаний як стандарт для англійської мови.
- Різноманітність мовних ресурсів: Він підходить як для академічного вивчення мови, так і для повсякденного використання.
- Зручність у використанні: Інтуїтивно зрозумілий інтерфейс дозволяє швидко знаходити потрібну інформацію.
- Покращення вимови: Аудіовимова допомагає уникнути помилок та формує правильні мовні навички.

- **Офлайн-доступ:** Дозволяє працювати зі словником у будь-який час і будь-де.

Недоліки:

- **Відсутність інтерактивності:** Програма не містить тестів, вправ чи ігрових елементів, які могли б зробити процес навчання більш захопливим.
- **Вартість преміум-підписки:** Хоча базова версія програми є, розширені функції доступні лише за додаткову плату.
- **Обмежені можливості персоналізації:** Програма не завжди враховує рівень знань користувача чи його конкретні інтереси.

Merriam-Webster Dictionary - це авторитетний американський словник, який містить визначення більше ніж 300,000 слів. Він постійно оновлюється новими словами та значеннями і є частиною компанії Merriam-Webster, Incorporated, яка також публікує інші довідкові книги, включаючи словники синонімів і географічні терміни.

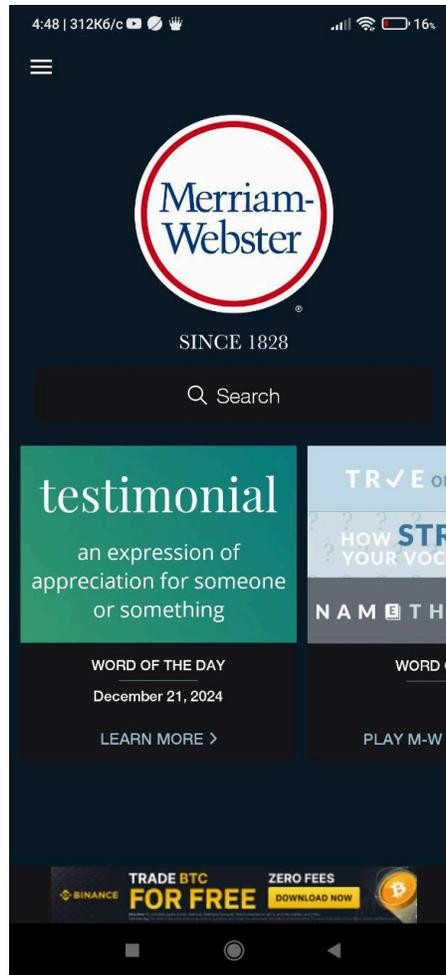


Рис.1.2. Головний екран «Merriam-Webster Dictionary»

Проаналізуємо по основним вимогам.

Відповідність основним вимогам:

1. Доступність і простота використання:

Merriam-Webster Dictionary пропонує інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам швидко знаходити необхідну інформацію про слова. Додаток доступний для різних платформ, включаючи Android та iOS.

2. Розширена інформація про слова:

Додаток містить докладні визначення, синоніми, антоніми та приклади використання слів у контексті, що сприяє поглибленню знань.

3. Інтерактивність:

Завдяки вікторинам та тестам, додаток залучає користувачів до активного навчання, роблячи процес вивчення цікавим та мотивуючим.

4. Додаткові функції:

Голосовий пошук та аудіовимова спрощують використання програми та допомагають у розвитку навичок слухання та правильної вимови.

Переваги Merriam-Webster Dictionary:

1. Інтерактивний підхід: Вікторини, "Слово дня" та інші інтерактивні елементи мотивують користувачів.
2. Доступність безкоштовної версії: Більшість функцій додатка доступна безкоштовно, що робить його популярним серед широкої аудиторії.
3. Різноманітність контенту: Велика база даних слів із додатковою інформацією, як-от етимологія чи частота використання, є корисною для різних категорій користувачів.
4. Універсальність: Додаток підходить як для початківців, так і для тих, хто прагне поглибити свої знання.

Недоліки Merriam-Webster Dictionary:

1. Відсутність персоналізації: Додаток не враховує індивідуальні потреби користувачів, як рівень знань чи специфічна тематика слів.
2. Обмежене охоплення контексту: Інформація про стилістичне чи ситуаційне використання слів є недостатньою, що може обмежувати розуміння складних контекстів.
3. Обмежена функціональність для навчання граматики: Акцент зроблено переважно на лексику, без інтеграції граматичних вправ чи правил.

4. Платний преміум-доступ: Деякі додаткові функції доступні лише за підпискою, що може бути обмеженням для частини користувачів.

Cambridge Advanced Learner's Dictionary – це словник, призначений для молодих людей та дорослих, що містить 140 000 слів, фраз і прикладів британської англійської мови. Він особливо підходить для підготовки до таких іспитів, як IELTS, BEC, а також Cambridge English First, Advanced і Proficiency.

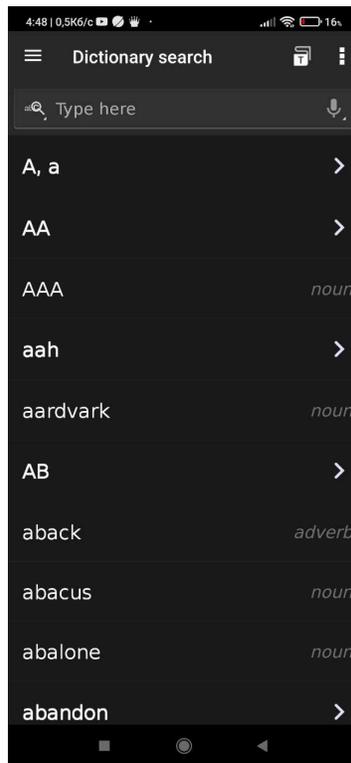


Рис.1.3. Головний екран «Cambridge Advanced Learner's Dictionary»

Характеристика застосунку:

- Якість лексикографічного контенту:

CALD пропонує велику базу даних слів і фраз, адаптовану до потреб тих, хто вивчає англійську. Слова пояснюються простою мовою, що сприяє легкому розумінню навіть для новачків.

- Контекст використання:

Для кожного слова доступні приклади використання в реальних ситуаціях, що допомагає користувачам краще зрозуміти значення слова та його стилістичні особливості.

- Граматична підтримка:

CALD надає граматичні підказки, що дозволяє користувачам одночасно вивчати лексику та граматику, поєднуючи ці аспекти у навчальному процесі.

- Доступність:

Застосунок доступний на популярних платформах (Android та iOS), а його простий інтерфейс забезпечує легкість використання.

- Інтерактивність:

CALD, хоча і пропонує якісний лексикографічний контент, обмежений у плані інтерактивності, оскільки не містить ігрових елементів чи тестів для закріплення знань.

Переваги CALD:

1. Прості та зрозумілі визначення: Додаток використовує просту лексику для пояснення значень слів, що зручно для тих, хто вивчає англійську мову.
2. Реальні приклади використання: Це сприяє кращому розумінню контексту та стилістики.
3. Граматичні підказки: Інтеграція граматики в навчальний процес допомагає користувачам розуміти правила використання слів.
4. Тематика: Наявність тематичних списків слів дозволяє зосереджуватися на певних аспектах лексики, залежно від мети навчання (наприклад, робота, подорожі тощо).

Недоліки CALD

1. Відсутність інтерактивних функцій: Немає ігор чи вікторин, які могли б зробити навчання цікавішим.
2. Відсутність адаптації: Додаток не враховує рівень знань користувача та його індивідуальні потреби, що може знизити ефективність навчання.
3. Обмежена функціональність у відстеженні прогресу: Немає інструментів для моніторингу результатів чи оцінки успішності.

Duolingo - це один з найпопулярніших мобільних додатків для вивчення мов, який завоював популярність завдяки своїй інтуїтивності та інтерактивному підходу до навчання. Додаток використовує елементи гейміфікації, що дозволяє користувачам засвоювати нові слова і фрази в ігровій формі, отримуючи бали за виконання вправ і досягнення певних рівнів. Це робить процес вивчення мов не лише ефективним, а й цікавим. Duolingo пропонує широкий вибір мов, серед яких англійська, французька, німецька, іспанська та інші. Завдяки простому інтерфейсу і доступності на різних платформах (iOS, Android, веб), додаток став популярним серед користувачів з різними рівнями володіння мовою.

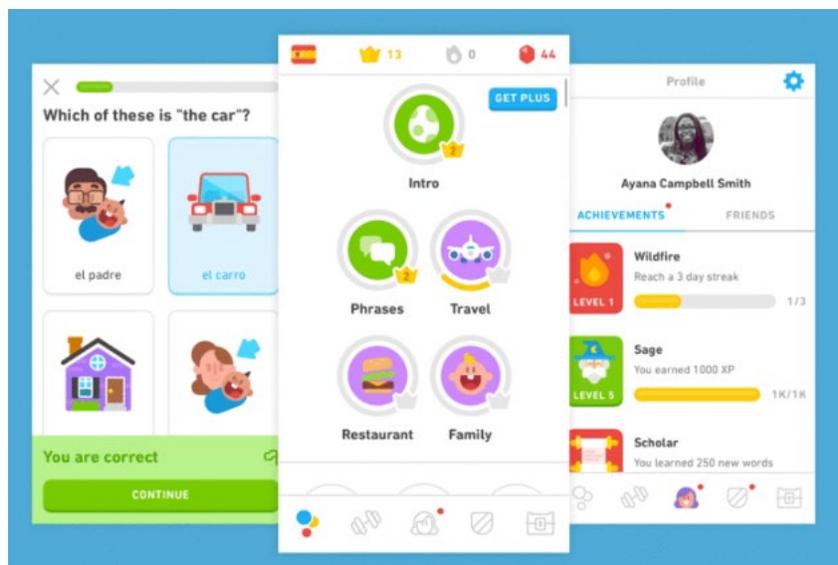


Рис.1.4. Інтерфейс мобільного застосунку «Duolingo»

Основною перевагою Duolingo є його доступність та безкоштовний формат, що дозволяє користувачам безкоштовно вивчати різні мови. Додаток пропонує різноманітні вправи, які допомагають розвивати всі ключові мовні навички: слухання, говоріння, читання і письмо. Завдяки системі досягнень та лідербордів, Duolingo спонукає користувачів до регулярного навчання та мотивації, створюючи елементи конкуренції і відстежуючи їхній прогрес.

Однак, хоча Duolingo і є дуже ефективним для початкового та середнього рівнів, його функціональність може бути обмеженою для тих, хто хоче досягти глибшого розуміння мови або покращити свої навички на більш високому рівні. Додаток в основному фокусується на базовій лексиці та граматиці, але не надає достатньо складних граматичних пояснень або контекстуальних прикладів для глибшого розуміння мови. Важливим недоліком є те, що Duolingo не пропонує повноцінної практики реального спілкування, а вправи для розмовної мови обмежуються простими фразами. Також є певні обмеження у вимові, оскільки система розпізнавання мови може не завжди точно оцінювати правильність вимови користувача.

Програмне забезпечення для вивчення мов є важливим інструментом, що дозволяє користувачам ефективно опановувати нові мови в сучасному інформаційному середовищі. Мобільні додатки, такі як Oxford Dictionary, Merriam-Webster, Cambridge Advanced Learner's Dictionary та Duolingo, пропонують різноманітні функції, що забезпечують зручність, доступність та інтерактивність у процесі навчання. Однак, кожен із цих додатків має свої переваги та недоліки. Наприклад, Oxford Dictionary та Merriam-Webster забезпечують вичерпні словникові дані, але їхній інтерфейс може бути менш зручним для початківців. Duolingo є привабливим завдяки своїй гейміфікації, проте його можливості для глибокого вивчення мови обмежені. Cambridge

Advanced Learner's Dictionary є корисним для детального вивчення значень слів, але не надає інтерактивної практики мовних навичок.

Таким чином, зважаючи на виявлені недоліки та потреби користувачів, доцільно розробити новий мобільний додаток, який поєднає функції інтуїтивно зрозумілого інтерфейсу, розширену лексичну інформацію. Розроблений додаток має стати ефективним інструментом для вивчення англійської мови.

РОЗДІЛ 2. Опис моделі та методів її реалізації.

2.1. Вимоги до мобільного застосунку словника

Мобільні застосунки для вивчення мов стали незамінним інструментом у процесі навчання. Вони дають змогу швидко отримувати доступ до словникової інформації та ефективно покращувати мовні навички. Щоб мобільний додаток словника був корисним і зручним, необхідно враховувати кілька ключових вимог до його функціональності та дизайну.

Однією з основних вимог є простота та інтуїтивно зрозумілий інтерфейс. Користувач повинен без труднощів знаходити необхідну інформацію, не витрачаючи зайвий час на пошук. Інтерфейс має бути зрозумілим навіть для новачків, тому він повинен бути максимально простим, без зайвих елементів, які можуть відволікати увагу.

Додаток також повинен мати набір функцій, що дозволяють користувачу зручно працювати зі словами. Це має бути швидкий пошук за різними критеріями, наприклад, за буквами чи частинами мови, а також наявність аудіо для правильної вимови слів. Крім того, важливо, щоб словник надавав контекстні приклади, які допоможуть краще зрозуміти значення слів та їхнє використання в реальному житті. Крім того, важливо, щоб додаток міг відслідковувати прогрес користувача, надаючи статистику про вивчені слова.

Не менш важливим є забезпечення зручності використання на андроїд платформах. Мобільний додаток повинен забезпечувати комфортний досвід на різних пристроях, від смартфонів до планшетів.

З урахуванням цих вимог, мобільний застосунок для вивчення мови може стати ефективним інструментом, що дозволить користувачам швидко вивчати нові слова.

2.2. Опис моделі мобільного застосунку словника

Мобільний додаток словника для вивчення англійської мови має бути зручним, функціональним і інтуїтивно зрозумілим для користувачів. Розглянемо модель, яка включає основні компоненти, необхідні для досягнення цих цілей (рис.2.1).

Наведемо загальні вимоги мобільного додатка словника та його основні функціональні елементи:

- Головний екран — відображає основні категорії користувача: пошук слова, історія запитів, персоналізовані рекомендації та статистики прогресу.
- Пошук слів — користувач може швидко знайти слово, вводячи його вручну або за допомогою голосового введення. Після вибору слова надається повна інформація: визначення, приклади використання, вимова, синоніми та антоніми.
- Функціонал додавання нових слів — користувач може додавати нові слова до свого словника, включаючи власні зауваження або примітки.
- Історія пошуку — зберігаються всі нещодавно пошукові запити користувача для швидкого доступу до повторного використання.
- Прогрес — користувачі можуть переглядати статистику своєї активності, кількість вивчених слів, час, витрачений на навчання, а також досягнуті результати.

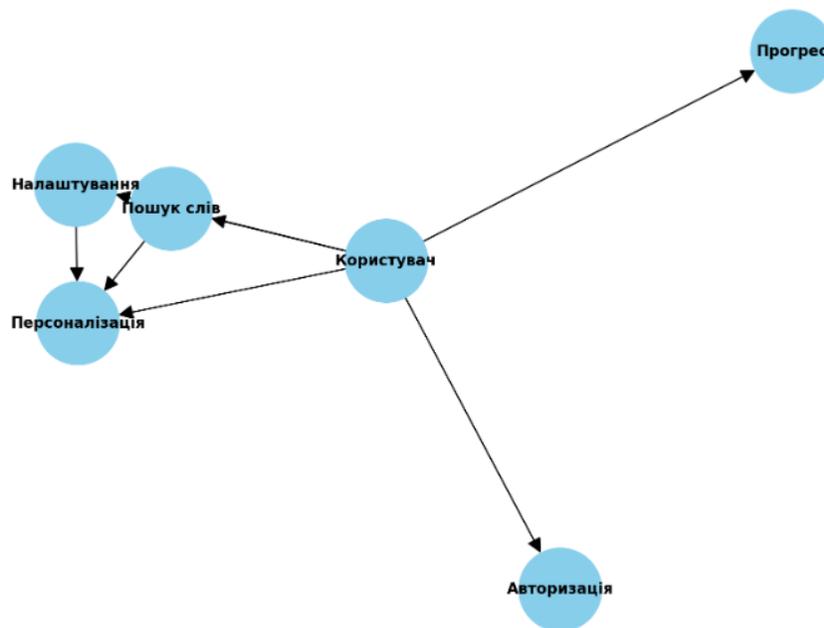


Рис.2.1. Організаційна модель додатка

Нижче показано загальну схему роботи мобільного додатка словника, яка включає необхідні функції та компоненти (рис.2.2):

- Авторизація та реєстрація — користувачі можуть створювати облікові записи або входити до існуючих облікових записів для збереження персональних налаштувань і прогресу навчання.
- Пошук та перегляд слів — пошук здійснюється через введення тексту або через відповідні кнопки. Після вибору слова користувач отримує детальну інформацію.
- Додавання слова у базу даних.
- Прогрес — додаток зберігає дані про навчання користувача, включаючи кількість вивчених слів.

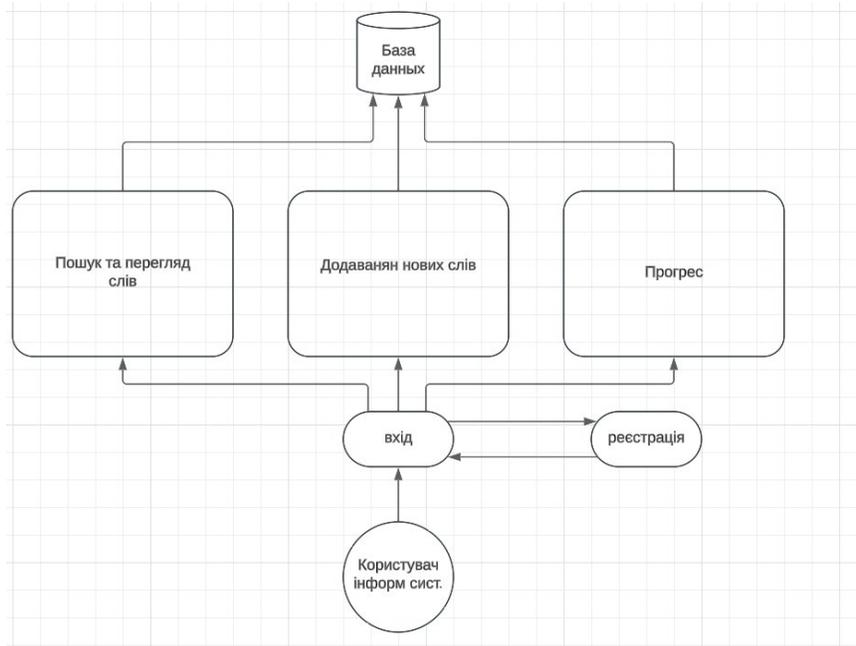


Рис.2.2. Загальна схема мобільного додатка

2.3. Проектування мобільного застосунку за допомогою UML-діаграм

Для логіки побудови мобільного застосунку було розроблено декілька діаграм UML. Зокрема були використанні наступні діаграми: діаграма класів, компонентів, прецедентів, активностей та послідовностей.

UML-діаграми — це візуальні моделі, створені за допомогою уніфікованої мови моделювання (UML), яка використовується для опису, проектування та документування систем у програмному забезпеченні[5-6]. Вони допомагають зрозуміти структуру і поведінку системи, а також полегшують комунікацію між розробниками та замовниками.

На зображенні 2.3. показана діаграма класів, яка містить класи, методи та зв'язки між ними.

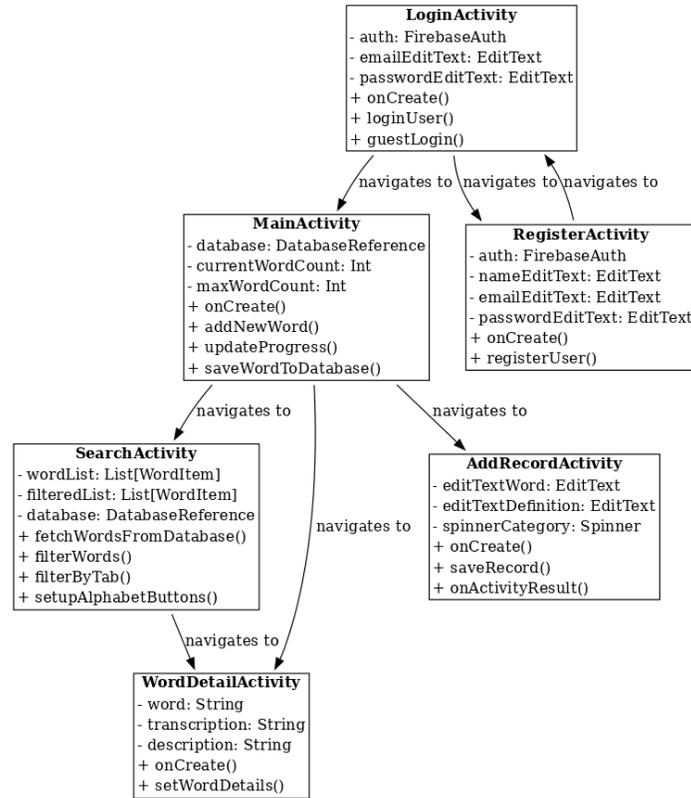


Рис.2.3. Діаграма класів застосунку

Наступна діаграма компонентів на рисунку 2.4 відображає взаємодії основних компонентів мобільного додатку. Кожен компонент відповідає певній функціональності або активності. Деталі діаграми:

Компоненти

1. LoginActivity (Логін):

- Відповідає за авторизацію користувача.
- Використовує Authentication Interface (інтерфейс аутентифікації) для перевірки облікових даних.
- Може перенаправляти користувача до MainActivity (після успішного входу) або RegisterActivity (якщо користувач ще не зареєстрований).

2. RegisterActivity (Реєстрація):

- Дозволяє новим користувачам створити обліковий запис.

- Використовує Authentication Interface для створення нового користувача.
 - Повертає користувача до LoginActivity після успішної реєстрації.
3. MainActivity (Головна активність):
- Головний екран програми, що об'єднує основні функції.
 - Використовує Database Interface для взаємодії з базою даних.
 - Може перенаправляти користувача до таких компонентів:
SearchActivity (пошук слів),
WordDetailActivity (перегляд деталей слова),
AddRecordActivity (додавання нового запису).
4. SearchActivity (Пошук):
- Відповідає за пошук слів у базі даних.
 - Використовує Database Interface для завантаження та фільтрації списку слів.
 - Може перенаправляти користувача до WordDetailActivity для перегляду деталей обраного слова.
5. WordDetailActivity (Деталі слова):
- Відображає детальну інформацію про вибране слово (опис, транскрипцію тощо).
6. AddRecordActivity (Додавання запису):
- Дозволяє користувачу додавати нові слова та визначення до бази даних.
 - Використовує Database Interface для збереження записів.

Інтерфейси

1. Authentication Interface:
- Використовується компонентами LoginActivity і RegisterActivity для авторизації та реєстрації користувачів через FirebaseAuth.

2. Database Interface:

- Використовується компонентами MainActivity, SearchActivity та AddRecordActivity для взаємодії з базою даних через DatabaseReference.

Взаємодії

- LoginActivity навігаційно пов'язаний із MainActivity (після входу) та RegisterActivity (при переході до реєстрації).
- MainActivity забезпечує навігацію до інших функціональних компонентів: пошуку, деталей слова та додавання запису.
- SearchActivity дозволяє переходити до WordDetailActivity для перегляду деталей слова.

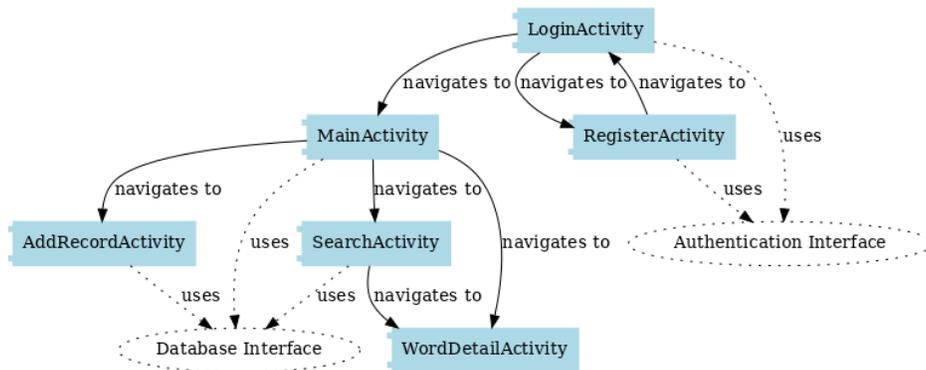


Рис.2.4. Діаграма компонентів

Наступна діаграма на рисунку 2.5, показує послідовність процесів під час виконання пошуку слова. Пояснення:

Процес пошуку слова:

1. Користувач вводить слово для пошуку:
 - Користувач передає запит до Каталогу слів.
2. Каталог слів запитує базу даних:

- Каталог слів надсилає запит до Бази даних, щоб знайти слова, які відповідають пошуковому запиту.
3. База даних повертає результати:
 - База даних передає список відповідних слів до Каталогу слів.
 4. Каталог слів виводить результати користувачу:
 - Каталог слів відображає список знайдених слів для користувача.
 5. Користувач вибирає слово:
 - Користувач вибирає конкретне слово зі списку.
 6. Деталі слова запитують інформацію в бази даних:
 - Деталі слова запитують у Бази даних додаткову інформацію про обране слово.
 7. База даних повертає деталі слова:
 - База даних надсилає деталі слова (опис, транскрипцію тощо) до компонента Деталі слова.
 8. Деталі слова відображають інформацію:
 - Користувачу відображається детальна інформація про обране слово.

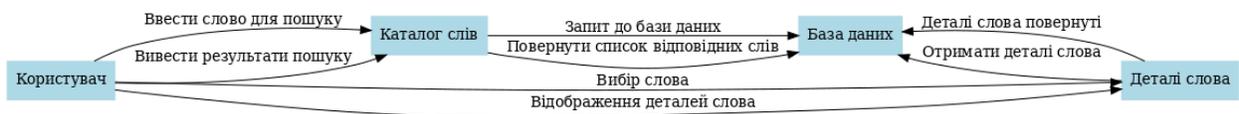


Рис.2.5. Діаграма послідовності пошуку слів

Діаграма активностей надає наочне зображення поведінки системи, тобто процесу, як одна функція переходить до іншої. Нижче наведено діаграму додавання слова у базу даних (рис. 2.6).

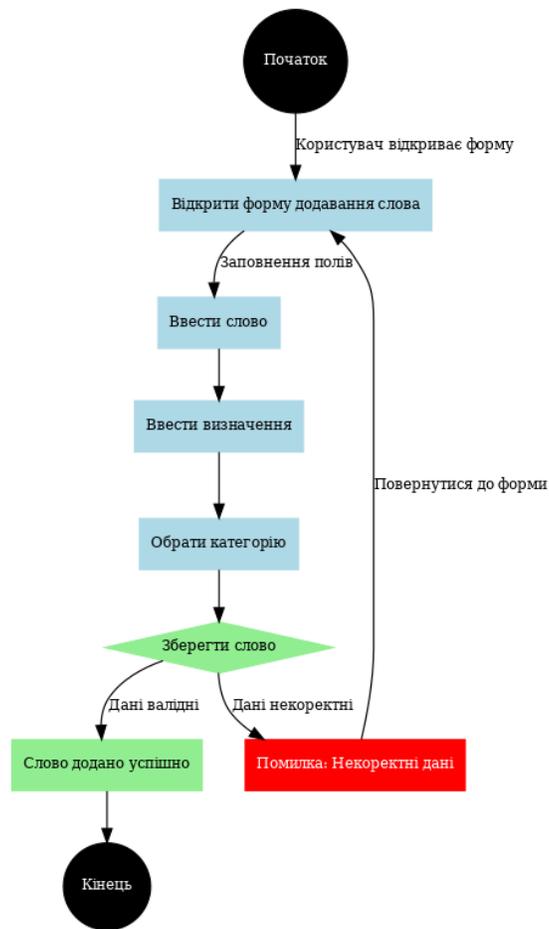


Рис.2.6. Діаграма активностей додавання слова

У діаграмі "Реєстрація та вхід користувача" зображено взаємодію користувачів із системою для доступу до її функцій. Користувач має можливість зареєструватися у системі, використовуючи свої дані (прецедент "Реєстрація"), а також увійти до свого облікового запису, якщо він вже зареєстрований (прецедент "Вхід до системи"). Ці дії дозволяють користувачу отримати персоналізований доступ до функціоналу. Гість, який не має облікового запису, може скористатися "Гостьовим доступом", що дає обмежений набір функцій без необхідності авторизації. Це зручно для нових користувачів, які ще не готові реєструватися, але хочуть ознайомитися з системою.

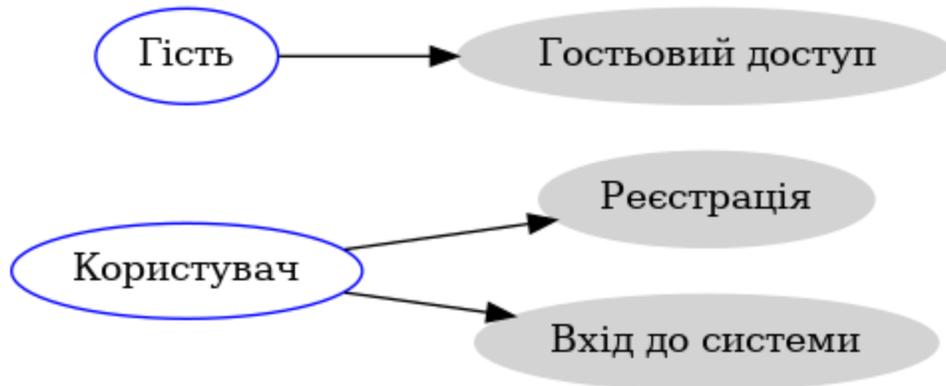


Рис.2.7. Діаграма прецеденту входу

У діаграмі "Додавання та пошук слів" акцент зроблено на взаємодії користувачів із функціоналом, пов'язаним із роботою зі словами. Користувач має повний набір можливостей, включаючи пошук потрібних слів у базі даних (прецедент "Пошук слова") та додавання нових слів до системи (прецедент "Додавання слова"). Це дозволяє Користувачу активно розширювати базу даних та працювати з нею. Гість, на відміну від Користувача, може лише здійснювати пошук слів (прецедент "Пошук слова"). Такий підхід обмежує його дії в системі, але надає доступ до ключового функціоналу, що може спонукати його зареєструватися для отримання повних можливостей.

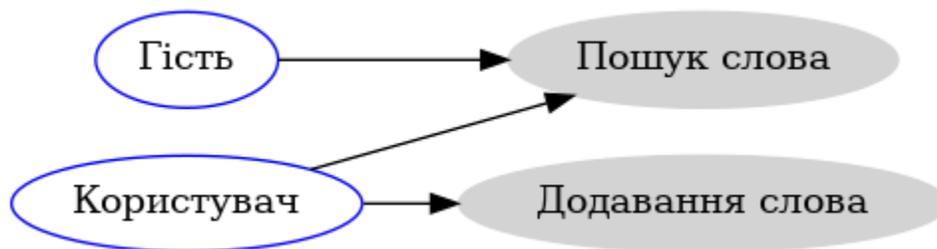


Рис.2.8. Діаграма прецедент додавання слів

2.4. Вибір технологій розробки

Починаючи новий проект, вибір мови програмування є однією з найважливіших речей, які слід враховувати. Вибір відіграє важливу роль у ефективності, масштабованості, обслуговуванні та загальній продуктивності вашої програми. Дві мови, які виділяються в мобільній розробці, це Kotlin і Java. Це широко використовувані, універсальні, надійні, об'єктно-орієнтовані мови, які можуть слугувати надійною основою для ваших потреб розробки.



Рис.2.9. Логотипи

Java є об'єктно-орієнтованою мовою програмування високого рівня, в якій основними структурними елементами є класи та об'єкти [7-8]. Її головною перевагою є незалежність від платформи, що досягається завдяки використанню віртуальної машини Java (JVM). Це означає, що написаний один раз код можна виконувати на будь-якій платформі, де підтримується JVM. Java здобула популярність завдяки своїй здатності обробляти великі навантаження, що робить її надійним вибором для веб-розробки та створення програмного забезпечення. Багато провідних організацій, таких як Google, Uber і Airbnb, інтегрували Java у свої технологічні стеки.

Ще однією перевагою Java є активна спільнота розробників, яка підтримує широкий спектр відкритого коду, бібліотек, фреймворків та інших інструментів [11]. Це сприяє доступу до великої кількості навчальних

матеріалів і ресурсів, що робить Java привабливим вибором для початківців і досвідчених розробників.

Kotlin — це сучасна статично типізована мова програмування з відкритим кодом, розроблена для повної сумісності з JVM [9-10]. Водночас можливості Kotlin виходять за межі JVM, забезпечуючи підтримку інших середовищ, таких як веб-браузери. Ця мова поєднує функціональні та об'єктно-орієнтовані концепції, роблячи її універсальним інструментом для розробки.

Kotlin усуває деякі обмеження Java, пропонуючи більш компактний, виразний та безпечний синтаксис [11]. Завдяки сумісності з Java, Kotlin дозволяє використовувати існуючий код, бібліотеки та фреймворки, що значно спрощує міграцію проєктів або інтеграцію з уже існуючим середовищем.

У 2017 році Google офіційно визнала Kotlin мовою для розробки застосунків на платформі Android. Це рішення стимулювало значне поширення Kotlin серед Android-розробників і сприяло його інтеграції в нові мобільні проєкти.

Однією з ключових функцій Kotlin є можливість компіляції в JavaScript за допомогою технології Kotlin/JS [11]. Це відкриває перспективи використання Kotlin у веб-розробці, дозволяючи інтегрувати унікальні особливості мови з популярними JavaScript-бібліотеками.

Як Kotlin, так і Java є хорошим вибором для веб-розробки та розробки програм. Java має чудову підтримку спільноти та багато бібліотек, які підтримує спільнота. Однак Java багатослівна і не підтримує сучасні мовні функції, такі як нульова безпека. Для порівняння, Kotlin є набагато лаконічнішим, сумісним із Java і має стабільно зростаючу спільноту [11].

Крім того, хоча Java має примітивні типи, які ви повинні явно визначити для змінних і методів, Kotlin розглядає їх як об'єкти та може виводити типи даних. Kotlin також надає різні синтаксичні засоби, такі як класи даних, функції розширення, інтелектуальні приведення та переваження операторів, що робить код виразним і придатним для обслуговування.

NoSQL - це тип нереляційної бази даних, який забезпечує механізми зберігання та обробки даних, що відрізняються від традиційних реляційних баз [12]. Бази даних NoSQL спеціально створені для конкретних моделей даних і зберігають інформацію в гнбких схемах, які легко масштабуються для сучасних програм. Бази даних NoSQL отримали широке поширення у зв'язку з простою розробкою, функціональністю та продуктивністю при будь-яких масштабах.

Однією з основних переваг NoSQL є масштабованість. Такі бази даних легко адаптуються до зростання навантаження за рахунок горизонтального масштабування, що робить їх ідеальними для систем із великим обсягом даних. Крім того, NoSQL підтримує високу швидкість обробки операцій читання та запису, що важливо для застосунків реального часу. NoSQL часто обирають для проєктів, які вимагають гнучкості структури даних, наприклад, у розробці застосунків, що активно змінюються або працюють із нестандартними наборами даних. Завдяки цьому NoSQL є популярним вибором для великих технологічних компаній і стартапів, які працюють у сфері великих даних, аналітики.

Firebase Realtime Database — це рішення для зберігання та синхронізації даних у реальному часі, яке особливо підходить для мобільних застосунків [13]. Це рішення надає простий інтерфейс для синхронізації

даних між клієнтами та сервером, що дозволяє зберігати і оновлювати словникові записи в режимі реального часу. Firebase Realtime Database працює за принципом документо-орієнтованої бази даних, де дані зберігаються у вигляді ієрархії JSON.

Для мобільної розробки словника Firebase Realtime Database може бути ідеальним вибором, оскільки вона пропонує вбудовану підтримку реального часу, легкість у налаштуванні та масштабованість для мобільних застосунків. Вона дозволяє швидко створювати і синхронізувати словникові записи, забезпечуючи зручний досвід для користувачів. У той час як NoSQL є більш універсальним і підходить для проєктів, де потрібна гнучкість у зберіганні даних та велика масштабованість,

РОЗДІЛ 3. Опис програмної реалізації мобільного застосунку.

3.1. Створення бази даних

В базі даних для мобільного додатка створено необхідні поля у форматі .JSON (рис. 3.1 – 3.3).

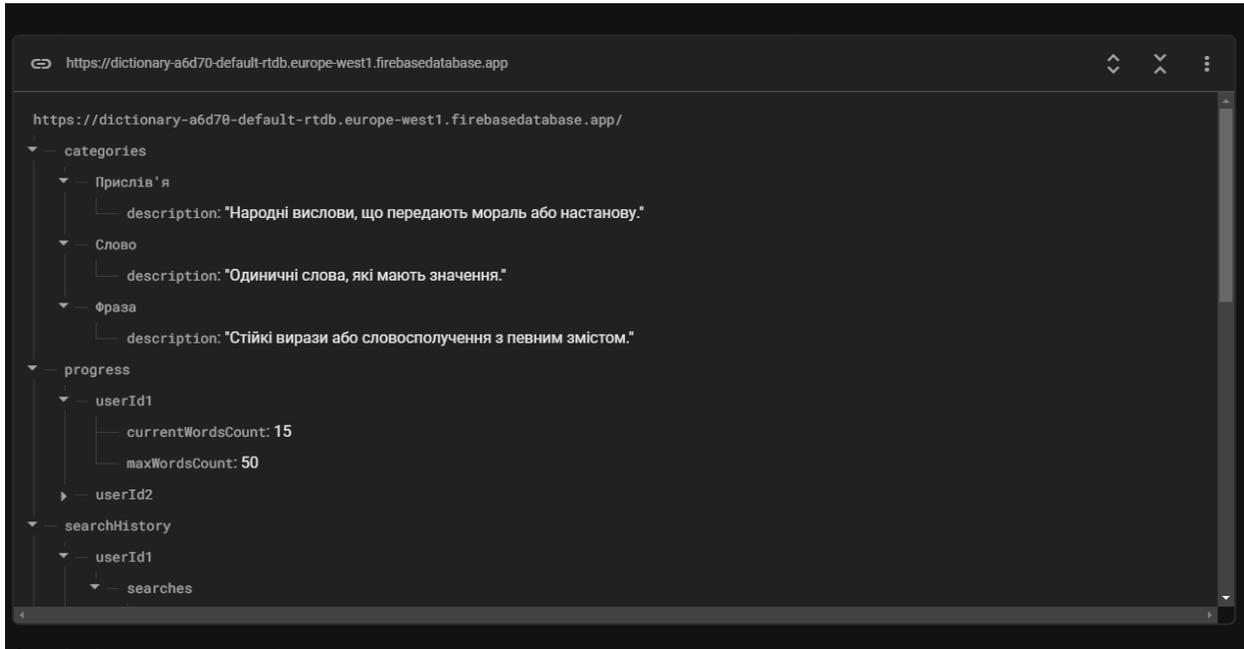


Рис.3.1. База даних

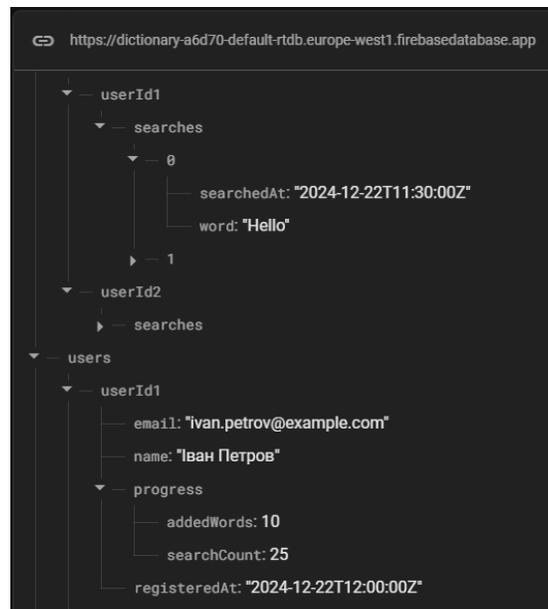


Рис.3.2. Продовження бази даних

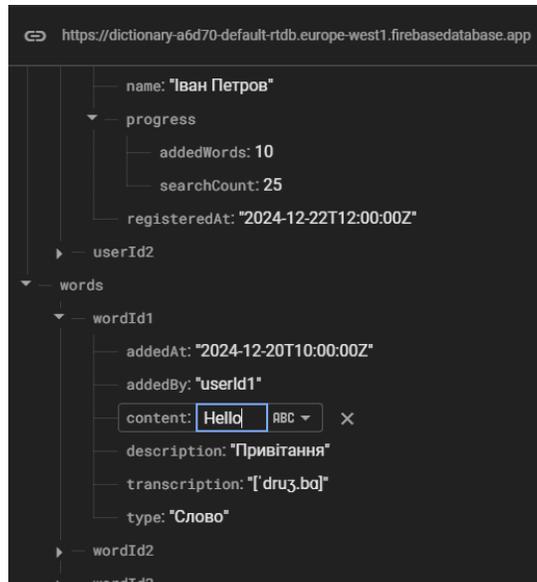


Рис.3.3. Продовження бази даних

Проаналізуємо розроблену базу даних:

1. users:

- Зберігає інформацію про користувачів, включаючи ім'я, email, дату реєстрації.
- Додано розділ progress, який відображає прогрес користувача:
 - addedWords: кількість слів, доданих користувачем.
 - searchCount: кількість виконаних пошуків.

2. words:

- Слова, прислів'я, фрази згруповані за типом через атрибут type.
- Кожен запис містить зміст, транскрипцію, опис, ID користувача.

3. searchHistory:

- Історія пошуків для кожного користувача.

4. categories:

- Описує категорії слів:
 - Слово: звичайні слова.
 - Прислів'я: народна мудрість.

- Фраза: стійкі вирази.
5. progress:
- Відображає загальний прогрес користувача:
 - `currentWordsCount`: поточна кількість слів у базі користувача.
 - `maxWordsCount`: максимальна кількість слів, доступних для додавання.

Ця структура підтримує пошук, додавання нових слів, прогрес, і категорії.

3.2. Кодування в Android Studio

Для реалізації входу в мобільний додаток була використана бібліотека `com.google.firebase.auth.FirebaseAuth`. Це клас із бібліотеки Firebase, який забезпечує основний інтерфейс для роботи з Firebase Authentication. Він дозволяє додаткам виконувати аутентифікацію користувачів за допомогою різних методів: електронна пошта та пароль, Google, Facebook, анонімний вхід та інші. Основні методи цього класу включають `createUserWithEmailAndPassword()` для створення нового користувача, `signInWithEmailAndPassword()` для входу, `signInAnonymously()` для анонімного входу, `signOut()` для виходу та `getCurrentUser()` для отримання інформації про поточного користувача. Він також обробляє події автентифікації, такі як успішний вхід або помилки, через механізм колбеків.

Файл у додаток `LoginActivity.kt` відповідає за обробку вхідної сторінки, де користувач може:

- Увійти за допомогою електронної пошти та пароля.
- Увійти як гість (анонімно).
- Перейти на сторінку реєстрації.

Метод `onCreate()` ініціалізує `Firebase Authentication` та пов'язує з елементами інтерфейсу:

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    supportActionBar?.hide()
    setContentView(R.layout.activity_login)

    // Ініціалізація Firebase
    if (FirebaseApp.getApps(this).isEmpty()) {
        FirebaseApp.initializeApp(this)
    }

    // Ініціалізація FirebaseAuth
    auth = FirebaseAuth.getInstance()

    emailEditText = findViewById(R.id.emailEditText)
    passwordEditText = findViewById(R.id.passwordEditText)
    loginButton = findViewById(R.id.loginButton)
    registerLink = findViewById(R.id.registerLink)
    guestLoginButton = findViewById(R.id.guestLoginButton)

    loginButton.setOnClickListener {
        val email = emailEditText.text.toString()
        val password = passwordEditText.text.toString()

        if (email.isNotEmpty() && password.isNotEmpty()) {
            auth.signInWithEmailAndPassword(email, password)
                .addOnCompleteListener(this) { task ->
                    if (task.isSuccessful) {
                        Toast.makeText(this, "Login
```

```

successful", Toast.LENGTH_SHORT).show()
                val intent = Intent(this,
MainActivity::class.java)
                startActivity(intent)
                finish()
            } else {
                Toast.makeText(this, "Login failed:
${task.exception?.message}", Toast.LENGTH_SHORT).show()
            }
        }
    } else {
        Toast.makeText(this, "Please fill in all
fields", Toast.LENGTH_SHORT).show()
    }
}

registerLink.setOnClickListener {
    val intent = Intent(this,
RegisterActivity::class.java)
    startActivity(intent)
}

guestLoginButton.setOnClickListener {
    val intent = Intent(this, MainActivity::class.java)
    startActivity(intent)
    finish()
}
}
}
}

```

LoginUser зчитує введенні електронну пошту та пароль, перевірає чи поля порожні, використовуючи `.isEmpty()` :

```
if (email.isNotEmpty() && password.isNotEmpty())
```

Далі викликає `auth.signInWithEmailAndPassword(email, password)`, щоб перевірити відповідність даних Firebase. Якщо вхід успішний перенаправляє на головну сторінку. Якщо помилка – виводить повідомлення про помилку:

```
if (task.isSuccessful) {
    Toast.makeText(this, "Login successful",
        Toast.LENGTH_SHORT).show()
    val intent = Intent(this, MainActivity::class.java)
    startActivity(intent)
    finish()
} else {
    Toast.makeText(this, "Login failed:
    ${task.exception?.message}", Toast.LENGTH_SHORT).show()
}
```

GuestLogin перенаправляє на головну сторінку без пароля і логіна.

`openRegister` створює `Intent` для переходу до сторінки реєстрації (`RegisterActivity`):

```
registerLink.setOnClickListener {
    val intent = Intent(this, RegisterActivity::class.java)
    startActivity(intent)
}
```

Похожий код реалізований для файлу реєстрації. Також створено інтерфейс для сторінки входу у файлі `activity_login.xml` в який входить:

- Поле електронної пошти (`EditText`):

Призначене для введення `email`.

- Поле пароля (EditText):

Має тип `textPassword`, щоб приховувати символи.

- Кнопки:

Для входу, входу гостя та переходу на сторінку реєстрації.

Код файла:

```
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white">

    <!-- Фон -->
    <View
        android:id="@+id/backgroundView"
        android:layout_width="0dp"
        android:layout_height="450dp"
        android:layout_marginBottom="2dp"
        android:background="@drawable/rounded_corners"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <!-- Заголовок -->
    <TextView
        android:id="@+id/loginTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Вхід до облікового запису"
```

```
android:textSize="24sp"  
android:textStyle="bold"  
android:layout_marginTop="60dp"  
android:layout_centerHorizontal="true"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent" />
```

```
<!-- Поле ввода email -->
```

```
<EditText  
    android:id="@+id/emailEditText"  
    android:layout_width="300dp"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="80dp"  
    android:hint="Электронна пошта"  
    android:inputType="textEmailAddress"  
    android:padding="10dp"  
    android:background="@drawable/rounded_edittext"  
    app:layout_constraintTop_toBottomOf="@id/loginTitle"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintEnd_toEndOf="parent" />
```

```
<!-- Поле ввода пароля -->
```

```
<EditText  
    android:id="@+id/passwordEditText"  
    android:layout_width="300dp"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="20dp"  
    android:hint="Пароль"  
    android:inputType="textPassword"  
    android:padding="10dp"  
    android:background="@drawable/rounded_edittext"
```

```
app:layout_constraintTop_toBottomOf="@id/emailEditText"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintEnd_toEndOf="parent" />
```

```
<!-- Кнопка для входу -->
```

```
<!-- Кнопка для реєстрації -->
```

```
<Button  
    android:id="@+id/loginButton"  
    android:layout_width="300dp"  
    android:layout_height="50dp"  
    android:layout_marginTop="8dp"  
    android:background="@drawable/round_button"  
    android:text="Увійти"  
    android:textColor="@android:color/white"  
    android:textSize="16sp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.504"  
    app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@id/passwordEditText" />
```

```
<TextView  
    android:id="@+id/registerLink"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Не маєте акаунту? Зареєструйтесь"  
    android:textSize="14sp"  
    android:textColor="@color/brick_red"  
    android:layout_marginTop="20dp"  
    app:layout_constraintTop_toBottomOf="@id/loginButton"  
    app:layout_constraintStart_toStartOf="parent"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        android:gravity="center" />

<!-- Кнопка для входу як гість -->
<Button
    android:id="@+id/guestLoginButton"
    android:layout_width="300dp"
    android:layout_height="50dp"
    android:layout_marginTop="12dp"
    android:background="@drawable/round_button"
    android:text="Вхід як гість"
    android:textColor="@android:color/white"
    android:textSize="16sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.504"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/registerLink"
/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

3.3. Опис інтерфейсу мобільного додатку

Для реалізації мобільного застосунку словника з англійської мови використано прости та сучасний інтерфейс, що включає такі компоненти:

- View - базовий клас усіх елементів інтерфейсу в Android, відповідає за відображення елемента на екрані та взаємодію з користувачем, може бути невидимим для розмітки.

- `CardView` - контейнер, що відображає вміст із закругленими кутами та тінню, використовується для створення карток, наприклад, записів зі словом та його описом, дозволяє налаштовувати тінь, радіус заокруглення та відступи.
- `LinearLayout` - контейнер, що розташовує дочірні елементи в рядок, підтримує атрибути `orientation` та `weight`, який визначає пропорційне розміщення елементів.
- `RelativeLayout` - контейнер, що дозволяє розташовувати елементи відносно один одного або до країв контейнера.
- `TextView` - елемент для відображення тексту.
- `ImageView` - елемент для відображення зображень.
- `ProgressBar` - елемент для відображення процесу завантаження чи виконання завдання.
- `EditText` - елемент для введення тексту користувачем.
- `Bottom` - кнопка для взаємодії з користувачем.
- `BottomNavigationView` - елемент для відображення нижньої панелі навігації, використовується для швидкого доступу до основних функцій додатка, містить кілька вкладок із текстом або іконками, що змінюються при натисканні.

Форма входу зображена на рисунку 3.4.

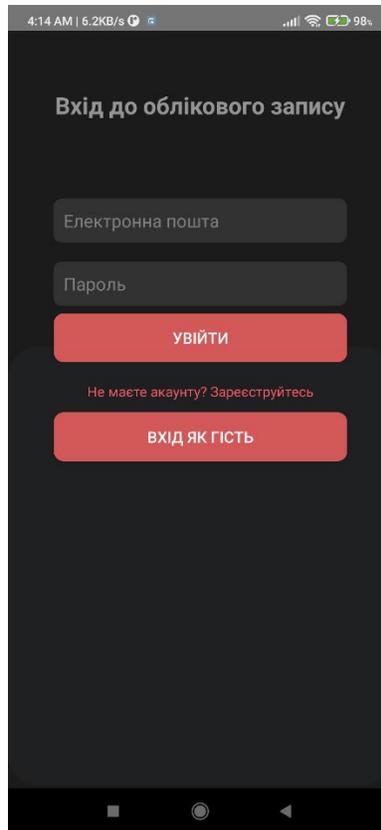


Рис.3.4. Інтерфейс входу у застосунок
Інтерфейс реєстрації показано нижче на рисунку 3.5.

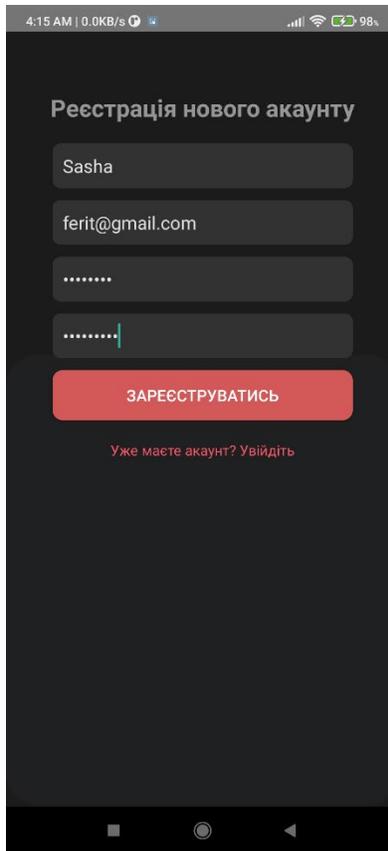


Рис.3.5. Інтерфейс реєстрації у застосунок

Після входу потрапляємо на головну сторінку, яка показана на рисунку

3.6.

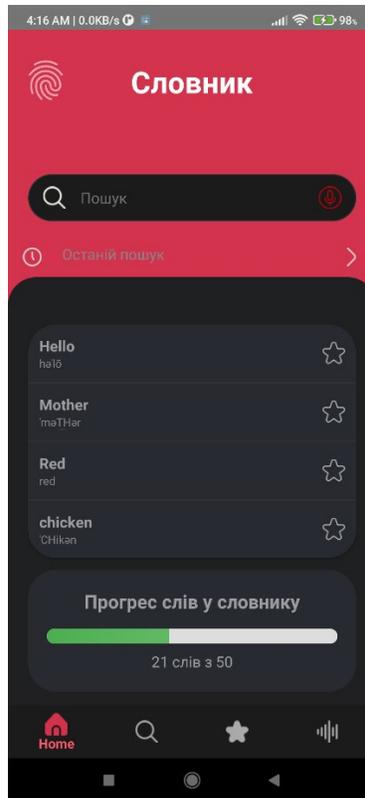


Рис.3.6. Інтерфейс Головної сторінки у застосунок

Щоб знайти потрібне слово є інтерфейс пошуку, який зображено на рисунку 3.7.

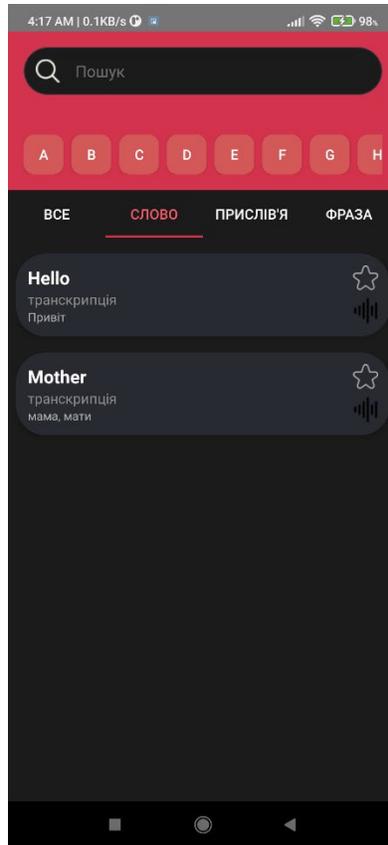


Рис.3.7. Інтерфейс пошуку в застосунок

Інтерфейс про додаткову інформацію слова на рисунку 3.8.

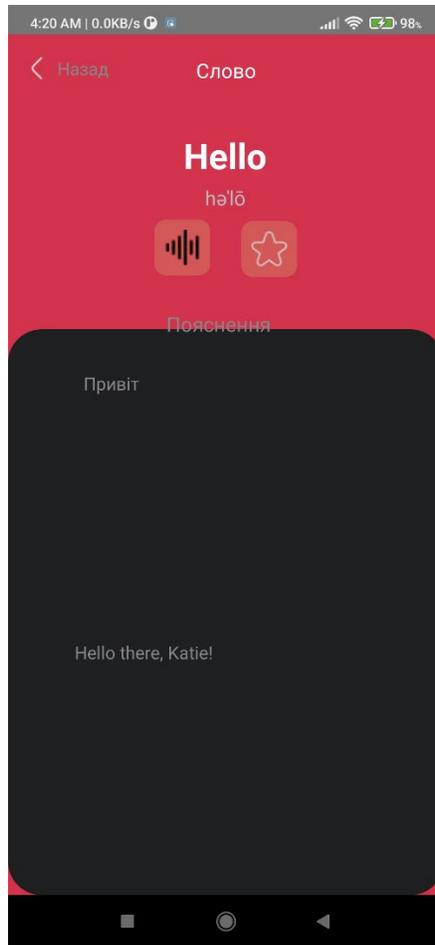


Рис.3.8. Інтерфейс додаткової інформації у застосунок

Для додавання слова у словник створено інтерфейс, який зображений на рисунку 3.9.

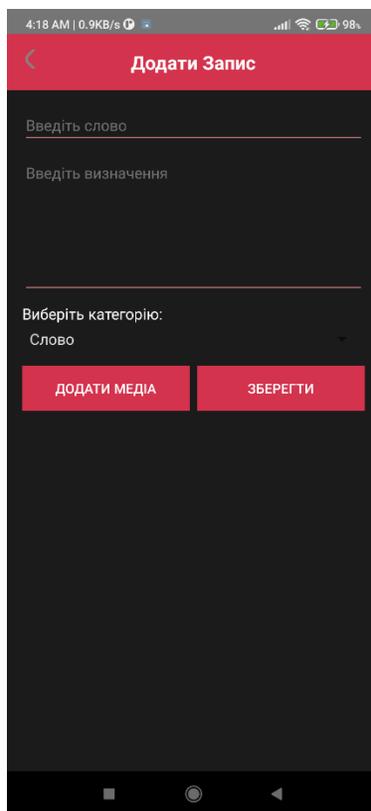


Рис.3.9. Інтерфейс додавання слова у застосунок

3.4. Опис застосунку

Мобільний застосунок словник з англійської мови, створений за допомогою kotlin (Android) з інтеграцією Firebase Realtime Database та Firebase Authentication.

Створена система надає можливості додавати нові записи (слова, прислів'я, фрази) до бази даних, здійснювати пошук записів за категоріями, сортувати за алфавітом та переглядати деталі кожного запису. Для зручності користувачів реалізовано функцію перегляду прогресу навчання, що дозволяє слідкувати за кількістю вивчених слів.

Зареєструватися можуть усі бажаючі за допомогою електронної пошти та пароля. Якщо користувач не зареєстрований, він може натиснути кнопку «Create Account», що відкриє інтерфейс для введення необхідних даних. При успішній реєстрації користувач повертається до сторінки входу і може розпочати роботу в системі. Також передбачена можливість анонімного входу для гостей.

Усі дані синхронізуються з Firebase Realtime Database, що дозволяє швидко оновлювати інформацію та забезпечує доступ до даних.

Застосунок має простий і зручний інтерфейс у використанні.

ВИСНОВКИ

Сучасні мобільні застосунки відіграють важливу роль у навчанні, автоматизації процесів і спрощенні доступу до інформації. У сфері вивчення мов вони допомагають користувачам удосконалювати свої знання завдяки інтерактивним інструментам і зручному інтерфейсу.

Розроблений мобільний застосунок для вивчення англійської мови є словником, створеним для платформи Android. Він надає можливості реєстрації користувачів, пошуку слів, додавання нових записів (слова, прислів'я, фрази), сортування за категоріями та перегляду прогресу у вивченні. Використання Firebase Realtime Database забезпечує надійне збереження даних, їх швидку обробку та синхронізацію між пристроями, а Firebase Authentication дозволяє здійснювати безпечну автентифікацію.

Застосунок має простий та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам швидко адаптуватися. Користувачі можуть взаємодіяти із системою, додаючи слова, переглядаючи їхні транскрипції та описи, а також слідкувати за своїми досягненнями у навчанні.

Для розробки застосунка використано Kotlin та Android Studio, які забезпечують широкий функціонал і зручність створення сучасних мобільних інтерфейсів. Система створена з акцентом на ефективність і доступність, щоб допомогти користувачам у вивченні англійської мови.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Application fundamentals. URL: <https://developer.android.com/guide/components/fundamentals> (дата звернення 02.10.2024)
2. Мобільні додатки як засоби формування іншомовної лексичної компетентності студентів нефалологічних спеціальностей. URL: https://www.researchgate.net/publication/338354758_MOBILNI_DODATKI_AK_ZASOBI_FORMUVANNA_INSOMOVNOI_LEKSICNOI_KOMPETENTNOSTI_STUDENTIV_NEFILOLOGICNIH_SPECIALNOSTEJ (дата звернення 08.10.2024)
3. English as a Global Language. Second edition. Cambridge University Press. Crystal, David. URL: https://www.academia.edu/59223905/Crystal_David_2003_English_as_a_Global_Language_Second_edition_Cambridge_University_Press (дата звернення 10.10.2024)
4. Emerging technologies: Mobile apps for language learning? URL: https://www.researchgate.net/publication/279637329_Emerging_technologies_Mobile_apps_for_language_learning (дата звернення 12.10.2024)
5. UML Distilled. A Brief Guide to the Standard Object Modeling Language URL: https://books.google.com.ua/books/about/UML_Distilled.html?id=nHZslSr1gJAC&redir_esc=y (дата звернення 20.10.2024)
6. Unified Modeling Language User Guide. Addison Wesley. URL: <https://patologia.com.mx/informatica/uug.pdf> (дата звернення 01.11.2024)
7. IBM. What is Java? URL: <https://www.ibm.com/topics/java> (дата звернення 05.11.2024)

8. Geeksforgeeks. Introduction to Java. URL: <https://www.geeksforgeeks.org/introduction-to-java/> (дата звернення 07.11.2024)
9. TECHTARGET. What is Kotlin. URL: <https://www.techtarget.com/whatis/definition/Kotlin> (дата звернення 10.11.2024)
10. DeveloperAndroid. Android's Kotlin-first approach. URL: <https://developer.android.com/kotlin/first> (дата звернення 10.11.2024)
11. Medium. Kotlin vs Java for Android Development: Which One Should You Choose?. URL: <https://medium.com/@kajal.suthar/kotlin-vs-java-for-android-development-which-one-should-you-choose-de015fc11194> (дата звернення 15.11.2024)
12. Mongo. What is NoSQL? URL: <https://www.mongodb.com/resources/basics/databases/nosql-explained> (дата звернення 20.11.2024)
13. Medium. Firebase, Firebase Realtime Database, and Firestore URL: <https://medium.com/@ramadan123sayed/firebase-firebase-realtime-database-and-firestore-8b4fcddb1059> (дата звернення 25.11.2024)

ДОДАТОК А

Логіка головної сторінки MainActivity.kt

```
package com.example.translateapp

import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import
com.google.android.material.bottomnavigation.BottomNavigationVie
w
import android.text.Editable
import android.text.TextWatcher
import android.widget.EditText
import android.widget.ImageView
import android.widget.ProgressBar
import android.widget.TextView
import android.widget.Toast
import androidx.cardview.widget.CardView
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase

class MainActivity : AppCompatActivity() {

    private lateinit var bottomNavigationView:
BottomNavigationView
    private lateinit var database: DatabaseReference

    // Змінні для прогресу
    private var currentWordCount = 20 // Приклад, початкова
кількість слів
```

```
private val maxWordCount = 50 // Максимальна кількість слів  
у словнику
```

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    supportActionBar?.hide()  
    setContentView(R.layout.activity_intro)  
  
    // Ініціалізація Firebase Database  
    database = FirebaseDatabase.getInstance().reference  
  
    // Пошук елементів у макеті  
  
    val searchText: EditText =  
    findViewById(R.id.searchEditText)  
    val microphoneIcon: ImageView =  
    findViewById(R.id.microphoneIcon)  
    val progressBar: ProgressBar =  
    findViewById(R.id.progressBar)  
    val progressText: TextView =  
    findViewById(R.id.progressText)  
    val imageViewArrow: ImageView =  
    findViewById(R.id.imageViewArrow)  
    val cardView: CardView = findViewById(R.id.cardView)  
  
    // Встановлюємо початковий прогрес  
    progressBar.max = maxWordCount  
    progressBar.progress = currentWordCount  
    progressText.text = "$currentWordCount слів з  
$maxWordCount"  
  
    // Логіка для пошуку
```

```

        searchText.addTextChangedListener(object :
TextWatcher {
            override fun beforeTextChanged(s: CharSequence?,
start: Int, count: Int, after: Int) {}
            override fun onTextChanged(s: CharSequence?, start:
Int, before: Int, count: Int) {
                // Логіка пошуку
                Toast.makeText(this@MainActivity, "Пошук:
${s.toString()}", Toast.LENGTH_SHORT).show()
            }
            override fun afterTextChanged(s: Editable?) {}
        })

// Логіка для мікрофона
microphoneIcon.setOnClickListener {
    Toast.makeText(this, "Мікрофон натиснутий",
Toast.LENGTH_SHORT).show()
}

// Логіка для натискання на ImageView і перехід на іншу
сторінку
imageViewArrow.setOnClickListener {
    // Створюємо Intent для переходу до нової активності
    val intent = Intent(this,
SearchActivity::class.java)
    startActivity(intent)
}

// Логіка для додавання нового слова для бази даних
addNewWord()

// Логіка для натискання на елемент в CardView

```

```
        cardView.setOnClickListener {
            val intent = Intent(this,
WordDetailActivity::class.java)
            startActivity(intent)
        }

        bottomNavigationView =
findViewById(R.id.bottom_navigation)

        bottomNavigationView.setOnItemSelectedListener { item ->
            when (item.itemId) {
                R.id.nav_home -> {
                    // Логіка для головної сторінки
                    true
                }
                R.id.nav_record -> {
                    // Перехід до AddRecordActivity
                    val intent = Intent(this,
AddRecordActivity::class.java)
                    startActivity(intent)
                    true
                }
                R.id.nav_search -> {
                    val intent = Intent(this,
SearchActivity::class.java)
                    startActivity(intent)
                    true
                }

                R.id.nav_saved -> {
                    val intent = Intent(this,
DictionaryActivity::class.java)
```



```
}  
  
// Функція для збереження слова в Firebase  
private fun saveWordToDatabase(word: String) {  
    val wordRef = database.child("words").push() // Додаємо  
нове слово в базу  
    wordRef.setValue(word)  
}  
}
```