

## ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему:

«Інтерактивний застосунок-тренажер для вивчення англomовних слів»

Галузь знань 12 «Інформаційні технології»

Спеціальність 122 «Комп'ютерні науки»

Освітня програма «Комп'ютерні науки»

Освітній рівень: бакалавр

Виконала: студентка 4 курсу, групи КН- 42



Саєнко К.Ю.

(прізвище та ініціали)



Керівник Красовська Г. В.

(прізвище та ініціали)

канд. техн. наук, доц.

(науковий ступінь, звання)

Випускна кваліфікаційна робота бакалавра допущена до захисту  
рішенням кафедри *інтелектуальних технологій*  
Протокол № 11 від 06.06.2022 р.  
Зав. кафедри \_\_\_\_\_ доц. Іларіонов О.Є.

Київ – 2022

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра інтелектуальних технологій

Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
інтелектуальних технологій  
Іларіонов О.Є.

“ \_\_\_ ” \_\_\_\_\_ 2022 р.

**ЗАВДАННЯ  
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

\_\_\_\_\_ Саєнко\_Катерині Юріївні \_\_\_\_\_


(прізвище, ім'я, по батькові)


1. Тема проекту (роботи) “Інтерактивний застосунок-тренажер для вивчення англomовних слів” затверджена протоколом засідання кафедри інтелектуальних технологій від 23 грудня 2021 р. № 4
2. Термін здачі студентом закінченого проекту (роботи) 29 травня 2022 року
3. Вихідні дані до проекту (роботи)
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)  
Вступ, системний аналіз інтерактивного застосунку-тренажеру для вивчення англomовних слів, проектування інтерактивного застосунку-тренажеру для вивчення англomовних слів, розробка інтерактивного застосунку-тренажеру для вивчення англomовних слів, висновок.
5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)
  1. Актуальність теми (1 слайд);
  2. Мета, предмет та об'єкт дослідження (1 слайд);
  3. Системний аналіз інтерактивного застосунку-тренажеру для вивчення англomовних слів (7 слайдів)
  4. Проектування інтерактивного застосунку-тренажеру для вивчення англomовних слів (7 слайдів)
  5. Розробка інтерактивного застосунку-тренажеру для вивчення англomовних слів (4 слайди)
  6. Перевірка працездатності застосунку-тренажеру (5 слайдів)
7. Висновки по роботі (1 слайд)

6. Консультанти з випускної кваліфікаційної роботи із зазначенням розділів випускної кваліфікаційної роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 15 лютого 2022 року

Керівник  / Красовська Г.В. /  
(ПІБ)

Завдання прийняв до виконання  / Саєнко К.Ю. /  
(ПІБ)

### КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
1	Дослідження джерел та збір інформації для випускної кваліфікаційної роботи.	15.02.2022 - 1.03.2022	
2	Системний аналіз інтерактивного застосунку-тренажеру для вивчення англомовних слів.	1.03.2022 - 15.03.2022	
3	Проектування інтерактивного застосунку-тренажеру для вивчення англомовних слів.	15.03.2022 - 05.04.2022	
4	Розробка інтерактивного застосунку-тренажеру для вивчення англомовних слів.	05.04.2022 - 09.05.2022	
5	Налагодження та тестування застосунку-тренажеру.	09.05.2022 - 16.05.2022	
6	Оформлення пояснювальної записки та підготовка презентаційних матеріалів.	16.05.2022 - 29.05.2022	

Студент-дипломник  / Саєнко К.Ю. /  
(ПІБ)

Керівник випускної кваліфікаційної роботи  / Красовська Г.В. /  
(ПІБ)

Анотація

Саєнко Катерина Юріївна виконала випускню кваліфікаційну роботу на тему «Інтерактивний застосунок-тренажер для вивчення англомовних слів» за спеціальністю 122 «Комп'ютерні науки».

У випускній кваліфікаційній роботі було проведено аналіз існуючих застосунків-тренажерів, створено інтерактивний застосунок-тренажер, за допомогою якого можливо якісно вивчати слова англійської мови.

Ключові слова: застосунок-тренажер, користувач, англомовні слова, картки.

### Summary

Saienko Kateryna Yuriyivna completed the final qualifying work on the topic "Interactive application-simulator for learning English words" in the specialty 122 - "Computer Science".

In the final qualifying work, an analysis of existing simulator applications was conducted, and an interactive simulator application was created, with the help of which it is possible to learn English words well.

Keywords: simulator application, user, English words, cards.

ВСТУП.....	6
РОЗДІЛ 1. АНАЛІЗ РОЗРОБКИ ІНТЕРАКТИВНОГО ЗАСТОСУНКУ – ТРЕНАЖЕРУ ДЛЯ ВИВЧЕННЯ АНГЛОМОВНИХ СЛІВ.....	6
1.1 Актуальність розробки інтерактивного застосунку – тренажеру для вивчення англomовних слів.....	6
1.2 Аналіз існуючих тренажерів-додатків для вивчення англійської мови.....	7
1.2.1 Lingualeo.....	8
1.2.2 Duolingo.....	9
1.2.3 EWA.....	10
1.2.4 Busuu.....	11
1.2.5 Memrise.....	12
1.3 Вимоги до застосунку – тренажеру для вивчення англomовних слів.....	14
РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ ІНТЕРАКТИВНОГО ЗАСТОСУНКУ – ТРЕНАЖЕРУ ДЛЯ ВИВЧЕННЯ АНГЛОМОВНИХ СЛІВ.....	17
2.1 Розробка архітектури інтерактивного застосунку – тренажеру для вивчення англomовних слів.....	17
2.1.1 Функціональний аналіз застосунку тренажеру.....	17
2.1.2 IDEF0 процесу виведення карток слів для вивчення англійської мови.....	20
2.1.3 Архітектура інформаційної системи.....	22
2.2 Розробка інформаційного забезпечення застосунку – тренажеру.....	22
2.3 Математична постановка основної задачі інтерактивного застосунку – тренажеру для вивчення англomовних слів.....	27
РОЗДІЛ 3. РОЗРОБКА ІНТЕРАКТИВНОГО ЗАСТОСУНКУ – ТРЕНАЖЕРУ ДЛЯ ВИВЧЕННЯ АНГЛОМОВНИХ СЛІВ.....	28
3.1 Вибір мови розробки.....	28
3.2 Опис основних програмних компонентів.....	29
3.3 Розробка та опис програмного інтерфейсу з користувачем (керівництво користувача) інтерактивного застосунку-додатку для вивчення англomовних слів.....	30
3.4 Перевірка працездатності програмного рекомендаційного модуля.....	36
ВИСНОВКИ.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47

## ВСТУП

Роль англійської мови в теперішньому світі дійсно дуже велика. Англійська мова є міжнародною, саме її в основному використовують для спілкування людей різних націй. Турки та фіни, чехи та угорці, італійці та українці, не знаючи мови один одного, можуть спілкуватися між собою цією мовою. Вона була визначена мовою для ведення переговорів у таких міжнародних організаціях як Генеральна Асамблея ООН, Союз Безпеки Європи, ЮНЕСКО, НАТО, Європейський Союз та організація країн-експортерів нафти (ОПЕК). Кожен світовий акт, технічна публікація, книга, інструкція, пісня, плакат, лист будуть прочитані та зрозумілі різними націями та народами, якщо їх викладають англійською мовою. Саме тому проблема вивчення англійської мови є актуальною для багатьох людей у світі.

Метою дипломної роботи є створення застосунку-тренажеру для вивчення англійських слів, який буде здатний полегшити вивчення англійської мови для всіх бажаючих.

Об'єктом дослідження є процес вивчення та перевірки знань у застосунках-тренажерах для вивчення англійської мови.

Предметом дослідження є методи перевірки та оцінювання знань при вивченні англійської мови.

У першому розділі було досліджено актуальність вивчення англійської мови, проведено аналіз існуючих застосунків-тренажерів та визначено основні функціональні та нефункціональні вимоги до застосунку.

У другому розділі була розроблена архітектура застосунку-тренажеру, побудовано діаграми функціонування, діаграми декомпозиції основних функцій та визначено математичну постановку основної задачі інтерактивного застосунку – тренажеру для вивчення англійських слів.

У третьому розділі описано вибір мови програмування, розроблено та описано програмний інтерфейс, а також перевірено працездатність застосунку-тренажеру.

## РОЗДІЛ 1. АНАЛІЗ РОЗРОБКИ ІНТЕРАКТИВНОГО ЗАСТОСУНКУ – ТРЕНАЖЕРУ ДЛЯ ВИВЧЕННЯ АНГЛОМОВНИХ СЛІВ

### 1.1 Актуальність розробки інтерактивного застосунку – тренажеру для вивчення англомовних слів

Англійська мова настільки популярна, що стала стандартною мовою міжнародного спілкування. В даний час 75% світової пошти англійською, 60% радіопрограм транслюються англійською, більш ніж половина періодичних видань світу друкуються англійською мовою.

Таке поширення англійської мови по всьому світу відбувається через те, що Великобританія була і є світовою морською державою. Вона мала колонії, а потім і домініони у всьому світі. Тому багато країн в даний час використовують англійську мову нарівні зі своєю державною мовою: Індія, Ірландія, Канада, Нова Зеландія, Сінгапур, ПАР, Федеративні Штати Мікронезії та інші. Деякі країни обирають цю мову як державну: США, Антигуа і Багами, Барбадос, Гана, Домініканська Республіка, Нігерія, Соломонові острови, Тринідад і Тобаго, Ямайка та інші. Англійська мова має статус офіційної або другої офіційної в 45 країнах світу.

Зі статистичних даних відомо, що не менше 1100000000 людей у світі розмовляють англійською мовою, і кількість бажаючих вивчити її постійно зростає, адже це найважливіша мова для медицини, авіації, бізнесу та світу інформаційних технологій. Кожен програміст, який планує розвивати свої навички та підтримувати їх у актуальному стані, повинен як мінімум розуміти англійську мову і наведу кілька причин, чому це так:

- Технічна документація: вся технічна документація для всіх мов програмування складається англійською мовою. Наприклад, мову Ruby винайшли в Японії, а документацію пишуть англійською.
- Професійна література: 60% технічно-наукових публікацій робляться англійською мовою.
- Міжнародні ІТ-компанії: більшість ІТ-компаній мають офіси у кількох країнах. Для комунікації між ними, знову ж, найчастіше обирають англійську мову.

Тому, тема швидкого та якісного вивчення англійської мови досить актуальна для кожного, незалежно від міста проживання, роботи та сфери інтересів.

Наразі існує багато способів вивчення іноземної мови, найчастіше люди обирають проходження групових курсів, індивідуальні заняття з репетитором, самостійне вивчення по книжкам і т.д. Проте, з розвитком ІТ-технологій з'явився ще один, наразі досить популярний, спосіб, - це тренажери-додатки.

## 1.2 Аналіз існуючих тренажерів-додатків для вивчення англійської мови

Розроблено багато додатків для вивчення англійської мови. Користувачів приваблює використання тренажерів-додатків так як це досить зручно, адже маючи при собі телефон можна у будь-якому місці присвятити час навчанню. Також більшість додатків допомагають вирішити одну з основних проблем, яка виникає при вивченні мови – вони створюють зовнішню мотивацію. Цьому зазвичай сприяє ігрова техніка вирішення завдань. Всі програми створені для людей з різним знанням англійської мови. Ви можете встановлювати рівень складності, вивчати мову у зручному для вас режимі та у вільний час.

Більшість додатків пропонують користувачу спочатку пройти тест, який визначає рівень знань, а потім пропонують матеріал для вивчення, відповідно до

рівня знань . Це може бути теоретичний матеріал, тести, аудіювання, вивчення слів, читання та письмо. Проте, через певний час, матеріали, зазвичай, стають платними. У наступному розділі проведемо розбір аналогів-додатків по вивченню іноземної мови.

До найпопулярніших додатків відносяться:

- Lingualeo
- Duolingo
- EWA
- Busuu
- Memrise

### 1.2.1 Lingualeo

Lingualeo вважається найпопулярнішим додатком для вивчення мови та нараховує близько 9 мільйонів користувачів. Вивчення англійської мови розроблено таким чином, щоб забезпечити індивідуальний підхід до кожного користувача. На підставі вхідного тестування, програма Lingualeo визначає курс навчання і щодня пропонує учневі необхідну для вивчення кількість слів та граматичного матеріалу.

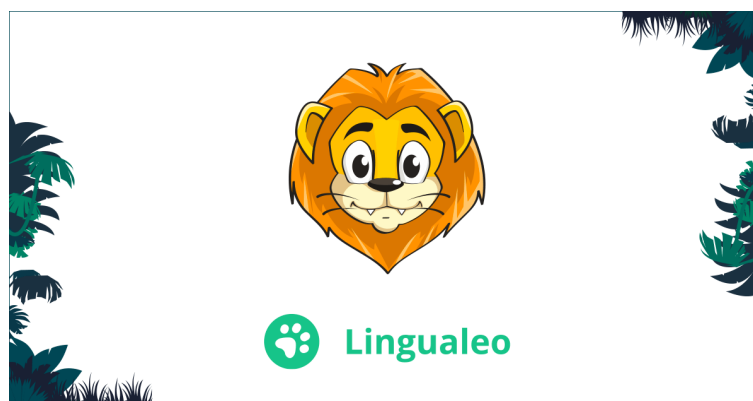


Рисунок 1.1. Заставка додатку Lingualeo

Переваги:

- LinguaLeo має зручний інтерфейс, який зможуть зрозуміти діти;
- великий вибір навчальних матеріалів;
- система заохочення за пройдений матеріал;
- чіткі інструкції: «вчимо це...», «дивимося це...»;
- кілька варіантів тренувань: читання, писання, слухання, говоріння;

Недоліки:

- багато функцій є платними;
- немає можливості вибору того, що вчити та пропустити матеріал, який вже знаєш;
- мінімальна кількість матеріалів для учнів, чий рівень англійської вищий за середній;
- немає форуму або чатів для підтримки рівня англійської мови у спілкуванні з іншими користувачами;

### 1.2.2 Duolingo

Додадок досить схожий на попередній та налічує близько 150 мільйонів користувачів. Сервіс розроблений так, що в міру проходження уроків користувачі паралельно допомагають перекладати веб-сайти, статті та інші документи. Дослідження, проведене професорами Міського університету Нью-Йорка та Університету Південної Кароліни, показало, що 34 години на Дуолінго дають стільки ж навичок читання та письма, скільки дає початковий семестровий курс в американському вищому навчальному закладі, що займає близько 130 годин. Дослідження не вимірювало розмовних навичок. У процесі дослідження 108 зі 196 осіб, які вивчають мову на Дуолінго, кинули заняття менш ніж за дві години.

Дуолінго надає функції "Дуолінго для шкіл", що дозволяють вчителям відстежувати своїх учнів.



Рисунок 1.2 - Заставка додатку Duolingo

Переваги:

- щотижневі змагання для підтримки мотивації;
- уроки для різного рівня знань;
- кожен урок супроводжується теорією;
- можливість вивчати інші мови, крім англійської;

Недоліки:

- наявність великої кількості реклами;
- користувачі відзначають велику кількість багів;
- згодом, деякі функції додатку стають платними;
- недостатня кількість пояснень до уроків;

### 1.2.3 EWA

Додаток для навчання мови за 3 рівнями – від початкової до просунутої. Окремо Ewa пропонує курс англійської для бізнесу та подорожей. Додаток має в своєму розпорядженні медіатеку з книг, фільмів та аудіо.

Програма EWA не навчить вас граматиці і не розповість, як правильно писати складні пропозиції, не навчить говорити правильно і з інтонацією. Але, програма у дуже зручному вигляді дозволяє вам вивчати англійську мову та збільшувати кількість слів, які ви знаєте. Це велика перевага, адже якщо припинити розвивати свій словниковий запас, то рано чи пізно ви почнете слова забувати.

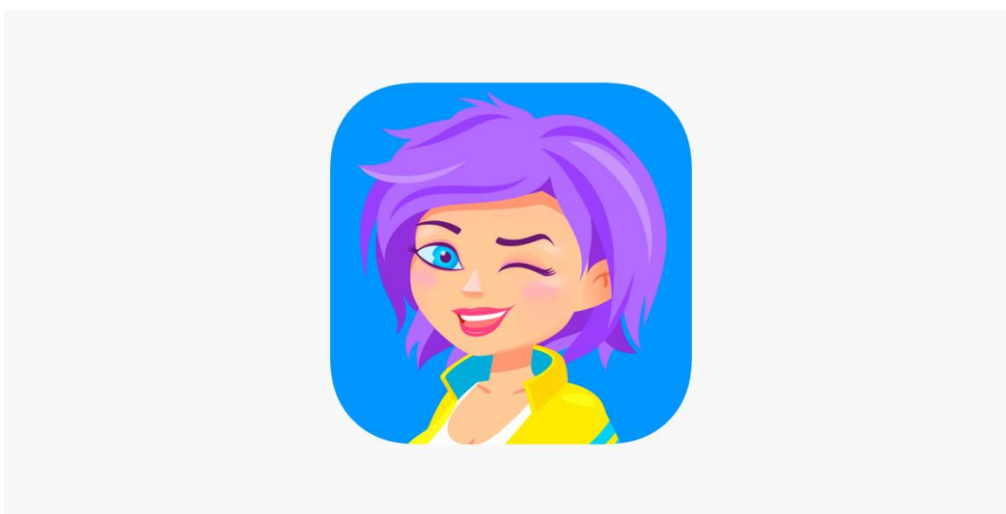


Рисунок 1.3 - Заставка додатку EWA

Переваги:

- підходить для вивчення мови з 0;
- ілюстрація матеріалу з фільмів та серіалів;
- існує секція з книгами;

Недоліки:

- велика кількість реклами;
- іноді існує невідповідність матеріалу темам;
- безкоштовний пробний період триває лише 5 днів. Вартість місячної підписки – 200 грн, річної – 500 грн;

#### 1.2.4 Busuu

Busuu - це інтерактивний навчальний сервіс, де ви можете вивчати онлайн не тільки англійську, але й інші іноземні мови: іспанську, італійську, французьку, турецьку, арабську, португальську, німецьку, російську, японську та китайську. Сервіс є списком рівневих тематичних курсів з короткими завданнями на теорію і практику.

Навчання мови здійснюється за принципом "10 хвилин на день". Навчальний графік задаєте самі, а рівень складності визначається за попереднім тестом.



Рисунок 1.4 - Заставка додатку Bussu

Переваги:

- є тренажер для вивчення слів;
- можливість проходження тестів для перевірки знань;
- якісна градація по рівням;
- є оффлайн доступ до додатку;

Недоліки:

- безкоштовне використання 7 днів;

- підходить тільки для початкового рівня;
- мало можливостей для тренування вимови та мови;
- користувачі скаржуються на велику кількість багів;

### 1.2.5 Memrise

Додаток Memrise орієнтований на вивчення слів. Програма слів підійде швидше користувачам з рівнем не вище Intermediate. Вашому вибору пропонується 2 рівні складності - для початківців та просунутих. Принцип вивчення мови простий: прослуховуємо слова з відео та аудіоподкастів, а потім повторюємо їх, набираючи на віртуальній клавіатурі або натискаючи на картки. Налічує більше 50 мільйонів користувачів.

При створенні нового облікового запису користувачеві пропонується вибрати мову для вивчення та рівень знань. Після цього користувач потрапляє на сторінку, де запропонують пройти невеликий урок з урахуванням його знання мови. В основному всі уроки складаються з вивчення карток слів (вимова, написання та переклад), слово може озвучити як носій мови, демонструватися відео де воно фігурує, так і просто бути написане текстом.



Рисунок 1.5 - Заставка додатку Memrise

Переваги:

- можливість користування оффлайн;
- підходить для початкового рівня володіння мовою;
- можливість вивчення кількох мов;
- акцент на запам'ятовування слів;

Недоліки:

- у безкоштовній версії кількість уроків обмежена;
- користувачі відмічають незручний інтерфейс;
- іноді показує неправильний переклад слів;

Для наведених додатків можна виділити такі спільні функціональні риси:

- ✓ наявність тесту для визначення рівня знань;
- ✓ наявність матеріалу для різних рівнів володіння мовою;
- ✓ можливість вивчення інших мов, крім англійської;

Спільні нефункціональні риси:

- ✓ можливість використання оффлайн;
- ✓ наявність реклами;
- ✓ короткий безкоштовний період використання;

Таблиця 1.1 - Порівняння існуючих тренажерів-додатків

	Lingualeo	Duolingo	EW A	Busuu	Memrise
1. Зручний інтерфейс	+	+	-	-	-
2. Початковий тест	+	+	-	+	-
3. Підходить для різних рівнів	+	+	+		+
4. Можливість оффлайн використання	-	+	+	+	+
5. Наявність реклами	+	+	+	+	+
6. Можливість вивчення інших мов	+	+	-	-	-
7. Наявність теоретичного матеріалу	+	-	-	+	-

8. Тренажер для вивчення слів	-	-	+	+	+
9. Ілюстрація матеріалу	-	+	+	-	-
10. Безкоштовність використання	-	-	-	-	-
11. Перевірка знань	+	-	+	-	-
12. Можливість додавати матеріал	-	-	-	-	+

Вище наведено таблицю порівнянь існуючих аналогів застосунків-тренажерів та основні вимоги користувачів до них. На основі цього можна зробити висновок, що кожен аналог відповідає вимогам приблизно на 50-60%, тому, реалізуючи ці вимоги у розроблюваному застосунку-тренажеру можна зробити застосунок кращим, ніж вже існуючі аналоги.

### 1.3 Вимоги до застосунку – тренажеру для вивчення англомовних слів

Проаналізувавши різні тренажери додатки та дослідивши відгуки користувачів можна зробити висновок про основні вимоги до застосунку:

- ✓ ефективна методика навчання;
- ✓ зручний та простий інтерфейс;
- ✓ безкоштовне використання;
- ✓ відсутність реклами;
- ✓ достовірність інформації;
- ✓ можливість самостійно додавати матеріал для вивчення;
- ✓ можливість використанні оффлайн.

Функціональні вимоги:

- наявна база карток з англійськими словами та перекладом до них;
- створення власних карток;
- створення збірок карток;
- можливість реєстрації та авторизації.

Розглянемо детальніше деякі пункти:

1) Ефективна методика навчання.

При розробці власного застосунку важливим був вибір правильної методики для вивчення іноземної мови. Мною була обрана методика «5000 слів», детальніше про неї:

Сучасні мови дотримуються принципу Парето - 20% найбільш поширених слів використовуються в 80% випадків. Відповідно до різних досліджень, вам потрібно розуміти від 95 до 98% мови, щоб досягти рівня, на якому ви не тільки можете зрозуміти майже все, але й можете вгадати значення слів, яких ви не знаєте без своєї рідної мови.

На основі даних Оксфордського університету досліджено, що 5000 найпоширеніших слів у розмовній англійській мові охоплюють 96% розмовної мови. Якщо ми помістимо це в контекст CEFR (визнаний у всьому світі стандарт для опису рівнів володіння іноземною мовою) то отримаємо:

Рівень CEFR	Кількість слів	Накоплений словарний запас	Покриття
A1	600 слів	600 слів	85%
A2	1000 слів	1600 слів	91%
B1	1100 слів	2700 слів	94%
B2	1000 слів	3700 слів	95%
C1	1000 слів	4700 слів	96.13%
C2	300 слів	5000 слів	96.30%

Отже, 5000 найпоширеніших слів у розмовній англійській охоплюють 96% розмовної мови. Використовуючи математичні розрахунки, легко порахувати, що якщо вчити по 20 слів кожного дня та періодично їх повторювати – достатньо менше року, щоб вивчити найпоширеніші слова англійської мови.

## 2) Зручний та простий інтерфейс.

Розроблений інтерфейс має бути максимально простим, щоб його могли легко розуміти діти та не досвідчені користувачі телефоном.

## 3) Достовірність інформації .

Так як, із зовнішньої інформації необхідний тільки переклад слів, буде створено базу даних із 5000 найпопулярнішими словами англійської мови та взято їх переклад з Оксфордського словника, який є головним історичним словником англійської, що видається видавництвом Оксфордського університету.

## 4) Можливість самостійно додавати матеріал для вивчення.

У додатку буде можливість самостійно додавати необхідні слова для вивчення та створювати колекції (категорії) слів.

## 5) Можливість використання оффлайн.

Дуже часто вільний час для вивчення чогось з'являється тоді, коли відсутній інтернет, наприклад у літаку, у потязі і так далі. Тому, важливо, щоб була можливість використання оффлайн.

## РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ ІНТЕРАКТИВНОГО ЗАСТОСУНКУ – ТРЕНАЖЕРУ ДЛЯ ВИВЧЕННЯ АНГЛОМОВНИХ СЛІВ

### 2.1 Розробка архітектури інтерактивного застосунку – тренажеру для вивчення англомовних слів

#### 2.1.1 Функціональний аналіз застосунку тренажеру

Побудуємо дерево функцій розроблювальної системи. (рис. 2.1). Вершиною дерева функцій є функціональний модуль вивчення англійських слів через додаток. Функції даного застосунку діляться на клієнтські та адміністративні.

До клієнтських функцій системи входять реєстрація, пошук нових збірників та ведення акаунту. Реєстрація - функція яка дозволяє відрізнити авторизованих користувачів та гостей. Також вона передбачає можливість створити новий профіль користувача. Редагування профіля - його ведення містить функції для додавання нових збірників карток з словами, редагування картки слів та створювати новий збірник слів. Пошук готових збірників дозволяє переглядати слова за темами які існують, наприклад, “Їжа”, “Математика” та інші.

До адміністративних функцій відноситься робота з контентом. Адміністратор має функції редагування існуючих слів та збірників які вже існують. Створювати нові картки та збірники для всіх користувачів.

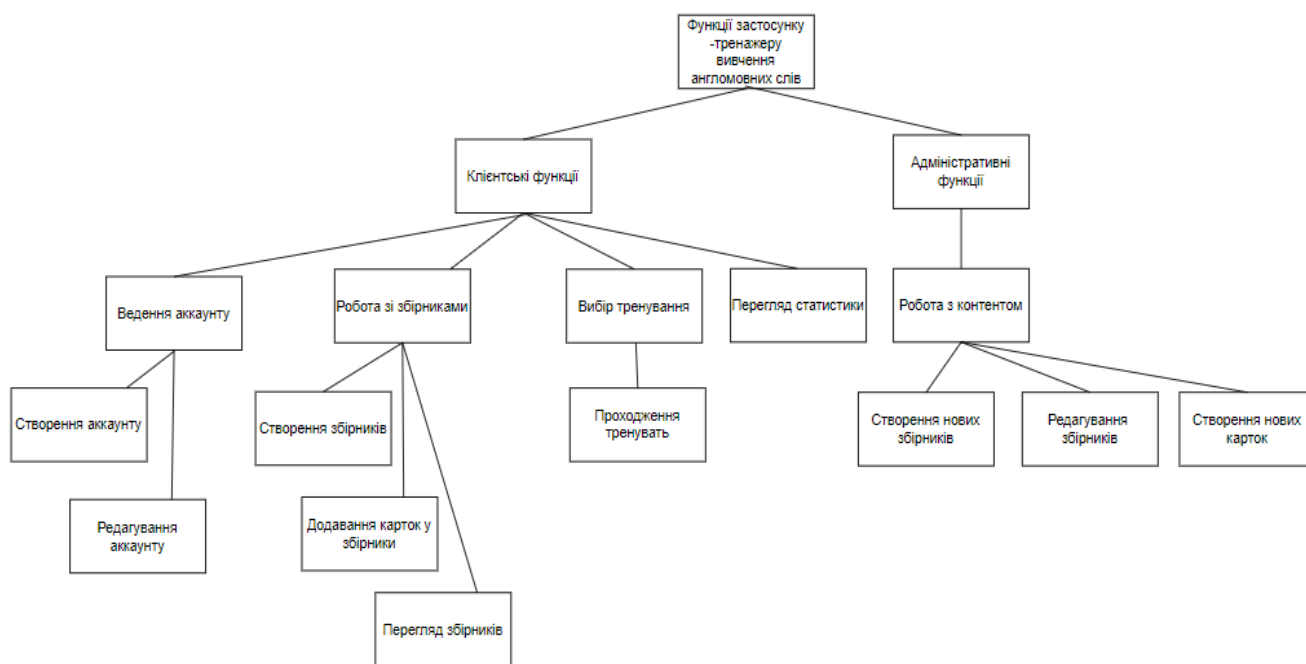


Рисунок 2.1 – “Дерево функцій” застосунку-тренажеру

Робота системи починається тоді, коли користувач відкриває завантажений додаток на своєму пристрої. Далі користувачу на вибір пропонується дві функції, для нових користувачів - реєстрація, після проходження якої йде запис про нового

користувача база даних користувачів, після реєстрація новий користувач може зайти до особистого кабінету так само як і старі користувачі. Користувачу після авторизації в додатку відкривається можливість відвідати сторінку збірників, а саме бібліотеку карток. Користувачу відкривається дві можливості на вибір, обрати особисту збірку слів, яку він може доповнювати та редагувати самостійно, або відкрити вже готові збірки карток які були сформовані адміністратором сайту та записані до особистої бази даних додатку. Додаток після всіх подій відображає користувачу всю необхідну інформацію в вигляді карток на яких записані слова.

Побудуємо Value Added Chain Diagram. На найвищому рівні знаходиться процес «Робота з додатком англійської мови». Його підпроцесами є «Реєстрація у додатку» та «Користування додатком вивчення» (відношення частина - ціле). Перший має також підпроцеси – «Введення основних даних» та «Підтвердження реєстрації». Між цими підпроцесами зв'язком є відношення попередній процес - наступний процес, тобто дії виконуються послідовно.

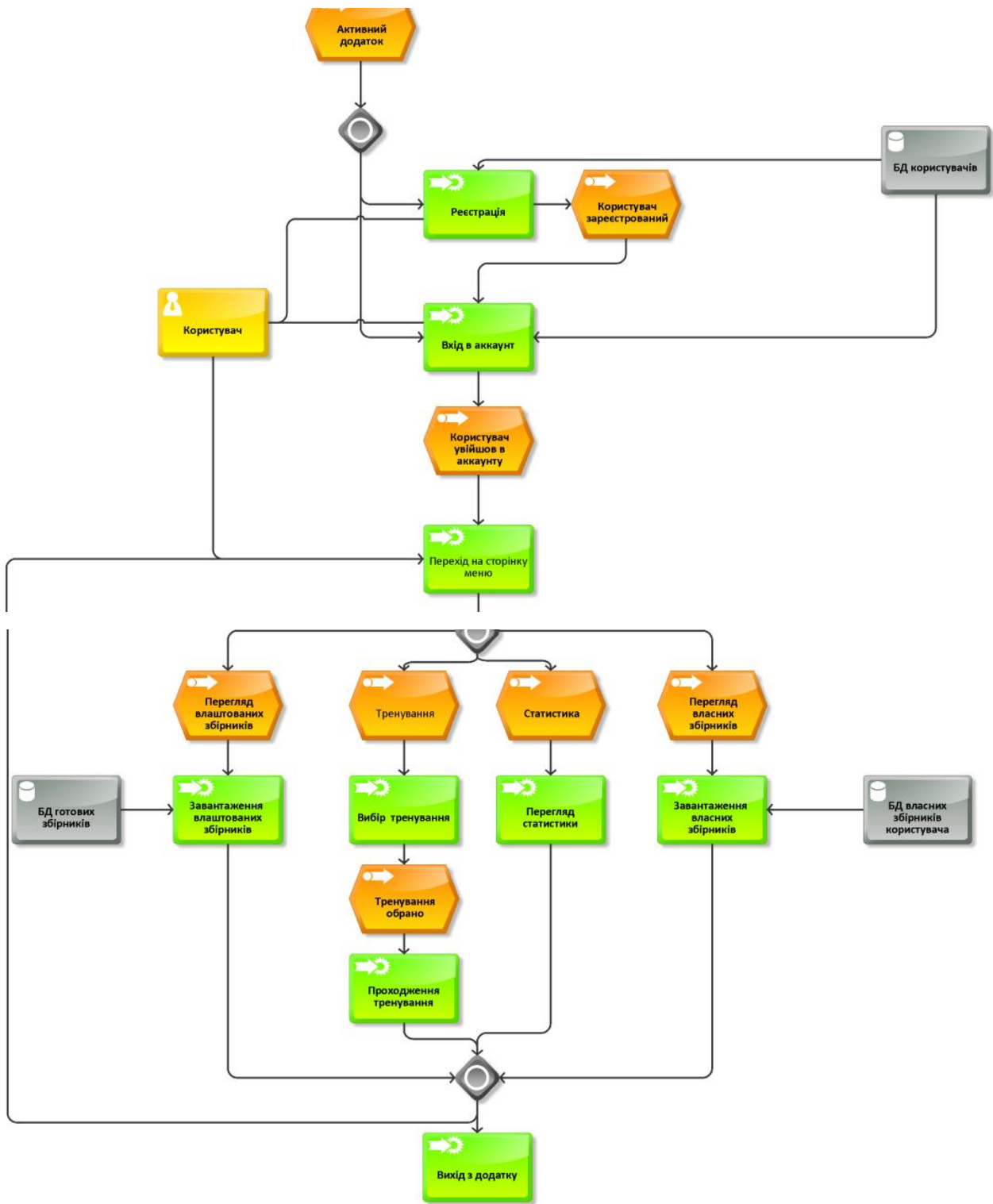


Рисунок 2.2 – Діаграма Event-Driven Process Chain для функціонування клієнтської частини застосунку-тренажера

Аналогічним чином побудований другий підпроцес «Користування додатком вивчення». Його підпроцесами є «Зміна особистого профілю», «Створення особистих збірників карток» та «Отримання готових збірників карток». Перший містить процеси редагування особистого профілю, другий має основні функції редагування додавання та зміна особистих збірників карток та останній дозволяє отримати готовий збірник карток.



Рисунок 2.3 – Діаграма Value Added Chain Diagram для функціонування застосунку-тренажеру

### 2.1.2 IDEF0 процесу виведення карток слів для вивчення англійської мови

IDEF0 – методологія функціонального моделювання та графічна нотація, призначена для формалізації та опису бізнес-процесів. Відмінною особливістю IDEF0 є акцент на супідрядність об'єктів. У IDEF0 розглядаються логічні відносини між роботами, а не їх тимчасова послідовність (потік робіт).

Проаналізуємо побудову контекстної діаграми. На вході система отримує дані користувача за якими він зареєструвався.

Елементом керування є закон України про «Захист персональних даних», тому що на кожному етапі зберігаються особисті дані користувача: при авторизації - це логін та пароль та Оксфордський словник, який містить слова та їх переклад.

Механізмами у даному випадку є користувач системи, та ІС, створена розробником додатку.

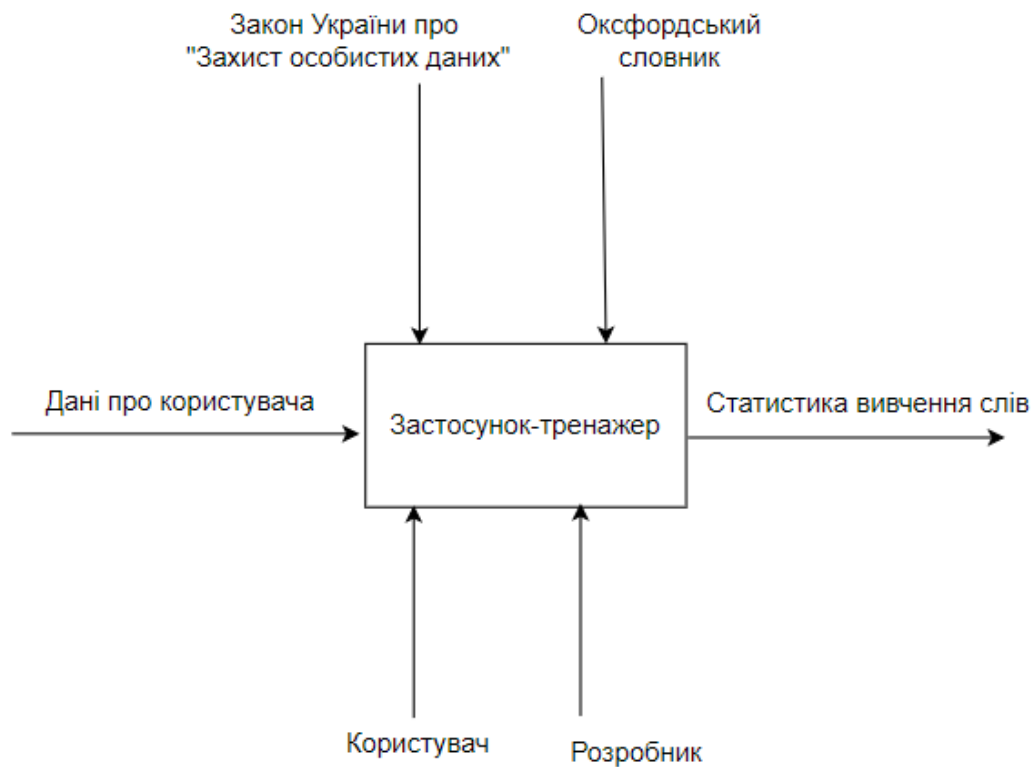


Рисунок 2.4 - Контекстна діаграма застосунку-тренажеру

Розглянемо декомпозицію основних процесів діаграми (рис. 2.5).

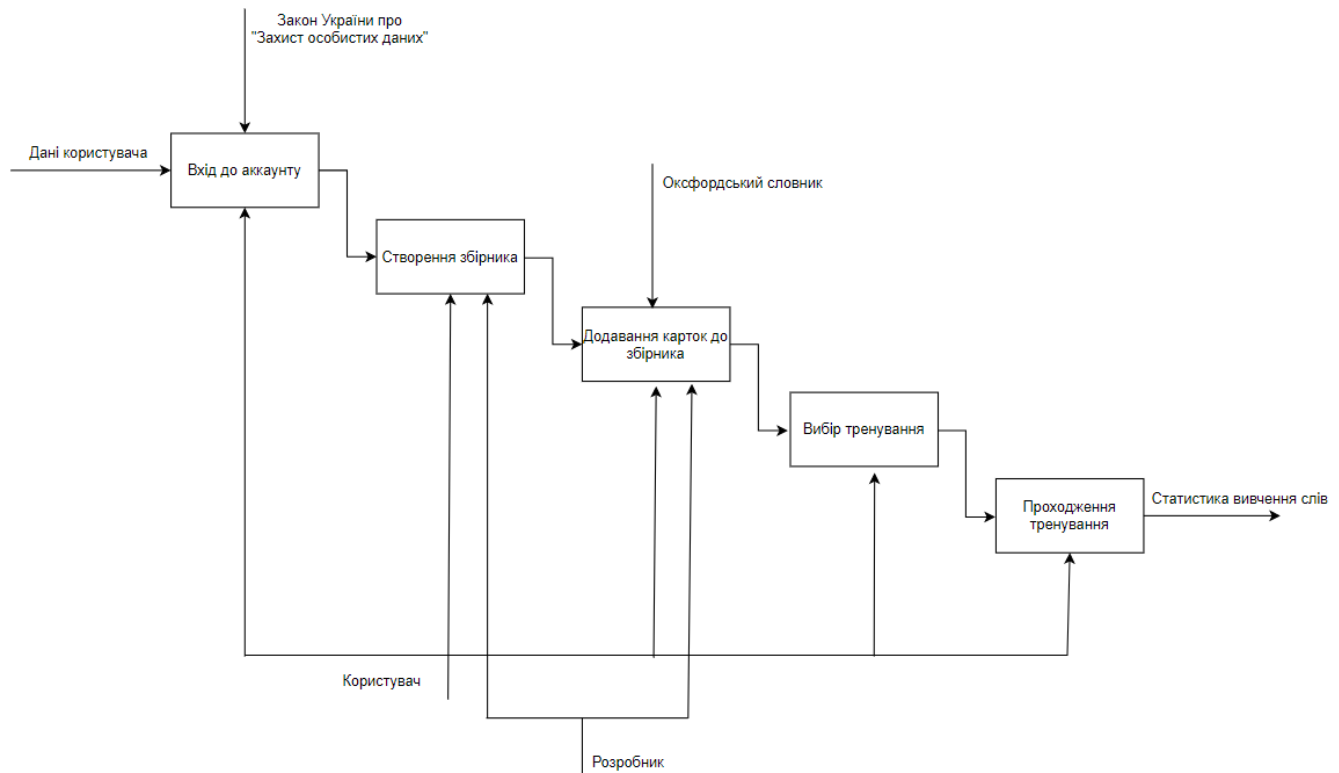


Рисунок 2.5 – Декомпозиція основних процесів діаграми ЯК БУДЕ

### 2.1.3 Архітектура застосунку-тренажеру

Архітектура інформаційної системи вивчення карток буде мати два модулі: модуль роботи з користувачем та адміністративний модуль.

Модуль роботи з користувачами, включає в себе чотири модулі:

- модуль реєстрації, який дозволяє новим користувачам створити особистий профіль для майбутнього користуванням системи, та зручності збереженням інформації за користувачем;
- модуль роботи з особистими збірниками дозволяє створювати, редагувати та видаляти особисті слова для кожного користувача;
- модуль готових збірників дозволяє користувачу користуватись збірниками від адміністрації додатку для швидкого вивчення слів за темою збірника;
- модуль тренувань, який дозволяє відслідковувати вивчений матеріал.

Адміністративний модуль включає в себе доступ до підсистеми бази даних

збірників карток та роботу з ним, дозволяє редагувати окремі збірники та картки для всіх користувачів в онлайн режимі.

## 2.2. Розробка інформаційного забезпечення застосунку - тренажеру

Важливим етапом є створення концептуальної моделі бази даних (рис. 2.7).

На ній для заданої предметної області відображено наступні сутності:

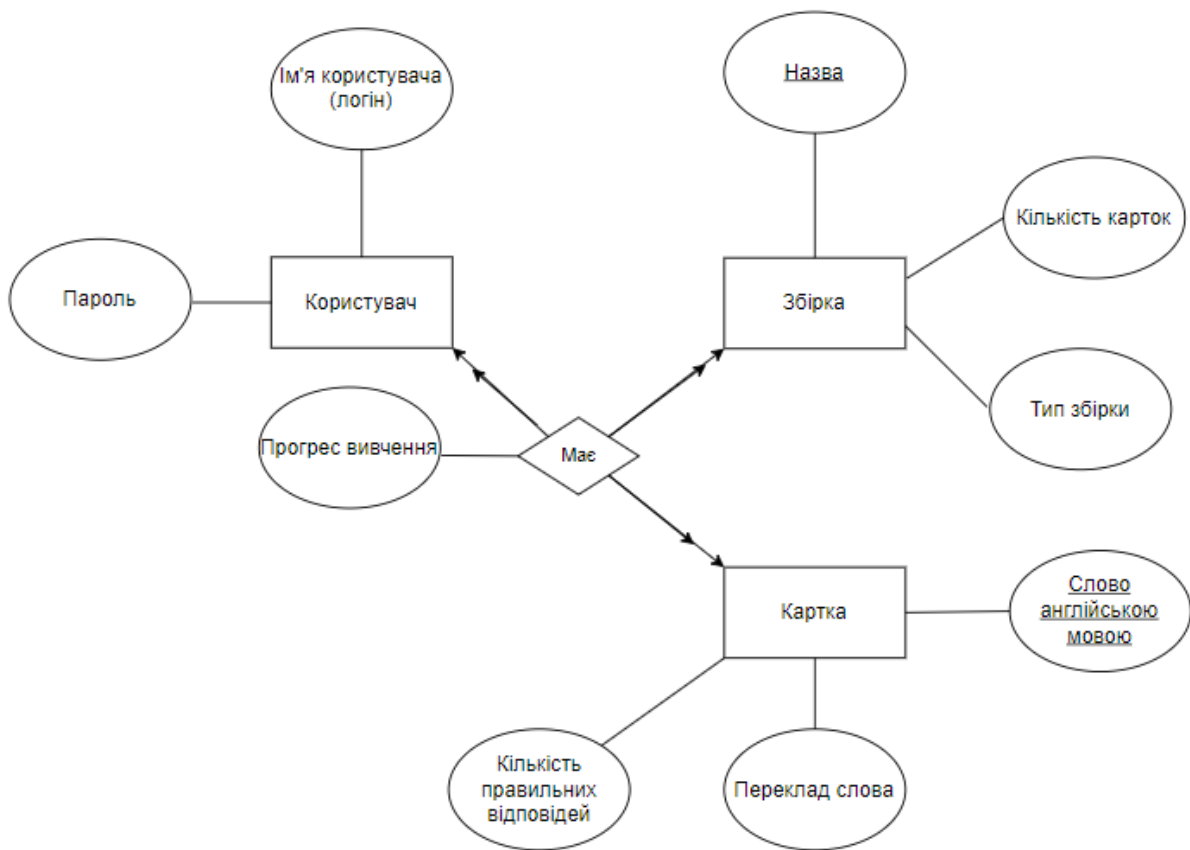
- Користувач
- Збірка
- Картка

*Таблиця 2.1 – Зв'язки між сутностями*

№	Сутності, що утворюють зв'язок	що	Тип зв'язку	Пояснення
1	Користувач збірки	–	Багато до багатьох	Один користувач створювати багато збірників. Один і той самий збірник може бути у кількох користувачів.
2	Збірник – Картка		Багато до багатьох	Збірник має багато карток, карта може міститися в кількох збірниках.
3	Користувач Картка	–	Багато до багатьох	Користувач може мати кілька карток. Одна і та сама картка може бути у різних користувачів.

**Рисунок 2.7 – Концептуальна модель застосунку-тренажеру**

Розглянемо логічну модель (рис. 2.8). Вона також, як і концептуальна, описує дані на досить абстрактному рівні, але все ж



більше наближеному до реальних баз даних. Також, на відміну від попередньої моделі, ми вже вказуємо первинні ключі та зовнішні ключі, які пов'язують таблиці між собою.

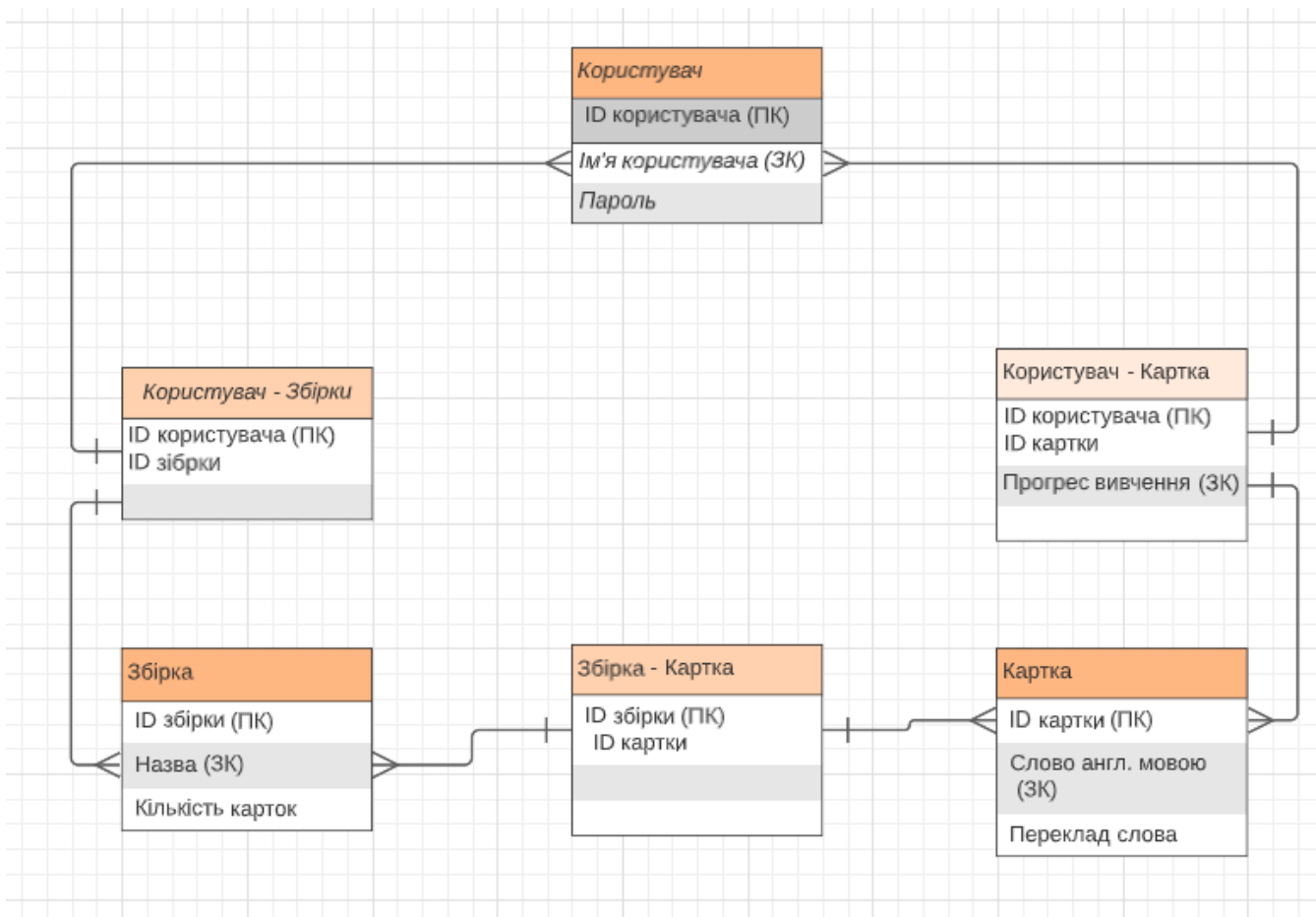


Рисунок 2.8 – Логічна модель бази даних

Розробимо фізичну модель, на основі даталогічної. Детальну інформацію про атрибути запишемо у таблицю.

Таблиця 2.2 - Склад та характеристика атрибутів таблиць

№ п/п	Назва елемента даних	Тип елемента даних	Обов'язкове значення	Обмеження	Ключ
<b>Таблиця «Користувач»</b>					
1	Ім'я користувача	Символьний	так	20	ПК
2	Пароль	Символьний	так	20	ЗК
<b>Таблиця «Збірки»</b>					

1	Назва	Символьний	так	20	ПК
2	Кількість карток	Ціле число	так		ЗК
<b>Таблиця «Картка»</b>					
1	Слово англ. мовою	Символьний	так	20	ПК
2	Переклад	Символьний	так	20	ЗК

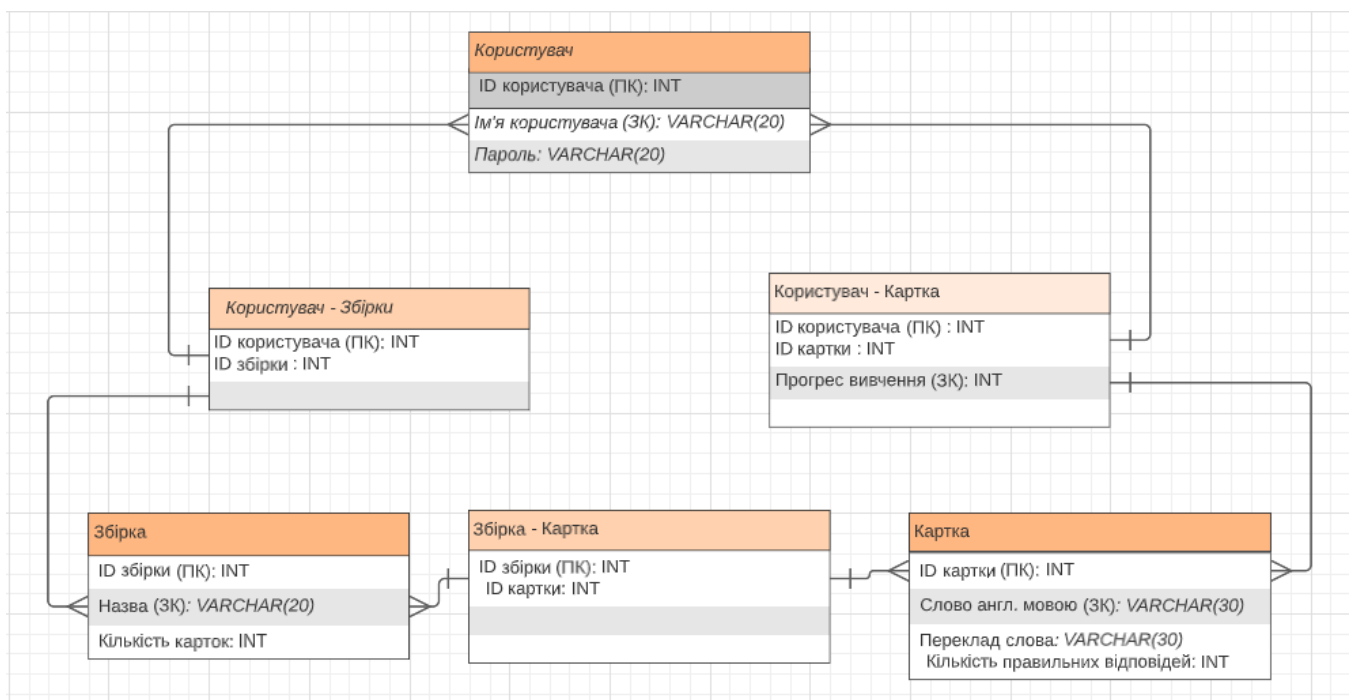


Рисунок 2.9 – Фізична модель бази даних

## 2.3 Математична постановка основної задачі визначення прогресу

Головною задачею інтерактивного застосунку-тренажеру є вивчення англomовний слів, тому кожен користувач буде мати можливість відстежувати свій прогрес навчання. Так для кожної збірки користувач буде знати відсоток слів, які він вивчив з неї. Слово буде вважати вивченим, якщо користувач правильно відповів на нього під час тренування мінімум 5 разів. Тренування для кожної збірки буде відбуватися наступним чином: рандомно з'являється картка з англійським словом, до цієї картки буде запропоновано 4 варіанти перекладу, лише один варіант є правильним. Якщо користувач одразу обирає правильний варіант відповіді – система запам'ятовує це. Для того, щоб слово вважалось точно вивченим, користувач має ще 4 рази з першого разу вказати правильний переклад цього слова. Інакше, якщо користувач обирає неправильний варіант перекладу – кількість варіантів відповіді зменшується на 1 (тобто обраний неправильний варіант зникає), і так до тих пір, поки користувач не обере правильну відповідь.

Тобто прогрес вивчення окремої картки рахується тільки тоді, коли користувач з першого разу правильно вказує переклад.

Загальний прогрес вивчення збірки  $V$  рахується як відношення повністю вивчених карток  $m$  до їх загальної кількості у збірці  $n$ :

$$V = m / n * 100\% .$$

Таким чином, для кожної збірки буде розраховано прогрес її вивчення, також відрахунок вивчених слів дозволить користувачу переглядати загальну статистику, а саме кількість правильних/неправильних відповідей, загальну кількість вивчених слів, та кількість слів, яку йому ще потрібно вивчити.

## РОЗДІЛ 3. РОЗРОБКА ІНТЕРАКТИВНОГО ЗАСТОСУНКУ – ТРЕНАЖЕРУ ДЛЯ ВИВЧЕННЯ АНГЛОМОВНИХ СЛІВ

### 3.1 Вибір мови розробки

Найпопулярнішою та найпрактичнішою мовою для розробки мобільних застосунків є Java. Java – один із найстаріших та перевірених методів створення якісної програми. Це популярна мова програмування для Android, яка у 2019 році увійшла до ТОП-5 найкращих та функціональних мов.

Мною мова Java була обрана через такі переваги:

- Концепція ООП (об'єктно-орієнтованого програмування). Це означає, що програміст сам визначає тип даних, його структуру та набір функцій, що застосовуються до нього. Це дозволяє багаторазово використовувати одні й самі об'єкти у різних програмах, зручніше організації структури програм (особливо з великим обсягом коду), допомагає уникати помилок і спрощує підтримку і модернізацію старого коду.
- Це мова високого рівня, тобто вона більше схожа на людську мову, а не на машинний код. Отже, у неї порівняно простий синтаксис, що робить його швидким для освоєння та зручним для написання коду, його читання та обслуговування.
- Безпека. У Java є кілька функцій, які ліквідують уразливості, що часто зустрічаються. Зокрема, це Security Manager – створювана для кожної програми політика безпеки, в якій можна прописати правила доступу.
- Зручність для розподіленого програмування. Ця мова спочатку створювалася для спільної роботи (у тому числі віддаленої), тому вона дозволяє спільно використовувати дані та програми декількома комп'ютерами одночасно.
- Принцип «написати один раз і використовувати скрізь» - написаний на Java додаток можна запустити на будь-якій платформі, що його підтримує.

- Стабільна спільнота, що постійно розвивається. За чисельністю та активністю з нею мало хто може змагатися. У Мережі є маса ресурсів, де на будь-яке питання з теми або вже є відповідь, або знайдеться хто-небудь, хто його підкаже, як і сотні курсів, семінарів і навчальних програм, як платних, так і безкоштовних.

Головною перевагою Java є наявність власного середовища розробки під назвою Android Studio. Саме у ньому і відбувалася розробка застосутку.

Android Studio також має багато переваг:

- Розширений редактор макетів: WYSIWYG, здатність працювати з UI компонентами за допомогою Drag-and-Drop, функція попереднього перегляду макета на кількох конфігураціях екрана.
- Складання додатків, засноване на Gradle.
- Різні види збірок та генерація декількох .apk файлів.
- Рефакторинг коду.
- Статичний аналізатор коду (Lint), що дозволяє знаходити проблеми продуктивності, несумісності версій та інше.
- Вбудований ProGuard та утиліта для підписування додатків.
- Шаблони основних макетів та компонентів Android.
- Підтримка розробки програм для Android Wear та Android TV.
- Вбудована підтримка Google Cloud Platform, яка включає інтеграцію з сервісами Google Cloud Messaging і App Engine.
- Android Studio 2.1 підтримує Android N Preview SDK, а це означає, що розробники зможуть розпочати роботу зі створення програми для нової програмної платформи.

3.2 Опис основних програмних компонентів застосунку- тренажеру для вивчення англомовних слів.

Logout.java - розробка можливості входу як гість, або як зареєстрований користувач.

MainScreen.java - розробка головного екрану при запуску застосунку-тренажеру.

NavigationDrawer.java - розробка налаштувань додатку, статистики та тренувань.

SetsandCards.java - розробка сторінок зі збірками та сторінок з картками.

Startpages.java - розробка таких компонентів як “Забув пароль”.

Translator.java - розробка можливості вибору варіантів перекладу для слів.

DB.java - описана вся робота з базою даних.

3.3 Розробка та опис програмного інтерфейсу з користувачем (керівництво користувача) інтерактивного застосунку-додатку для вивчення англомовних слів

Одразу після запуску додатку у користувача з’являється кілька можливостей входу в додаток, а саме: зареєструватися та увійти, увійти як вже зареєстрований користувач, та увійти як гість. Якщо користувач використовує додаток як гість, то його прогрес не відстежується на інших девайсах.

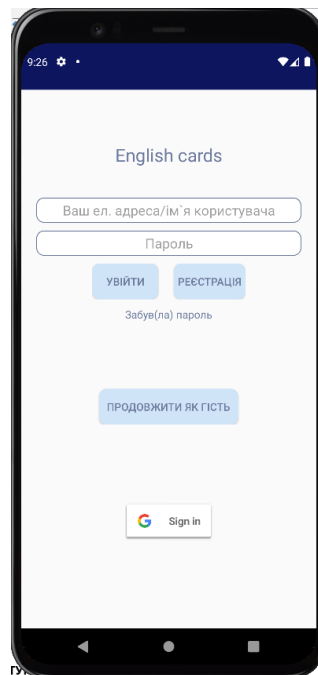


Рисунок 3.1 – Екран входу у застосунок

Після того, як користувач увійшов у додаток, йому буде запропоновано можливість переглянути збірки, створені розробником додатку, або створити нові збірки.



Рисунок 3.2 – Екран всіх наборів карток

Натиснувши на кнопку меню, користувач зможе перейти до його власностворених збірок, розпочати тренування або зареєструватися, якщо він був увійшов як гість.

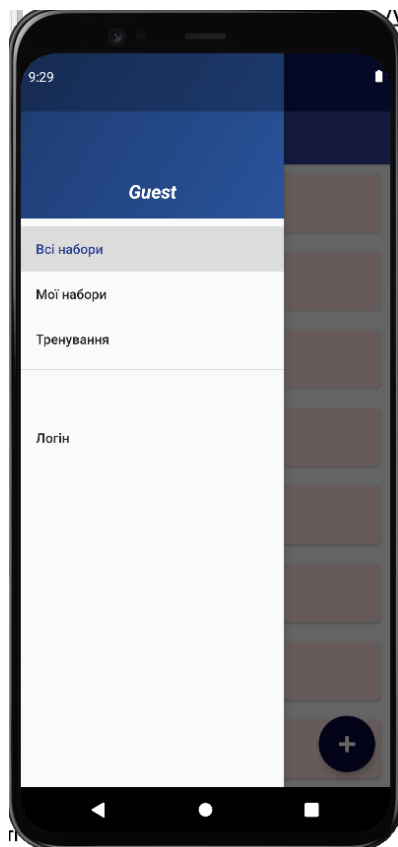


Рисунок 3.3 – Екран меню

Обравши «Всі набори», користувач зможе переглянути всі збірки, які створені розробником, або, безпосередньо, створені самим користувачем.



Рисунок 3.4 – Всі збірки користувача

Натиснувши «+» у правому нижньому кутку, користувач зможе створити нову власну збірку та додати картки до неї.

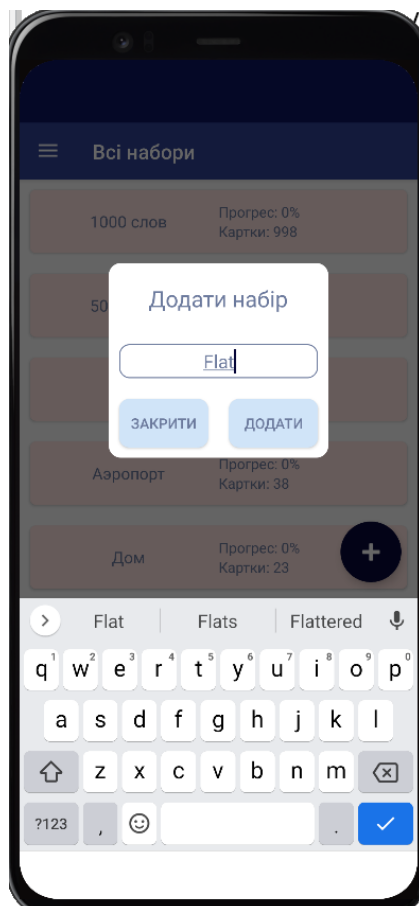


Рисунок 3.5 – Створення нової збірки

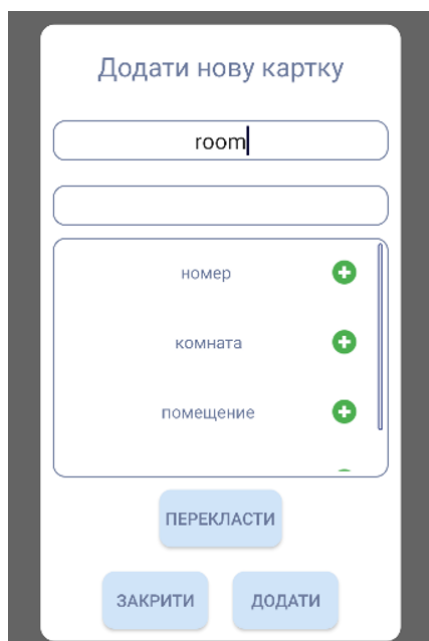


Рисунок 3.6 – Можливість створити нову картку та обрати переклад для неї

Також натиснувши на будь-який набір, користувач зможе переглянути всі слова, які в ньому є.



Рисунок 3.7 – Слова збірки «5000 слів»

Повернувшись до меню, користувач може обрати можливість розпочати тренування та обрати збірки (одну або кілька) для нього.

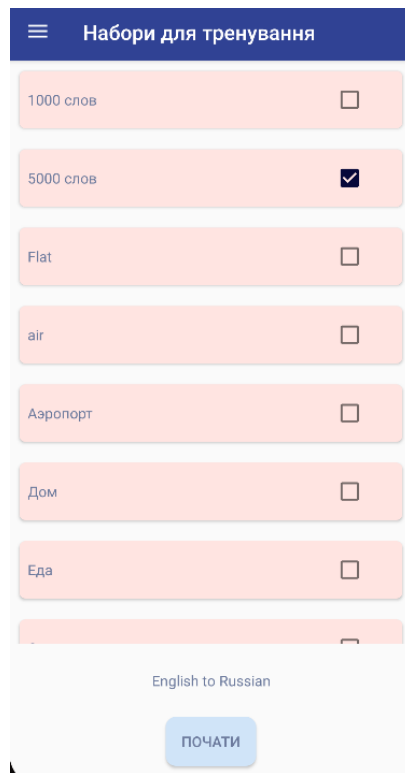


Рисунок 3.8 – Обрання збірки для тренування

Після цього, користувач розпочинає тренування. Процес тренування вже був описаний раніше.



Рисунок 3.9 – Одна з карток тренування «5000 слів»

Якщо користувач зареєструвався у додатку – з'явиться можливість переглядати статистику.

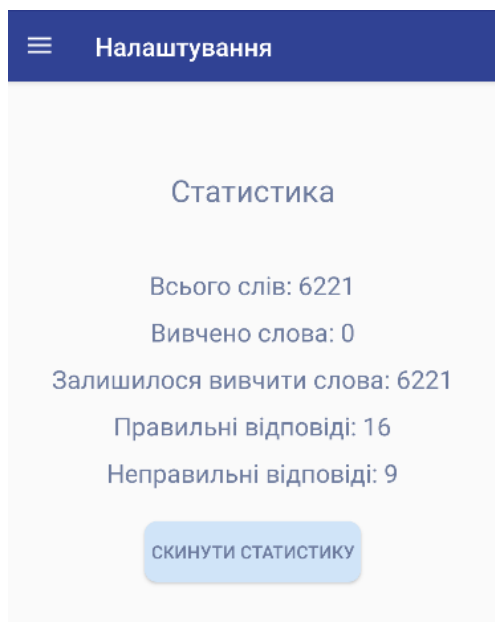


Рисунок 3.10 – Статистика вивчення слів

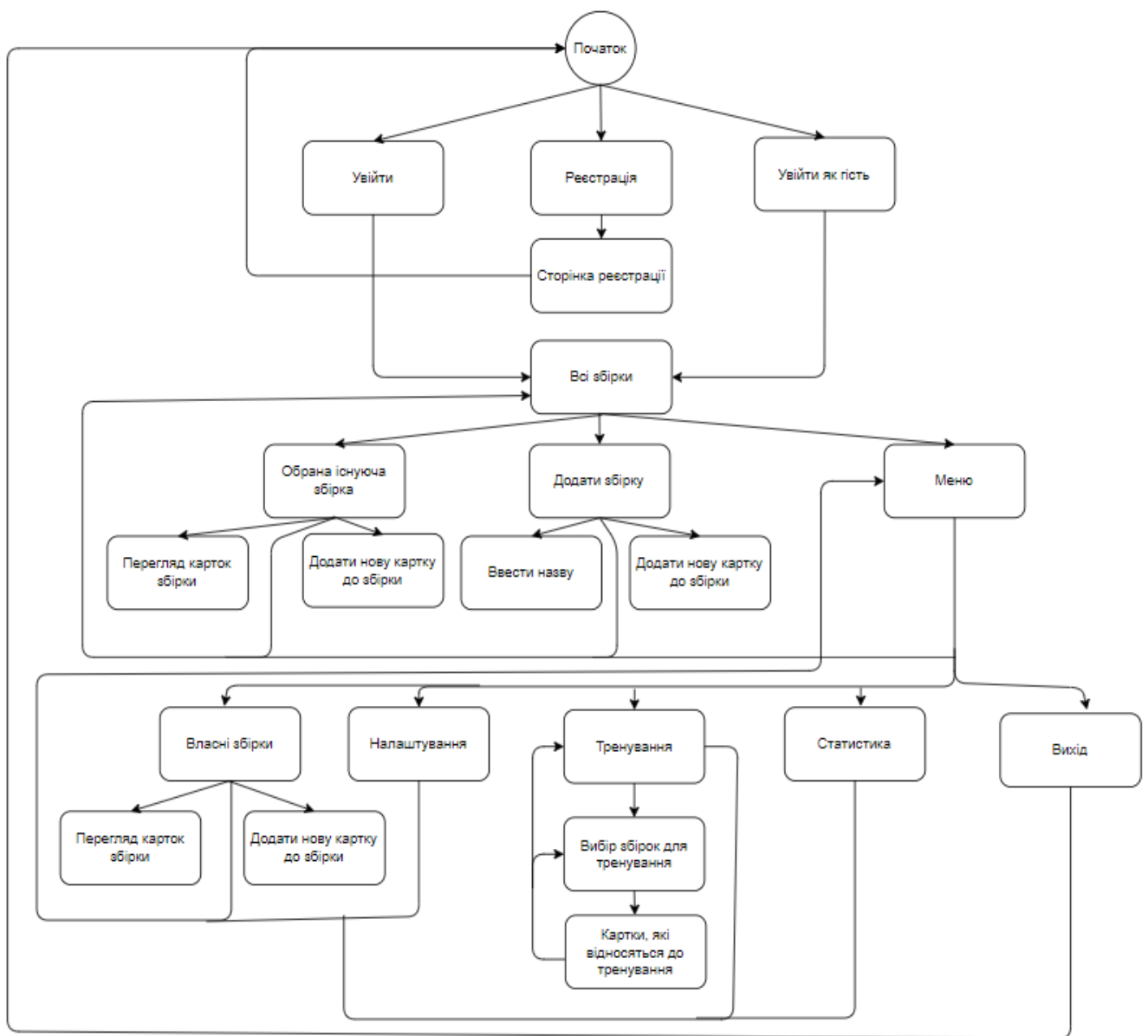


Рисунок 3.11 – Граф переходів між сторінками додатку

### 3.4 Перевірка працездатності інтерактивного застосунку-тренажеру для вивчення англомовних слів

Створимо нову збірку «School».

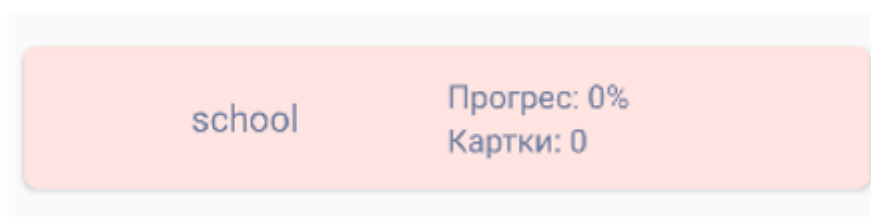


Рисунок 3.11 – Нова збірка «School»

Додамо картки до збірки.

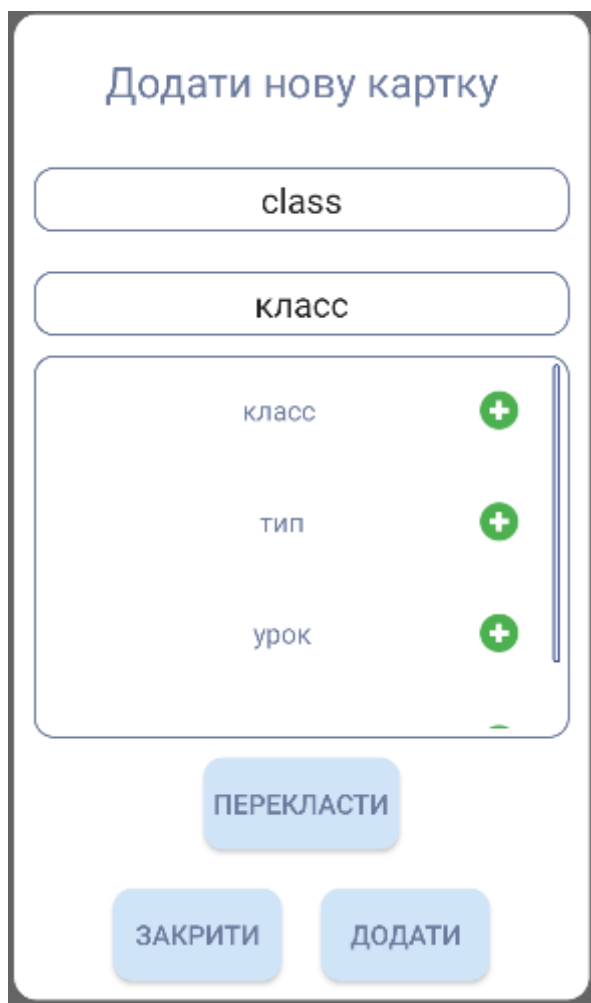


Рисунок 3.12 – Додавання картки «class» до збірки

Тепер збірка «school» містить 4 картки.



Рисунок 3.13 – Картки збірки «school»

Розпочинаємо тренування.



Рисунок 3.14 – Проходження тренування. Вибір правильної відповіді

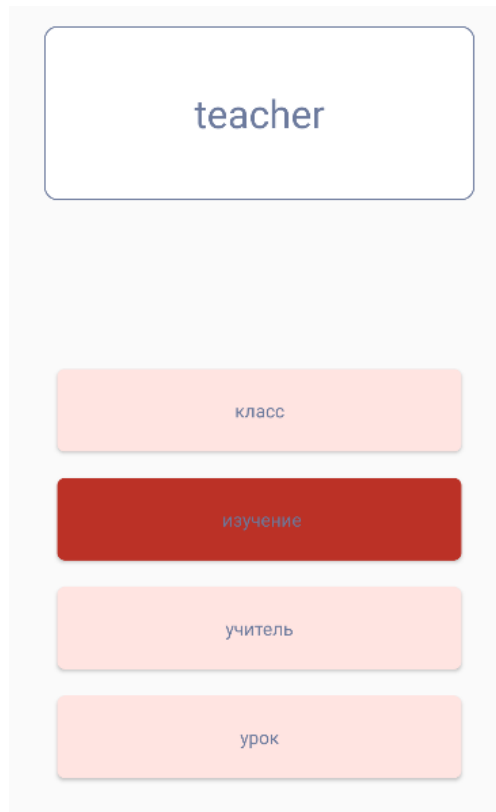


Рисунок 3.15– Проходження тренування. Вибір не правильної відповіді

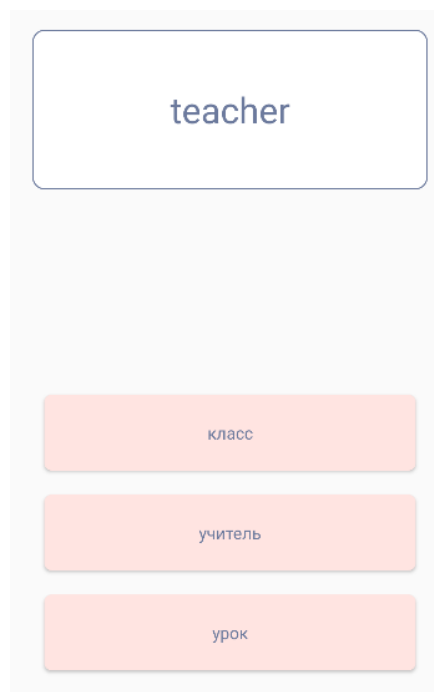


Рисунок 3.16 – Проходження тренування. Варіанти відповіді після вибору неправильної відповіді

Тепер після правильних 5 відповідей на 2 слова маємо прогрес вивчення 50%.

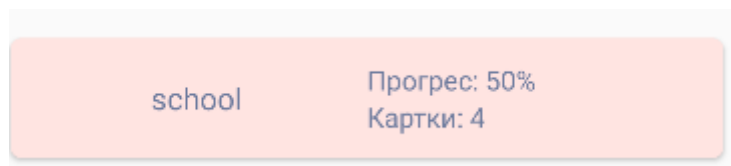


Рисунок 3.17 – Розрахунок прогресу

Отже, можна зробити висновок, що розроблений застосунок-тренажер працює коректно.

## ВИСНОВКИ

У випускній кваліфікаційній роботі було проведено аналіз існуючих застосунків-тренажерів, створено інтерактивний застосунок-тренажер, за допомогою якого можливо якісно вивчати слова англійської мови.

Застосунок – тренажер можливо надалі завантажити до Google Play, щоб багато користувачів мали до нього вільний доступ та змогли завантажити його на свої пристрої. На основі цього також можна провести бета-тестування застосунку.

## Список використаних джерел

1. Береза А.М. Основи створення інформаційних систем: Навчальний посібник / Береза А.М. — Київ : КНЕУ, 1998. — 307 с.
2. Галузинский Г.П. Сучасні технологічні засоби обробки інформації: Навчальний посібник / Галузинский Г.П., Гордієнко І.В. — Київ : КНЕУ, 1998. — 132 с.
3. Єр'оміна Н.В. Проектування баз даних: Навчальний посібник / Єр'оміна Н.В. — Київ : КНЕУ, 1998. — 305 с.
4. Браткевич В.В. Інформатика. Комп'ютерна техніка. Комп'ютерні технології / Браткевич В. В., Бутов. — Київ : Академія, 2002. — 242 с. — (Підручник).
5. Інформаційні системи і технології в економіці / Під ред. В.С. Пономаренко. — Київ : Академія, 2002. — 362 с.
6. Бердтис А. Структуры данных / Бердтис А. — М. : Статистика, 1974. — 408 с.
7. Бойков.В. Проектирование баз данных информационных систем / Бойков.В., Савинков В.М. — М.: Мир, 1997. — 209 с.
8. <https://www.cischool.ru/plyusy-i-minusy-java/>
9. [https://ru.wikipedia.org/wiki/android\\_studio](https://ru.wikipedia.org/wiki/android_studio)
10. <https://www.fluentu.com/blog/englishrus/%D0%BA%D0%B0%D1%80%D1%82%D0%BE%D1%87%D0%BA%D0%B8-%D0%B4%D0%BB%D1%8F-%D0%B8%D0%B7%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B0%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%BE%D0%B3%D0%BE/>
11. <https://englishfull.ru/znat/lingualeo.html>
12. <https://preply.com/ru/blog/obzor-babbel-duolingo/>
13. <https://kibit.edu.ua/ru/top-10-prilozhenij-dlya-izucheniya-anglijskogo/>
14. <https://lifehacker.ru/apps-to-learn-english/>
15. <https://pilot-school.ru/mobilnoe-prilozhenie-ewa-dlya-izucheniya-anglijskogo-yazyka-cto-ono-umeet-otzyvy-preimushhestva-i-nedostatki/>
16. <https://startpack.ru/compare/duolingo-learn-language/memrise-learn-language>

17. <https://pilot-school.ru/busuu-obzor-prilozheniya-vozmozhnosti-plyusy-i-minusy/>
18. <https://langformula.ru/busuu/>

## ДОДАТКИ

### DB.java

```
package com.en.cards.DB;

import android.app.ProgressDialog;
import android.content.Context;
import android.database.Cursor;
import android.database.DatabaseUtils;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteStatement;
import android.os.AsyncTask;

import com.en.cards.R;
import com.en.cards.SetsAndCards.CardsPage.Model.CardModel;
import com.en.cards.Helper;
import com.en.cards.Parse.Listeners.IGetAllUserData;
import com.en.cards.SetsAndCards.SetsPage.Model.SetModel;
import com.parse.ParseUser;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class DatabaseHelper extends SQLiteOpenHelper {
```

```

String DB_PATH=null;
private final static String DB_NAME= "engdb";
private SQLiteDatabase myDataBase;
private final Context myContext;
Helper helper;

public DatabaseHelper(Context context) {
    super(context, DB_NAME, null, 1);
    this.myContext = context;
    this.DB_PATH=
String.valueOf(context.getDatabasePath(DB_NAME));
    helper = new Helper(myContext);
}

public void createDataBase() {
    boolean mDataBaseExist = checkDataBase();
    if (!mDataBaseExist) {
        this.getReadableDatabase();
        try {
            copyDataBase();
        } catch (IOException mIOException) {
            mIOException.printStackTrace();
            throw new Error("Error copying database");
        } finally {
            this.close();
        }
    }
}

public void deleteDataBase()
{
    String mPath = DB_PATH + DB_NAME;
    final File file = new File(mPath);
}

```

```

        SQLiteDatabase.deleteDatabase(file);
    }

    private boolean checkDataBase() {
        try {
            final String mPath = DB_PATH + DB_NAME;
            final File file = new File(mPath);
            if (file.exists())
                return true;
            else
                return false;
        } catch (SQLiteException e) {
            e.printStackTrace();
            return false;
        }
    }

    private void copyDataBase() throws IOException {
        try {
            InputStream mInputStream =
myContext.getAssets().open(DB_NAME);

            String outFileName = DB_PATH + DB_NAME;

            OutputStream mOutputStream = new
FileOutputStream(outFileName);

            byte[] buffer = new byte[1024];
            int length;
            while ((length = mInputStream.read(buffer)) > 0) {
                mOutputStream.write(buffer, 0, length);
            }

            mOutputStream.flush();
            mOutputStream.close();
            mInputStream.close();
        } catch (Exception e) {
            e.printStackTrace();

```

```

    }
}

public boolean openDataBase() throws SQLException {
    String mPath = DB_PATH + DB_NAME;
    myDataBase = SQLiteDatabase.openDatabase(mPath, null,
SQLiteDatabase.OPEN_READWRITE);
    return myDataBase.isOpen();
}

@Override
public synchronized void close() {
    if (myDataBase != null)
        myDataBase.close();
    SQLiteDatabase.releaseMemory();
    super.close();
}

@Override
public void onCreate(SQLiteDatabase db) {

}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {

}

public ArrayList<SetModel> GetAllSets() {
    openDataBase();
    ArrayList<SetModel> models = new ArrayList<>();
}

```

```

        Cursor cursor = myDataBase.rawQuery("SELECT set_tbl.set_id,
set_tbl.set_name, set_tbl.train, avg(card_tbl.progress) as progress,
count(card_tbl.card_id) as cardsNumber, set_tbl.default_set from
set_tbl left join card_tbl on set_tbl.set_id = card_tbl.set_id GROUP
By set_tbl.set_name", null);

        while(cursor.moveToNext()) {

            Integer setId =
cursor.getInt(cursor.getColumnIndex("set_id"));

            String set_name =
cursor.getString(cursor.getColumnIndex("set_name"));

            Integer progress =
cursor.getInt(cursor.getColumnIndex("progress"));

            Integer cardsNumber =
cursor.getInt(cursor.getColumnIndex("cardsNumber"));

            Boolean isTraining =
cursor.getInt(cursor.getColumnIndex("train"))>0;

            Boolean default_set =
cursor.getInt(cursor.getColumnIndex("default_set"))>0;

            models.add(new SetModel(setId, set_name, progress,
isTraining, cardsNumber, default_set));

        }

        cursor.close();

        return models;

    }

```

```

public SetModel GetSet (String setName){
    ArrayList<SetModel> sets = GetAllSets();
    for (SetModel setModel : sets){
        if (setModel.getSetName().equals(setName))
            return setModel;
    }
    return null;
}

```

```

public ArrayList<SetModel> GetTrainingSets() {
    openDataBase();
    ArrayList<SetModel> models = new ArrayList<>();

```

```

        Cursor cursor = myDataBase.rawQuery("SELECT set_tbl.set_id,
set_tbl.set_name, set_tbl.train, avg(card_tbl.progress) as progress,
count(card_tbl.card_id) as cardsNumber, default_set from set_tbl
left join card_tbl on set_tbl.set_id = card_tbl.set_id WHERE
set_tbl.train=1 GROUP By set_tbl.set_name", null);

        while(cursor.moveToNext()) {

            Integer setId =
cursor.getInt(cursor.getColumnIndex("set_id"));

            String set_name =
cursor.getString(cursor.getColumnIndex("set_name"));

            Integer progress =
cursor.getInt(cursor.getColumnIndex("progress"));

            Boolean isTraining =
cursor.getInt(cursor.getColumnIndex("train"))>0;

            Integer cardsNumber =
cursor.getInt(cursor.getColumnIndex("cardsNumber"));

            Boolean default_set =
cursor.getInt(cursor.getColumnIndex("default_set"))>0;

            models.add(new SetModel(setId, set_name, progress,
isTraining, cardsNumber, default_set));

        }

        cursor.close();

        return models;

    }

```

```

    public void AddCard (String set_name, String firstText, String
secondText)

    {

        openDataBase();

        String tempEnText, tempRuText;

        if (Helper.WordBelongToEnAlphabet(firstText))

        {

            tempEnText = EscapeString(firstText);

            tempRuText = EscapeString(secondText);

        }

        else {

            tempEnText = EscapeString(secondText);

            tempRuText = EscapeString(firstText);

        }
    }

```

```

    }

    myDataBase.execSQL("INSERT INTO card_tbl (set_id, en_text,
ru_text) VALUES ((SELECT set_id from set_tbl where
set_name="+EscapeString(set_name)+"),
"+tempEnText+", "+tempRuText+" ");

}

public void UpdateCard(Integer card_id, String firstText, String
secondText) {

    openDataBase();

    String tempEnText, tempRuText;

    if (Helper.WordBelongToEnAlphabet(firstText))

    {

        tempEnText = EscapeString(firstText);
        tempRuText = EscapeString(secondText);

    }

    else {

        tempEnText = EscapeString(secondText);
        tempRuText = EscapeString(firstText);

    }

    myDataBase.execSQL("UPDATE card_tbl SET
en_text="+tempEnText+", ru_text="+tempRuText+" WHERE
card_id='"+card_id+"'");

};

public void DeleteCard (String card_name){

    openDataBase();

    myDataBase.execSQL("DELETE FROM card_tbl WHERE
en_text="+EscapeString(card_name)+" OR
ru_text="+EscapeString(card_name)+"");

}

```

### Logout.java

```

package com.en.cards.MainScreen.Logout;

import android.app.Activity;
import android.content.Intent;

```

```

import com.en.cards.DB.DatabaseHelper;
import com.en.cards.Dialogs.DialogHelper;
import com.en.cards.Dialogs.Listeners.IDialogYesNoMaybe;
import com.en.cards.R;
import com.en.cards.StartPages.StartPage;
import java.io.Serializable;

public class LogoutGuest implements ILogout, IDialogYesNoMaybe,
Serializable {
    DatabaseHelper databaseHelper;
    DialogHelper dialogHelper;
    Activity activity;

    public LogoutGuest(Activity activity) {
        databaseHelper = new DatabaseHelper(activity);
        dialogHelper = new DialogHelper (activity);
        this.activity = activity;
    }

    @Override
    public void Logout() {
        dialogHelper.ShowDialogYesNoMaybe(this,
activity.getResources().getString(R.string.logout_guest),
activity.getResources().getString(R.string.login),
activity.getResources().getString(R.string.cancel),
activity.getResources().getString(R.string.registration));
    }

    @Override
    public void PressYes() {
        databaseHelper.deleteDataBase();
    }
}

```

```

        activity.startActivity(new Intent(activity,
StartPage.class));
        activity.finish();
    }

    @Override
    public void PressNo() {

    }

    @Override
    public void PressMaybe() {
        activity.startActivity(new Intent(activity,
StartPage.class).putExtra("redirectToSignUp", true));
        activity.finish();
    }
}

```

### SetsandCards

```

package com.en.cards.SetsAndCards.CardsPage;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Handler;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

import androidx.fragment.app.DialogFragment;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;

```

```

import android.widget.TextView;

import
com.en.cards.SetsAndCards.CardsPage.AddNewCard.AddCardFragment;
import com.en.cards.SetsAndCards.CardsPage.Adapter.CardAdapter;
import com.en.cards.SetsAndCards.CardsPage.Model.CardModel;
import com.en.cards.DB.DatabaseHelper;
import com.en.cards.Helper;
import com.en.cards.R;
import com.en.cards.Tooltip.TooltipHelper;
import com.github.amlcurran.showcaseview.ShowcaseView;

import
com.google.android.material.floatingactionbutton.FloatingActionButto
n;

import java.util.ArrayList;
import java.util.Objects;

public class CardsPage extends AppCompatActivity implements
View.OnClickListener, DialogInterface.OnDismissListener {

    public static RecyclerView recyclerView;
    ArrayList<CardModel> cards;
    DatabaseHelper databaseHelper;
    String set_name;
    RecyclerView.LayoutManager layoutManager;
    CardAdapter cardAdapter;
    Helper helper;
    TextView directionTextView;
    public static TextView emptyView;
    FloatingActionButton addCard;
    AddCardFragment addCardFragment;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cards_page);
    }
}

```

```

        ActionBar actionBar = getSupportActionBar();
        assert actionBar != null;
        actionBar.setDisplayHomeAsUpEnabled(true);
        helper = new Helper(CardsPage.this);
        directionTextView = findViewById(R.id.direction);
        directionTextView.setOnClickListener(this);
        directionTextView.setText(helper.GetDirection().name);
        addCard = findViewById(R.id.fab);
        addCard.setScaleType(ImageView.ScaleType.CENTER);
        addCard.setOnClickListener(this);
        emptyView = findViewById(R.id.empty_view);
        addCardFragment = new AddCardFragment();
        databaseHelper = new DatabaseHelper(CardsPage.this);
        Intent intent = getIntent();
        set_name = intent.getStringExtra("set_name");
        actionBar.setTitle(set_name);
        cards = databaseHelper.GetCardsBySet(set_name);
        recyclerView = findViewById(R.id.cardrecyclerview);
        layoutManager = new GridLayoutManager(this, 2);
        recyclerView.setLayoutManager(layoutManager);
        cardAdapter = new CardAdapter(cards, CardsPage.this,
set_name);
        recyclerView.setAdapter(cardAdapter);
        registerForContextMenu(recyclerView);
        Helper.ShowLabelNoRecordsRecyclerView(CardsPage.emptyView,
cardAdapter, CardsPage.recyclerView);
        cardAdapter.notifyDataSetChanged();

        if (cards.size()>0) {
            final Handler handler = new Handler();
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {

```

```

        View view =
Objects.requireNonNull(recyclerView.findViewHolderForAdapterPosition
(0)).itemView;

        if (view != null) {

            TooltipHelper tooltipHelper = new
TooltipHelper(CardsPage.this);

            final ArrayList<ShowcaseView.Builder>
builders = new ArrayList<ShowcaseView.Builder>();

            ShowcaseView.Builder cardTooltip =
tooltipHelper.CardTooltip(view);

            // ShowcaseView.Builder fabButton =
tooltipHelper.FabCardPageTooltip(addCard);

            ShowcaseView.Builder direction =
tooltipHelper.DirectionTooltip(directionTextView);

            if
(cardTooltip!=null){builders.add(cardTooltip);}

            // builders.add(fabButton);

            if(direction!=null){builders.add(direction);}

            tooltipHelper.RunListShowcase(builders);

        }

    }

    }, 500);

}

}

@Override

public void onClick(View v) {

    switch (v.getId()){

        case R.id.direction:

            helper.SetNewDirection(directionTextView);

            cardAdapter = new CardAdapter(cards, CardsPage.this,
set_name);

            cardAdapter.notifyDataSetChanged();

            recyclerView.setAdapter(cardAdapter);

            break;

        case R.id.fab:

```

```

        Bundle bundle = new Bundle();
        bundle.putString("setName", set_name);
        addCardFragment.setArguments(bundle);
        addCardFragment.show(getFragmentManager(), "dlg1");
        break;
    case R.id.home:
        finish();
        break;
    }
}

@Override
public void onDismiss(DialogInterface dialog) {
    cards = databaseHelper.GetCardsBySet(set_name);
    cardAdapter = new CardAdapter(cards, CardsPage.this,
set_name);
    recyclerView.setAdapter(cardAdapter);
    Helper.ShowLabelNoRecordsRecyclerView(emptyView, cardAdapter,
recyclerView);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            finish();
            break;
    }
    return true;
}
}

```

StartPage.java

```

package com.en.cards.StartPages;

import android.os.Bundle;
import android.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;

import com.en.cards.Helper;
import com.en.cards.Parse.Listeners.IResetPassword;
import com.en.cards.Parse.ParseHelper;
import com.en.cards.R;

public class ForgotPassword extends Fragment implements
View.OnClickListener, IResetPassword {

    EditText email;
    Button reset;
    ParseHelper parseHelper;
    String login="";
    Helper helper;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        parseHelper = new ParseHelper(getActivity());
        helper = new Helper(getActivity());
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,

```

```

        Bundle savedInstanceState) {
            Bundle arguments = getArguments();
            if (arguments!=null) {
                login = arguments.getString("login", "");
            }

            View v = inflater.inflate(R.layout.fragment_forgot_password,
container, false);
            email = v.findViewById(R.id.email);
            email.setText(login);
            reset = v.findViewById(R.id.reset);
            reset.setOnClickListener(this);
            return v;
        }

        @Override
        public void onClick(View v) {
            switch (v.getId()){
                case R.id.reset:
                    parseHelper.ResetPassword(email.getText().toString(),
this );
                    break;
            }
        }

        @Override
        public void PasswordIsReset() {
            Bundle bundle = new Bundle();
            bundle.putString("login", email.getText().toString());
            Helper.ReplaceFragment(R.id.mainContainer,
LoginFragment.class, getActivity(), null);
        }
    }
}

```

