

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

Бакалаврська робота

на тему: «Розробка програми для вивчення і тестування з польської мови»

Виконав: студент _____ IV _____ курсу
групи _____ ЗП-18б _____
спеціальності _____

121 – Інженерія програмного забезпечення

(шифр і назва напрямку підготовки, спеціальності)

Манич А. М.

(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ Ракитянська Г. Б.

(прізвище та ініціали)

Рецензент: к.т.н., доц. каф. КН Арсенюк І. Р.

(прізвище та ініціали)

Допущено до захисту

Зав. кафедри _____

« _____ » _____ 2022 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти перший бакалаврський
Галузь знань 12 – Інформаційні технології
Спеціальність 121 – Інженерія програмного забезпечення
Освітньо-професійна програма – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ

Романюк О. Н. _____

“ 25 ” березня 2022 року

З А В Д А Н Н Я НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Маничу Андрію Михайловичу

1. Тема роботи – «Розробка програми для вивчення і тестування з польської мови».

Керівник роботи: Ракитянська Ганна Борисівна , к.т.н., доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від „24” березня 2022 р. № 66

2. Строк подання студентом роботи 13 червня 2022 р.

3. Вихідні дані до роботи: система управління базами даних – SQLite; мова запитів – SQL; бібліотека для розробки графічного інтерфейсу – Layout Editor; середовище розробки – Android Studio 2022; мова програмування – Java, Kotlin; бібліотека для тестування: JUnit.

4. Зміст розрахунково-пояснювальної записки: вступ; обґрунтування вибору методу розробки та постановка задач дослідження; розробка структури, методу, моделі та алгоритмів роботи програмного продукту; розробка програмного забезпечення; тестування програми; список використаних джерел; додатки.

5. Перелік графічного матеріалу: Титульний слайд, Актуальність теми, Мета, об'єкт та предмет дослідження, Задачі дослідження, Наукова новизна, Практична цінність одержаних результатів, Порівняльний аналіз аналогів, Блок-схема алгоритму створення тестів, Блок-схема алгоритму генерування текстових та графічних компонентів, Тестування програми (створення власних тестів), Тестування програми (проходження тестування), Тестування програми (скидання даних користувача), Тестування програми (звуковий супровід при вивченні нових слів), Апробація та публікації результатів роботи.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Ракитянська Г. Б., к.т.н., доцент кафедри ПЗ		

7. Дата видачі завдання 25 березня 2022 року**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання і вибір методу вирішення поставленої задачі дослідження	26.03.2022 - 29.03.2022	Вик.
2	Вибір архітектури програмного компоненту	01.04.2022 - 09.04.2022	Вик.
3	Розробка структури графічного інтерфейсу „Pollaung”	10.04.2022 - 19.04.2022	Вик.
4	Вибір середовища та мови розробки	20.04.2022 - 24.04.2022	Вик.
5	Розробка модулів програми	27.04.2022 - 11.05.2022	Вик.
6	Тестування програми	11.05.2022 - 15.05.2022	Вик.
7	Оформлення матеріалів до захисту БДР	16.05.2022 - 28.06.2022	Вик.

Студент

(підпис)**Манич А. М.**
(прізвище та ініціали)Керівник бакалаврської дипломної роботи
Б._____
(підпис)**Ракитянська Г.**
(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 107 сторінок формату А4, на яких є 40 рисунків, 4 таблиці, список використаних джерел містить 28 найменувань.

Дипломна робота присвячена розробці програмного забезпечення для вивчення і тестування з польської мови. Тема роботи є актуальною у зв'язку зі складною політичною ситуацією у нашій країні та чисельністю біженців, які залишають нашу країну на деякий час.

Встановлено об'єкт, предмет, завдання та методи дослідження. Система призначена для вивчення польської мови за короткий термін з ефективним результатом та зручним для користувача методом тестування. Для реалізації мети було розроблено додаток для ефективного вивчення іноземної мови. Запропоновано метод та алгоритм вивчення польської мови, який полягає у вивченні та тестуванні в середовищі додатку для Android.

Створений додаток розроблено з використанням мови програмування Java, бібліотек Android Java для роботи із Android додатками, середовища розробки Android Studio. Для розробки бази даних використана мова програмування SQL та система управління базами даних MySQL. У результаті виконання бакалаврської дипломної роботи розроблено програмний засіб, працездатність і правильність роботи якого перевірено, підготовлена інструкція користувача.

Ключові слова: андроїд, вивчення мови, навчання.

ABSTRACT

The bachelor's thesis consists of 107 A4 pages, which contain 40 figures, 4 tables, the list of sources used contains 28 titles.

Thesis is devoted to the development of software for learning and testing in the Polish language. The topic of the work is relevant due to the difficult political situation in our country and the number of refugees who leave our country for a while.

The object, subject, tasks and research methods are established. The system is designed to study the Polish language in a short time with an effective result and a user-friendly testing method. To achieve this goal, an application for effective foreign language learning was developed. A method and algorithm for learning the Polish language, which consists of learning and testing as an application for Android, is proposed.

The created application is developed using the Java programming language, Android Java libraries for working with Android applications, Android Studio development environment. SQL programming language and MySQL database management system were used to develop the database. As a result of the bachelor's thesis, a software tool was developed, the efficiency and correctness of which was checked, and a user manual was prepared.

Keywords: android, language learning, study.

ЗМІСТ

ВСТУП	8
1 ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ	12
1.1 Аналіз предметної області	12
1.2 Аналіз аналогів розроблюваного програмного продукту.....	15
1.3 Аналіз методів вирішення завдання.....	21
1.4 Постановка задач роботи	23
1.5 Висновки	23
2 РОЗРОБКА СТРУКТУРИ, МЕТОДУ, МОДЕЛІ ТА АЛГОРИТМІВ РОБОТИ ПРОГРАМНОГО ПРОДУКТУ	24
2.1 Розробка інтерфейсу за допомогою Layout Editor.....	24
2.2 Розробка структури графічного інтерфейсу „Pollaung”	26
2.3 Розробка методу роботи навчальної системи та алгоритму створення тестів.....	29
2.4 Розробка моделі роботи навчальної системи.....	32
2.5 Висновки	33
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	34
3.1 Варіантний аналіз та обґрунтування вибору мови програмування ...	34
3.2. Вибір середовища розробки та СКБД.....	36
3.3 Програмна реалізація додатку	40
3.4 Висновки	53
4 ТЕСТУВАННЯ ПРОГРАМИ.....	54
4.1 Аналіз методів тестування програмного забезпечення	54
4.2 Тестування розробленого програмного продукту	56
4.3 Розробка інструкції користувача.....	63
4.4 Вимоги до пристрою.....	70
4.5 Висновки	70
ВИСНОВКИ.....	72

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	73
ДОДАТКИ.....	76
Додаток А – Технічне завдання.....	77
Додаток Б – Перевірка на плагіат.....	81
Додаток В – Лістинг програми.....	82
Додаток Г – Графічна частина.....	101

ВСТУП

Обґрунтування вибору теми дослідження. Зараз, як ніколи, стрімкий ріст інформаційних та комп'ютерних технологій обумовлює розвиток і поширення інформаційних ресурсів. Мобільні пристрої, а з ними і мобільні додатки, досить міцно увійшли в життя сучасної людини. В сьогоденні важко уявити людину, яка б замість того, щоб шукати якусь інформацію в інтернеті, буде відкривати для цього підручник або енциклопедію. Оскільки програми для вивчення мови стають все доступнішими, люди, які хочуть вивчити нову мову, звертаються за допомогою до своїх мобільних пристроїв. В інтернеті існує множина платних і безкоштовних програм для вивчення майже всіх мов світу. Ці програми дозволяють користувачам вивчати нову мову з будь-якого місця в будь-який час, як онлайн, так і офлайн. Також вони є досить ефективними для тих, хто вивчає нову мову для подорожей та спілкування на простому рівні, оскільки в таких програмах часто є слова або ж фрази, необхідні для туристів, які знаходяться в іншій країні вперше.

У багатьох школах та університетах вже застосовують додатки для вивчення іноземної мови замість книжок. Діти, які вивчають мови, вивчають їх граючи, тому такий спосіб навчання є більш ефективнішим і перспективним. Творці додатків та ігор часто додають у них необхідність освоєння деякого обсягу знань для проходження рівнів.

„Незважаючи на те, що мільйони людей у всьому світі вже користуються програмами для вивчення мови, опублікованих досліджень щодо їхнього впливу на розмовні навички немає,” – говорить Шон Луевен, професор кафедри лінгвістики та германських, слов'янських, азіатських та африканських мов Університету штату Мічиган. Тому він провів власні дослідження, в яких його учні вивчали нову мову за допомогою телефонного додатку. „Загалом учні в цьому дослідженні підвищили свій рівень володіння усною мовою, що вимірюється покращенням на добре встановленому та дійсному тесті на мовленнєвих інтерв'ю», – зазначає Луевен [1]. Ці результати встановлюють, що використання додатків для вивчення мов може сприяти розвитку навичок

усного спілкування, а не лише засвоєння граматики та словникового запасу. Серед висновків звіту зазначено, що 59% учасників покращили рівень володіння усною мовою принаймні на один підрівень за шкалою рівня володіння мовою американської ради з викладання іноземних мов. Частка учнів, які покращили свої знання, зросла ще вище серед тих, хто більше часу використовував додаток. Серед тих, хто навчався щонайменше 6 годин, 69% підвищили принаймні один підрівень, а серед тих, хто навчався щонайменше 15 годин, покращили свій рівень 75% респондентів [1].

На жаль, у наш час вивчення інших мов – це вже є необхідною потребою, оскільки у зв'язку зі складною політичною ситуацією у нашій країні та чисельністю біженців, які залишають нашу країну на деякий час, їм обов'язково потрібно знати іншу мову. Крім того, необхідне наукове спілкування. Саме тому розробка власного додатку для вивчення іноземних мов за допомогою тестування та прослуховування найважливіших слів, а також їх опрацювання є доволі актуальною задачею.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

Мета та завдання дослідження. Метою дослідження є підвищення ефективності вивчення користувачем польської мови шляхом розробки та використання спеціалізованої мобільної навчальної системи з теоретичним матеріалом і тестами, що дозволить підвищити рівень володіння новою мовою.

Відповідно до поставленої мети в бакалаврській дипломній роботі потрібно вирішити такі **задачі**:

- 1) визначити найбільш ефективний метод вивчення іноземної мови;
- 2) розглянути можливі шляхи вирішення поставленої задачі;
- 3) розробити метод і модель навчальної системи;
- 4) провести варіантний аналіз та обґрунтувати вибір програмного забезпечення для реалізації розроблюваного програмного засобу.
- 5) розробити інтерфейс власного програмного продукту;

- 6) розробити модулі із завданнями;
- 7) реалізувати аудіо супровід при вивченні мови;
- 8) розробити алгоритм генерування тестів для вивчення мови;
- 9) розробити алгоритм генерування текстових та графічних компонентів для зручного вивчення нових слів та словосполучень;
- 10) провести тестування розробленого програмного продукту.

Об'єкт дослідження – процеси створення мобільної навчальної системи для вивчення польської мови.

Предмет дослідження – засоби програмної реалізації мобільного застосунку для вивчення польської мови.

Методи дослідження. У процесі розробки використовувалися:

- методи комплексного аналізу для дослідження переваг і недоліків існуючих програмних реалізацій аналогів та формування вимог до розроблюваної системи;
- методи аналізу та синтезу для побудови ключових моделей автоматизованої системи;
- методи моделювання роботи мобільної навчальної системи та її окремих блоків у вигляді блок-схем для формалізації алгоритмів роботи системи;
- методи побудови компонентів системи та їхніх взаємозв'язків для створення загальної структури збереження та обробки інформаційних ресурсів;
- методи розробки Android додатків для створення навчальної системи з вивчення польської мови;
- методи тестування для підтвердження працездатності системи та її відповідності заданим вимогам.

Новизна отриманих результатів.

1. Подальшого розвитку отримав метод створення мобільної навчальної системи для вивчення польської мови, який, на відміну від існуючих, реалізує логічні зв'язки між компонентами для синхронізації

модульних блоків з теоретичним і словниковим матеріалом, аудіо супроводом і тестами, що підвищує ефективність процесу вивчення нової мови.

2. Подальшого розвитку отримала модель мобільної навчальної системи для вивчення польської мови, яка, на відміну від існуючих, орієнтована на покращенні навчального процесу шляхом реалізації комплексного функціоналу для вивчення основ польської мови з поєднанням словникового матеріалу, аудіо супроводу та тестів, що дозволить реалізувати універсальний підхід до розробки доступного навчального Android додатку і забезпечить тренувальні потреби користувача в процесі вивчення нової мови.

Практична цінність отриманих результатів. Практична цінність одержаних результатів полягає у створенні мобільної навчальної системи для вивчення польської мови, що дозволить опановувати нову мову користувачам Android застосунків.

Особистий внесок здобувача. Усі наукові результати, викладені у бакалаврській дипломній роботі, отримані автором особисто. У науковій праці [2], опублікованій у співавторстві, автору належить порівняльна характеристика реляційних і NoSQL баз даних; у науковій праці [3], опублікованій у співавторстві, автору належить розробка мобільної навчальної системи для вивчення й тестування з польської мови.

Апробація результатів роботи. Результати роботи доповідалися на 2 конференціях:

- на XLIX науково-технічній конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2020);
- на Всеукраїнській науково-практичній інтернет-конференції "Молодь в науці: дослідження, проблеми, перспективи - 2022", Секція - Інформаційні технології та комп'ютерна інженерія.

Публікації. Основні результати досліджень опубліковано в 2 наукових працях у збірниках матеріалів конференцій [2-3].

Аналіз. У пояснювальній записці до бакалаврської дипломної роботи було розглянуто 4 розділи та використано 28 літературних джерел.

1 ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз предметної області

Додатки для вивчення мови дуже популярні в таких магазинах додатків як playmarket та appstore по всьому світу, кажуть що вони революціонізують вивчення мови. Програми для вивчення мов пропонують можливості для практики граматики і можуть бути дуже корисним способом вивчення лексики. Але велися дискусії про те, наскільки ефективними можуть бути такі програми, особливо коли мова йде про інші навички, такі як письмо та розмова.

Найпопулярніша програма для вивчення мов „Duolingo” зробило звіт, в якому описується статистика їхнього додатку. В їхньому звіті було розписано доведено що з кожним роком популярність додатків для вивчення іноземних мов росте. Звіт знаходиться на рисунку 1.1 [4].

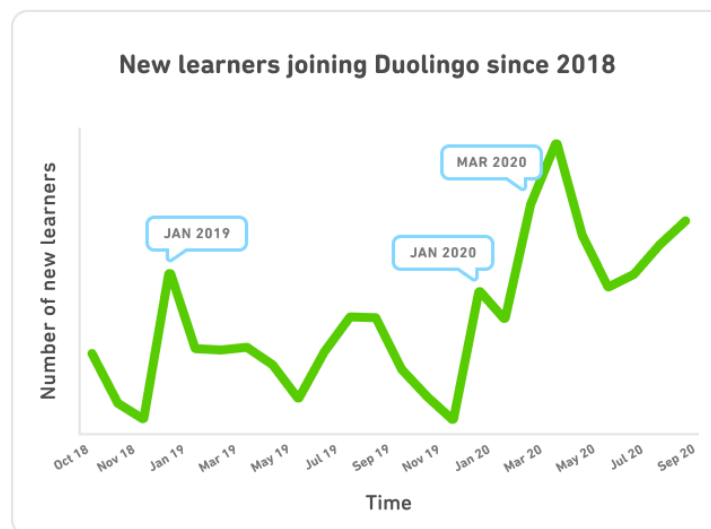


Рисунок 1.1 – Діаграма зростання нових користувачів в додатку Duolingo

Більшість викладачів мови визнають, що використання програми для вивчення мови має певну цінність. Вивчення нової мови вимагає регулярних, багаторазових вправ, а програми пропонують зручний і доступний спосіб навчання для учнів. Більшість програм є доступними безкоштовно, але з вибраною рекламою чи рекламними повідомленнями. Уроки прості, короткі,

захоплюючі та веселі, а гейміфікована функціональність змушує користувачів повертатися до даних додатків і не переставати вчити нову мову.

Також компанія яка розробляє додаток „Duolingo” у своєму звіті збило статистику, та показало для чого люди використовують додатки для вивчення мов. Статистика зображена на рисунку 1.2 [4].

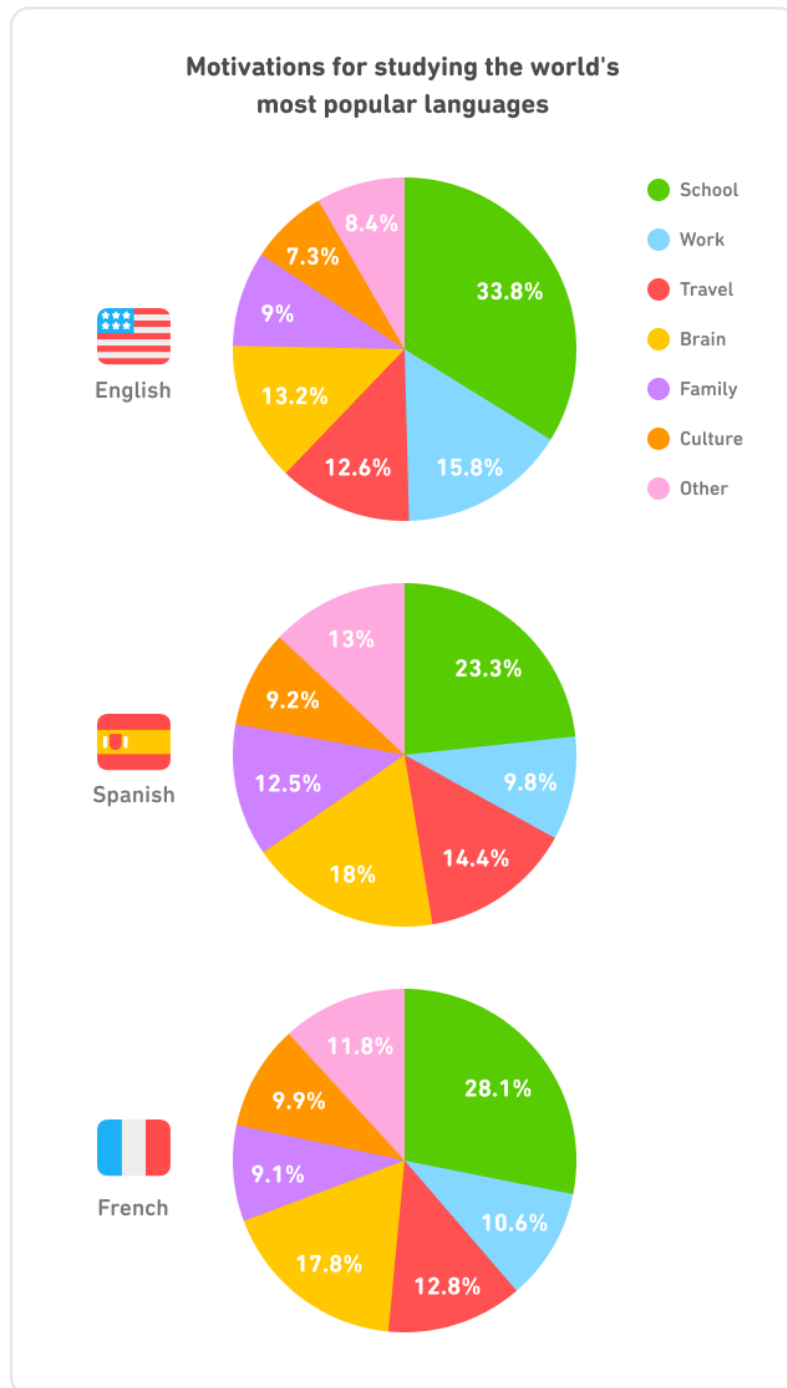


Рисунок 1.2 – Діаграма цілей використання додатку для вивчення мов Duolingo

Крім того, програми створені для зручності та призначені для використання, коли учні мають кілька вільних хвилин, наприклад, під час поїздки на роботу чи під час обідньої перерви. Ці короткі сплески регулярної активності можуть допомогти розширити знання цільової мови та допомогти створити основний словниковий запас і навички аудіювання та читання. Тому вони можуть бути корисними, щоб дати учням уявлення про те, яким буде вивчення їх нової мови, перш ніж витратити багато грошей на курс, який їм не подобається.

Таким чином, програми для вивчення мов, безумовно, можуть бути корисними, щоб дати студентам фору для тих мов, які мають іншу систему письма, наприклад, арабську чи китайську. Переглядаючи відгуки вчителів в Інтернеті, мовні програми можуть зіграти позитивну роль у зменшенні втрати навчання під час канікул або між навчальними роками. Регулярне спілкування з мовою може змусити студентів не забувати її, поки вони знаходяться далеко від офіційного мовного класу. Студенти також можуть, як повідомляють викладачі, зручно вивчати ключові фрази для використання під час подорожей.

Викладачі іноземних мов у нашому вузі, також зауважують, що учні, які використовують програми для вивчення мови, часто приходять до класу з додатковими запитаннями про слова та фрази, які вони зустрічали у своїй програмі, але які ще не були розглянуті на уроці. Більше того, деякі учні цінують той факт, що їхні помилки спілкування відомі лише програмі та їм самим. Це може допомогти подолати занепокоєння, яке іноді вражає деяких студентів під час вивчення нової іноземної мови.

Звичайно, ефективність мовних додатків була темою, яка викликала великий інтерес у академічних дослідженнях. Дослідження 2020 року, опубліковане Duolingo, стверджує, що студенти, які навчалися на своїх курсах іспанської та французької мови, виконували тести з читання та аудіювання так само добре, як і студенти, які проходили чотири семестри університетських занять — і приблизно за половину часу [5].

Отже, додатки для вивчення іноземних мов більш частіше знаходяться в телефонах людей та набувають у наш час все більшої популярності завдяки простоті використання, зрозумілості, та доступності, що досягається завдяки тому, що вони інтегровані на швидке та ефективно навчання користувача.

1.2 Аналіз аналогів розроблюваного програмного продукту

З появи карантину та коронавірусу програми для вивчення мов збільшили свою популярністю, оскільки користувачі шукали нові способи заповнити свої дні чимось цікавим. Але вивчення мови не тільки допомагає вбивати час, але й стає чудовим способом досліджувати світ з дому, тренувати свій мозок та відкривати нові можливості роботи.

Крім того, відчуття задоволення, яке можна отримати від спілкування новою мовою, може підвищити впевненість, яку важко досягти майже кожній людині. Існує багато програм для вивчення іноземних мов. Також вони діляться на категорії:

- 1) програми для вивчення граматики;
- 2) програми для вивчення нових слів;
- 3) програми для спілкування між людьми які знають іноземні мови;
- 4) програми для слухання або ж читання іноземних книг;
- 5) програми для тестування з іноземної мови.

Для порівняльної експертизи програм для вивчення іноземних мов було обрано такі існуючі програми: „Duolingo”, „Speechling”, „iTalki”, „Tandem”.

Duolingo — це програма для вивчення мови, яка використовує ігрові методи, щоб заохочувати користувачів повертатися, щоб вчитися щодня. Duolingo — чудовий приклад простої мовної програми [5].

Він простий у використанні. Потрібно лише створити профіль, обирати цільову мову, встановити цілі на тиждень і виконувати план. Кожен курс в Duolingo складається з модулів, які згруповані для формування навичок. Дана програма дозволяє проходити тестування з вивчених тем, а також переходити на різні рівні тестувань. Вона підійде як для початківців, так і для більш

професійних користувачів, які вже трішки знають мову, яку хочуть вивчати. Інтерфейс додатку зображено на рисунку 1.3.

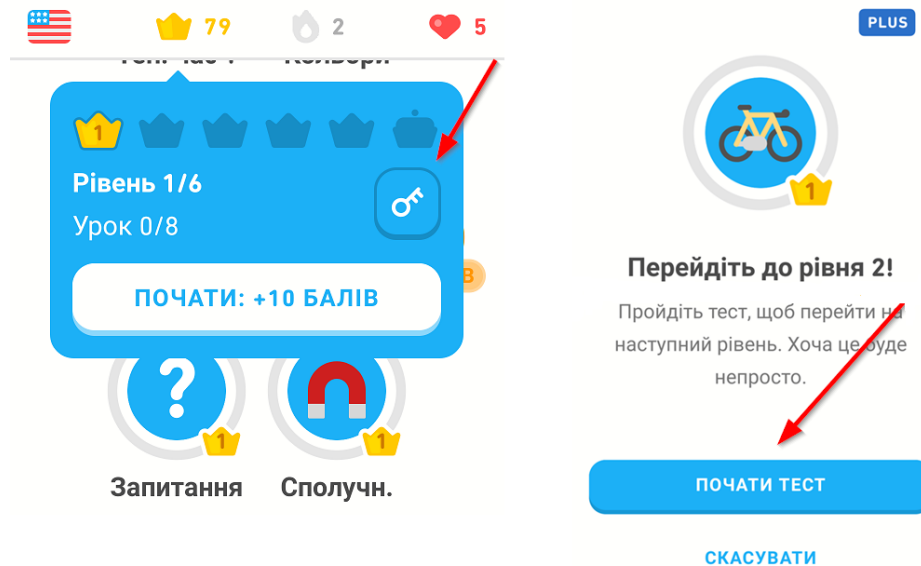


Рисунок 1.3 – Інтерфейс додатку Duolingo

І хоча дана програма є найпопулярнішою в світі, в неї всерівно є недоліки. Наприклад деякі речення є неприродними. Це особливо видно з прогресом, коли гравець вивчає складніші граматичні структури. Деякі речення, які програма змушує перекладати, зазвичай не використовуються в натуральній мові. Однак їхня суть полягає в тому, щоб проілюструвати певні мовні концепції.

У додатку дуже багато реклами, в тому числі для Duolingo Plus. Це трохи дратує і відволікає від вивчення мови. Недостатньо контролю над темами словника. Нові уроки та модулі активуються в додатку лише після того, як завершиться попередній урок. Тому іноді потрібно вивчити словниковий запас, який не обов'язково відповідає цілям навчання. Користувачі хотіли б мати більше контролю над вибором слів, які мають для них більше значення та які їм знадобляться більше.

Speechling — це організація, що базується в США [6], яка намагається вирішити проблему відсутності безкоштовних і доступних інструментів, щоб

люди могли максимально ефективно вивчати інші мови. Команда переконана, що кожен у світі заслуговує на рівний доступ до мовної освіти, тому вони зробили всю свою платформу безкоштовною. Хоча є можливість перейти до преміум-версії приблизно за 15 фунтів стерлінгів на місяць, що дає необмежений доступ до зворотного зв'язку з репетиторами один на один. За таку високу ціну можна вважати, що щедра безкоштовна версія — абсолютна крадіжка. Інтерфейс додатку зображено на рисунку 1.4.

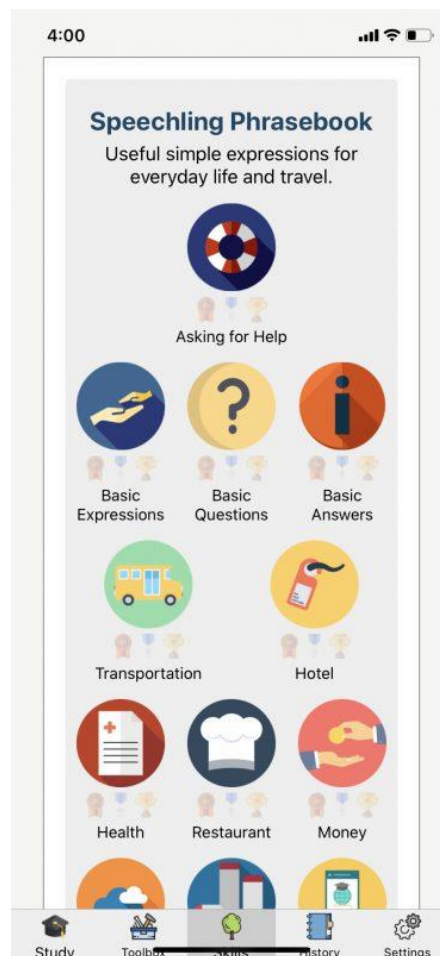


Рисунок 1.4 – Інтерфейс додатку Speechling

Це унікальне рішення фокусує увагу на розмові без тиску розмови в реальному житті. Починаючи з прослуховування корисних фраз і речень, записаних носіями мови, а потім з можливістю записати себе, повторюючи їх. Якщо спробувати дану функцію, можна отримали запис приблизно через 12 годин із виправленнями через голосову нотатку. Гадаю що можна вважати, що

цей метод чудово підійде для початківців, які хочуть отримати впевненість, тоді як учні середнього рівня можуть використовувати його, щоб покращити свій словниковий запас і вимову. Окрім мовленнєвої діяльності, у наборі інструментів Speechling є додаткові вправи для роботи над іншими мовними навичками: диктування, аудіювання, вікторина з вибором варіантів та картки.

iTalki — це платформа для репетиторів, яка дозволяє легко знайти доброзичливого репетитора, який допоможе вивчати омріяну користувачем мову. Цей додаток виштовхує користувача із зони комфорту та покращує його словниковий запас, а також перевірить на скільки він знає іноземну мову.

У додатку можна спробувати репетитора зі знижкою для користувачів які тільки почали користуватись додатком. Сеанси створені як вільні розмови про все що вибере учень, тобто він сам може обрати на яку тему він хоче говорити та розвивати свої навички. Також можливо принести додаткові матеріали, або репетитор може запропонувати статті чи вправи для спільного виконання. Інтерфейс додатку зображено на рисунку 1.5.

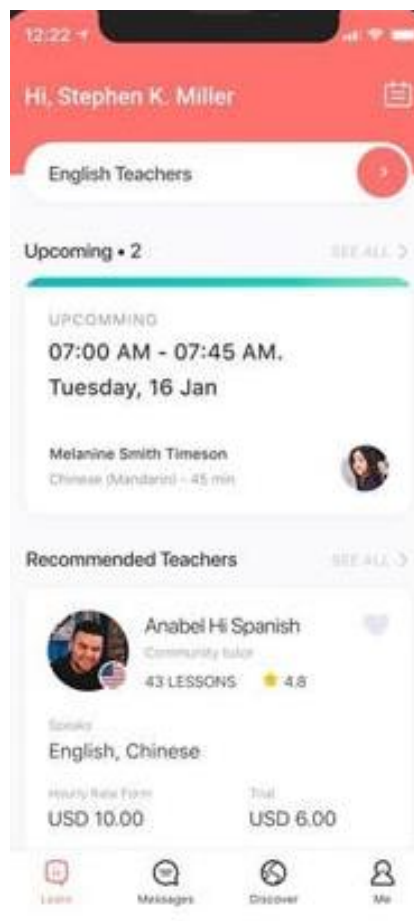


Рисунок 1.5 – Інтерфейс додатку iTalki

Недоліком додатку є те що займатись можна лише одну годину. Але також година вільної розмови іноземною мовою може бути виснажливою, і ніяк не можна зменшити або ж збільшити час уроку.

Тож якщо користувач трохи несміливий і не бажає говорити, це може бути невідповідним варіантом для нього, оскільки додаток не має більше функцій для того щоб вчити обрану користувачем мову.

Tandem — це безкоштовна програма, яка допоможе безкоштовно налаштувати мовний обмін з людьми з усього світу. Даний додаток простий у використанні, але так і як Tinder, він може здаватися більше схожим на додаток для знайомств, ніж на навчальний додаток.

Так само, як і програми для знайомств, доведеться трохи попрацювати, щоб відсіяти хороше від поганого. Однак, коли вже почнеться спілкування, тоді можна легко отримати реальний досвід з носіями мови. Приєднатися можна безкоштовно, хоча існує рекламаний „період очікування”. Та все ж його можна вимкнути, якщо підтвердити свою особу офіційно. Більшості користувачам достатньо менше 24 годин, щоб отримати схвалення на те щоб продовжувати користування додатком без жодних реклам. Інтерфейс додатку зображено на рисунку 1.6 [7].

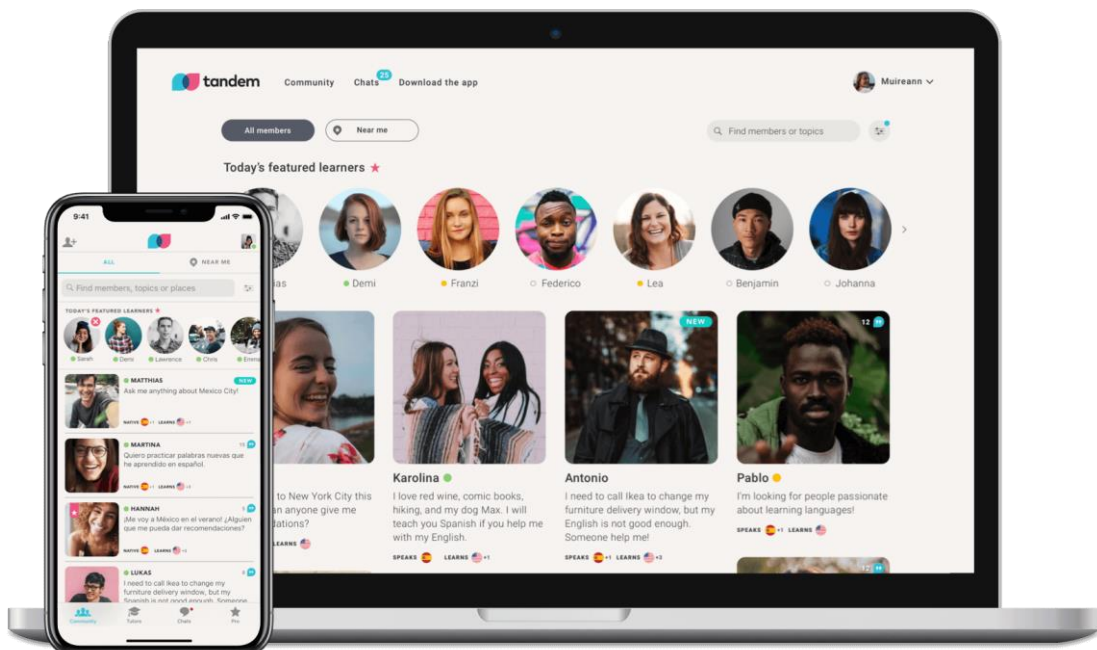


Рисунок 1.6 – Інтерфейс додатку Tandem

Під час процесу адаптації є чіткі попередження про наслідки будь-якої неправильної поведінки. Ці правила зводяться до: будьте поважними, не фліртуйте та не розсилайте спам — або очікуйте, що вас забанять. Також не можна надсилати аудіо або відеоповідомлення нікому до того, як користувач вперше обміняється текстовими чатами.

Користувачам пропонується вказати кілька слів у своєму профілі про свої інтереси, яку мову чи аспект мови вони хочуть вивчати, і про що вони хочуть говорити (або з ким вони хочуть поговорити). Завантаження селфі також є обов'язковою умовою.

Основна відмінність від інших перерахованих програм полягає в тому, що Tandem — це не програма для вивчення мови, а програма для практики. Якщо хтось не володіє базовими мовними навичками, Tandem не стане йому в нагоді. Але якщо хтось навчається одночасно в іншій програмі або хоче оновити свої попередні мовні навички, це зручний безкоштовний ресурс.

Після аналізу усіх аналогів визначено їхні переваги та недоліки та проведено порівняння із розроблюваним додатком для вивчення польської мови під назвою “Pollaung”. Результат порівняння зведено в таблицю 1.1.

Таблиця 1.1 – Порівняльні характеристики програмних продуктів

Критерій	Duoling o	Speechli ng	iTalki	Tandem	Pollaung
Тестування вивчаємої мови	1	1	0	0	1
Словник зі словами	0	0	0	0	1
Безкоштовність повного функціоналу	0	1	0	0	1
Аудіовідтворення іноземних слів	1	1	0	0	1
Спілкування з іншими учасниками	0	0	1	1	0

Підсумковий результат	2	3	1	1	4
-----------------------	---	---	---	---	---

В результаті порівняння існуючих аналогів було зроблено висновок, що розробка власного програмного додатку для вивчення польської мови є доцільною. В результаті розробки отримаємо продукт, який покриває недоліки існуючих аналогів та забезпечує порівняно вищу ефективність та більший обсяг функціоналу.

1.3 Аналіз методів вирішення завдання

Існує кілька методів для вивчення іноземних мов. Одним з таких є репетитор або ж вчитель іноземної мови. Але зараз професія вчителя іноземної мови стає складнішою, ніж будь-коли, через проблеми з низьким рівнем володіння та поганим рівнем результативності серед студентів.

Сьогодні програми, які допомагають вивчати іноземні мови все більше заміщають вчителів. Та все ж головна проблема, а саме немотивовані учні залишається. Тому декілька років тому було вигадано, як заохотити учнів до вивчення нових мов.

Edutainment — це слово, яке вказує на суміш розваг і освіти (Educational entertainment). Основною метою цієї програми є підтримка освіти та розваг. Edutainment використовувався як класична формула у створенні навчальних ігор засновані на теоріях навчання з 1970-х років [8].

Гейміфікація — це використання елементів ігрового процесу, таких як змагання, підрахунок балів і дотримання правил, для стимулювання взаємодії[8]. Він став широко використовуваним інструментом навчання. Наприклад, деяке освітнє програмне забезпечення змушує студентів пройти один рівень навчання, перш ніж перейти до наступного. Edutainment сприяє навчанню, а гейміфікація винагороджує учнів на кожному кроці. Разом вони підвищують мотивацію та стимулюють навчання.

Першою людиною, яка запропонувала ідею edutainment, є Роберт Хейман з Американського національного академічного союзу географії [9]. В той час як

Девід Бекінгем, який є експертом з масової освіти в Англії сказав, що концепція edutainment, яка потребує наочного матеріалу, — це стиль навчання, змішаний з грою. Основна мета Edutainment — залучити увагу студента та змусити його зосередитися на подіях та навчальних матеріалах під час проходження різних рівнів гри. Тобто студент який буде проходити рівні в додатку для вивчення різних мов, буде одночасно вчити мову та грати в цікаву для нього гру, з ігровим інтерфейсом. Взаємозв'язок Edutainment і навчання на основі ігор зображено на рисунку 1.7 [10].

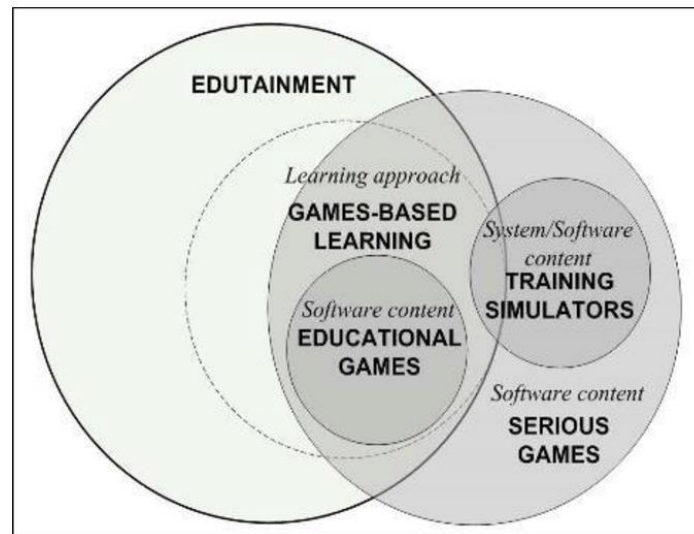


Рисунок 1.7 – Взаємозв'язок Edutainment і навчання на основі ігор, навчальні ігри, тренажери та типові ігри

Немає ієрархічної системи освіти, коли мова йде про навчання з програми [11]. Завдяки мобільним освітнім додаткам будь-яка людина може отримати доступ до будь-якої інформації, навіть якщо вона не входить до їхньої програми.

Освітні програми не тільки покращують систему освіти у світі, але й роблять їхніх студентів розумнішими. Це найбільш синхронізований спосіб навчання, і він також доступний для всіх, як багатих, так і бідних [12]. Поєднання навчання з розвагою може сприяти більшій взаємодії зі користувачами. Гейміфікація володіє силою перетворювати навчання на розвагу, що призводить до кращої взаємодії юзерів. Щоб підвищити рівень

концентрації в учнів, важливо додати гейміфікацію [13]. Таким чином, можна сказати, що бум гейміфікації дасть нові переваги онлайн-навчанню.

Отже, ігри більше не є лише способом розваги студентів, коли вони хочуть провести вільний час, а життєво важливий інструмент, який учні використовують для досягнення свого розумового зростання.

1.4 Постановка задач роботи

Після проведення аналізу питання розробки Android додатку для тестування та навчання іноземної мови, було визначено наступні завдання, які необхідно виконати для розробки програмного продукту:

- 1) визначити найбільш ефективний метод вивчення іноземної мови;
- 2) розглянути можливі шляхи вирішення поставленої задачі;
- 3) розробити метод і модель навчальної системи;
- 4) провести варіантний аналіз та обґрунтувати вибір програмного забезпечення для реалізації розроблюваного програмного засобу.
- 5) розробити інтерфейс власного програмного продукту;
- 6) розробити модулі із завданнями;
- 7) реалізувати аудіо супровід при вивченні мови;
- 8) розробити алгоритм генерування тестів для вивчення мови;
- 9) розробити алгоритм генерування текстових та графічних компонентів для зручного вивчення нових слів та словосполучень;
- 10) провести тестування розробленого програмного продукту.

Технічне завдання на розробку наведено в додатку А.

1.5 Висновки

У першому розділі було розглянуто стан питання існуючих програмних додатків для вивчення та тестування з іноземної мови на сьогоднішній день. Також було проведено аналіз існуючих аналогів та проведено їх порівняння між собою та розроблюваним додатком „Pollaung”. У результаті доведено доцільність розробки бакалаврської дипломної роботи, а також проведено

аналіз існуючих підходів до вирішення поставленої задачі. Було встановлено основні завдання, які необхідно виконати для розробки програмного продукту.

2 РОЗРОБКА СТРУКТУРИ, МЕТОДУ, МОДЕЛІ ТА АЛГОРИТМІВ РОБОТИ ПРОГРАМНОГО ПРОДУКТУ

2.1 Розробка інтерфейсу за допомогою Layout Editor

Layout Editor – це один із інструментів дизайну в Android Studio, який дає змогу розробникам попередньо переглядати та створювати макети на кількох пристроях [14].

Редактор макетів особливо потужний під час створення нового макета за допомогою Layout метода. За допомогою цього інструмента можливий попередній перегляд макетів для різних розмірів екрана або зразки даних, які дозволяють редагувати текст або зображення за допомогою цих зразків [14].

За допомогою редактора макетів Android Studio можливо створювати макети, перетягуючи компоненти на екран замість того, щоб писати макет XML вручну (див. рисунок 2.1).

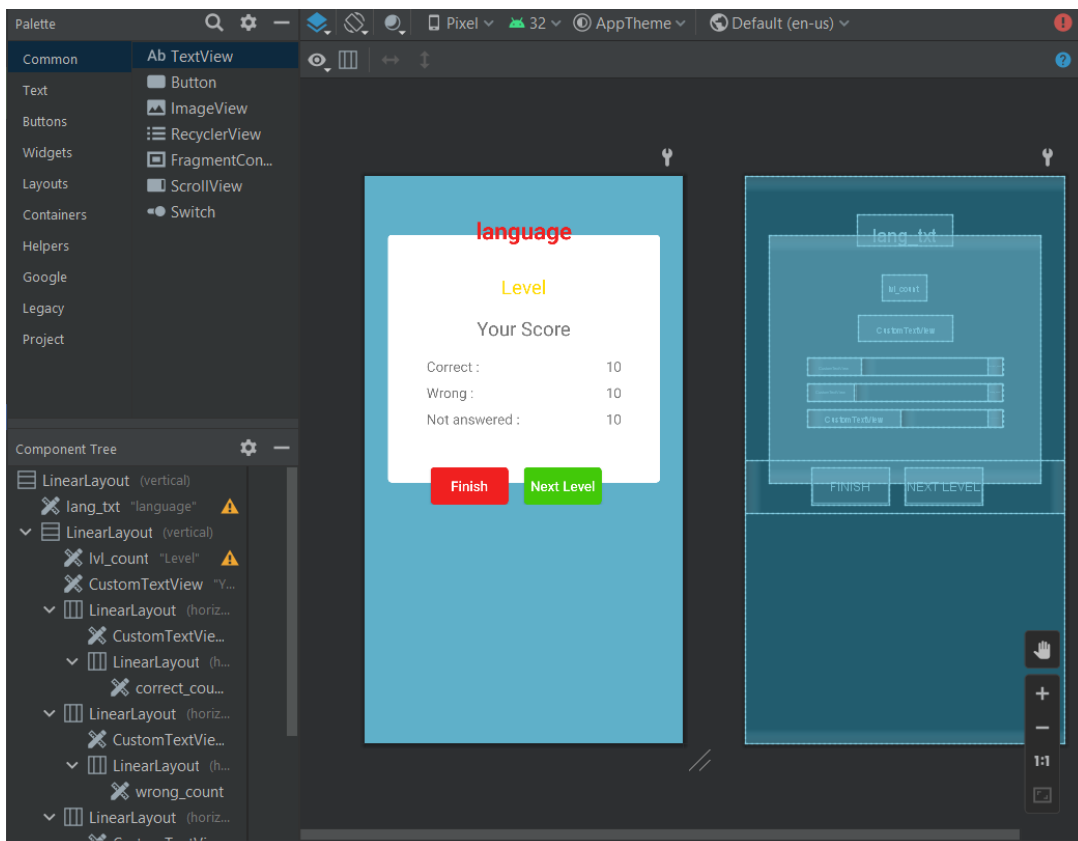
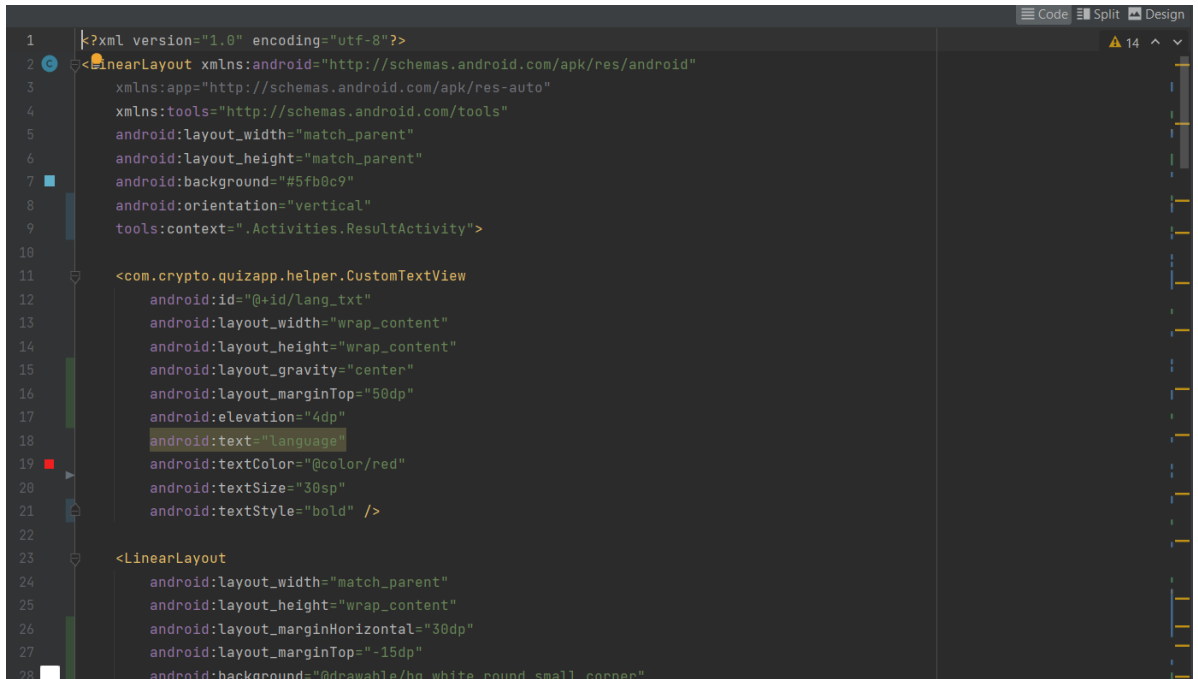


Рисунок 2.1 – Створення інтерфейсу додатку за допомогою перетягування компонентів

Та все ж для програмістів з досвідом роботи доступна така функція, тому вони можуть просто перемикати перегляд і редагувати XML за допомогою текстового редактора (див. рисунок 2.2).



```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:background="#5fb9c9"
8      android:orientation="vertical"
9      tools:context=".Activities.ResultActivity">
10
11     <com.crypto.quizapp.helper.CustomTextView
12         android:id="@+id/lang_txt"
13         android:layout_width="wrap_content"
14         android:layout_height="wrap_content"
15         android:layout_gravity="center"
16         android:layout_marginTop="50dp"
17         android:elevation="4dp"
18         android:text="language"
19         android:textColor="@color/red"
20         android:textSize="30sp"
21         android:textStyle="bold" />
22
23     <LinearLayout
24         android:layout_width="match_parent"
25         android:layout_height="wrap_content"
26         android:layout_marginHorizontal="30dp"
27         android:layout_marginTop="15dp"
28         android:background="@drawable/bg_white_round_small_corner"

```

Рисунок 2.2 – Створення інтерфейсу додатку за допомогою написання XML-коду

За допомогою Layout Editor в Android Studio можливо [15]:

- 1) додавати компоненти до макета за допомогою перетягування;
- 2) додавати та змінювати усі атрибути (властивості) компонентів;
- 3) переглядати та редагувати обмеження для макетів;
- 4) підготувати адаптивний дизайн із попереднім переглядом;
- 5) редагувати XML-код для макета.

Даний функціонал є достатнім для написання додатку для вивчення польської мови.

Редактор дизайну відображає макет додатку у комбінації переглядів дизайну та креслення. Більше того, це основна частина Android Studio Layout Editor, і тут можна бачити, усе що було зроблено візуально [16].

Отже для розробки даної задачі найбільш зручним інструментом буде Layout Editor. Саме за допомогою цього інструменту буде проводитись розробка інтерфейсу додатку.

2.2 Розробка структури графічного інтерфейсу „Pollaung”

При розробці інтерфейсу додатку для вивчення польської мови були реалізовані такі інтерфейси, як інтерфейс головного вікна разом з вкладкою меню, інтерфейс вікна проходження і показу результатів тестування.

На рисунках 2.3 та 2.4 зображені розроблені структурні схеми інтерфейсів вікон програмного додатку.

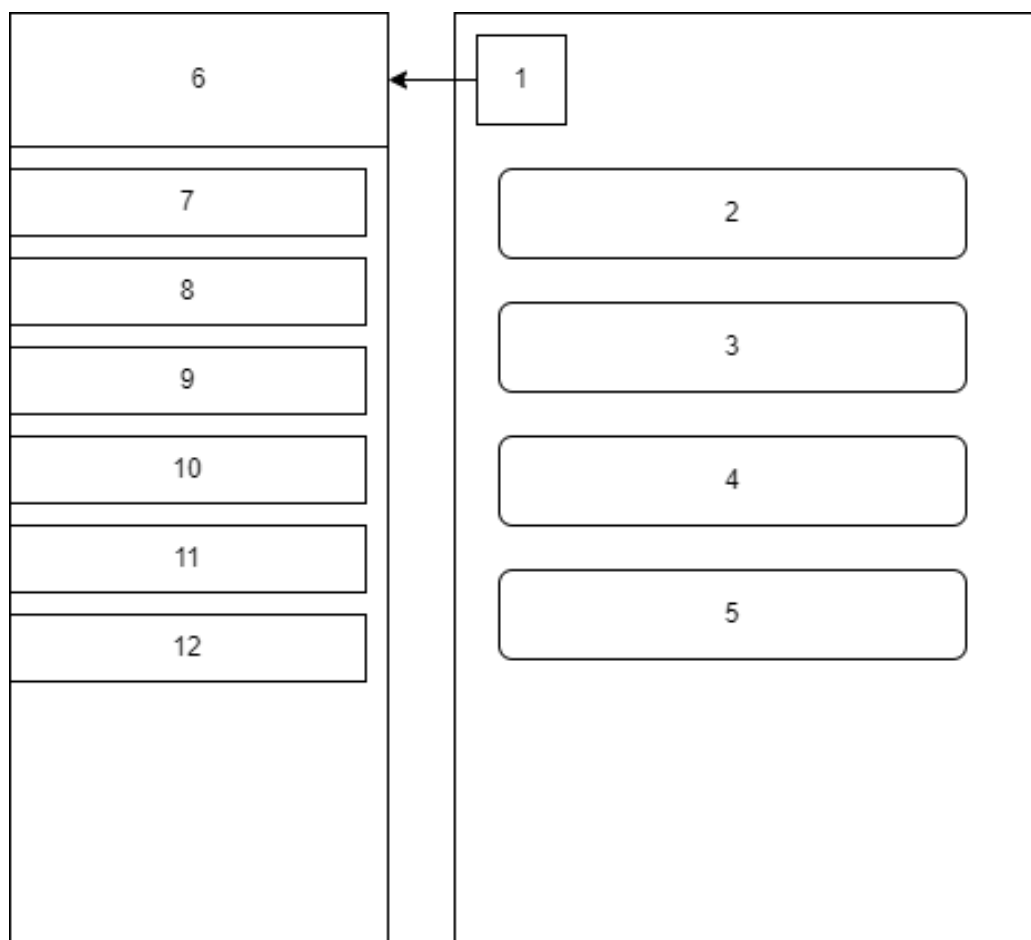


Рисунок 2.3 – Графічна схема головного вікна додатку

Основні елементи інтерфейсу головного вікна додатку «Pollaung»:

1. Кнопка „Menu”.
2. Кнопка „Пройдення тестування”.
3. Кнопка „Пройдення тестування”.
4. Кнопка „Пройдення тестування”.
5. Кнопка „Пройдення тестування”.
6. Назва додатку.
7. Пункт «Головна сторінка».
8. Пункт «Інформація».
9. Пункт «Очистити всі дані».
10. Пункт «Вивчення матеріалів».
11. Пункт «Створити власний тест».
12. Пункт «Вилогуватись».

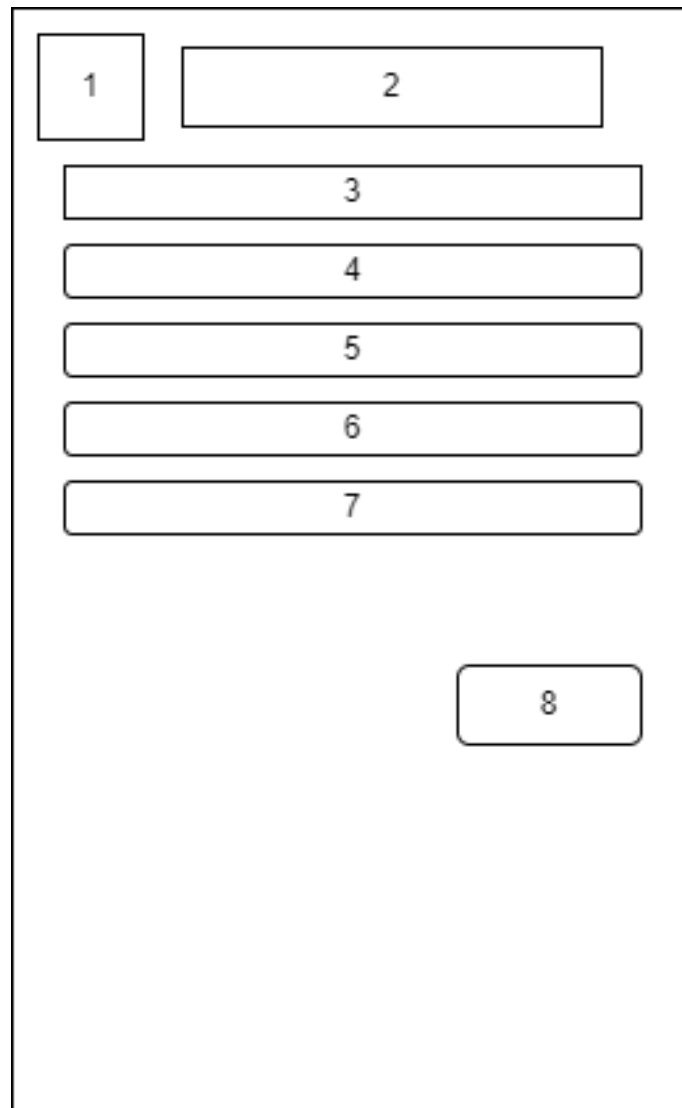


Рисунок 2.4 – Графічна схема вікна тестування

Основні елементи інтерфейсу вікна інформації про бота:

1. Кнопка „Повернутись назад”.
2. Поточний рівень тесту.
3. Запитання.
4. Кнопка „Відповідь”.
5. Кнопка „Відповідь”.
6. Кнопка „Відповідь”.
7. Кнопка „Відповідь”.
8. Кнопка „Підсумувати відповіді”.

Розробка інтерфейсу є найважливішим у процесі розробки додатку, оскільки переважна більшість користувачів в першу чергу звертають увагу на інтерфейс додатку, яким користуються, і віддають перевагу більш зручним у використанні та структурно зрозумілим програмам.

Також в даному підрозділі було описано вікно головного інтерфейсу додатку та вікно з тестуванням з польської мови.

2.3 Розробка методу роботи навчальної системи та алгоритму створення тестів

Важливою частиною розробки додатку для вивчення польської мови є розробка методу роботи системи з генеруванням та додаванням в БД тестів для вивчення мови.

Алгоритм генерування тестів наведено на рисунку 2.5.

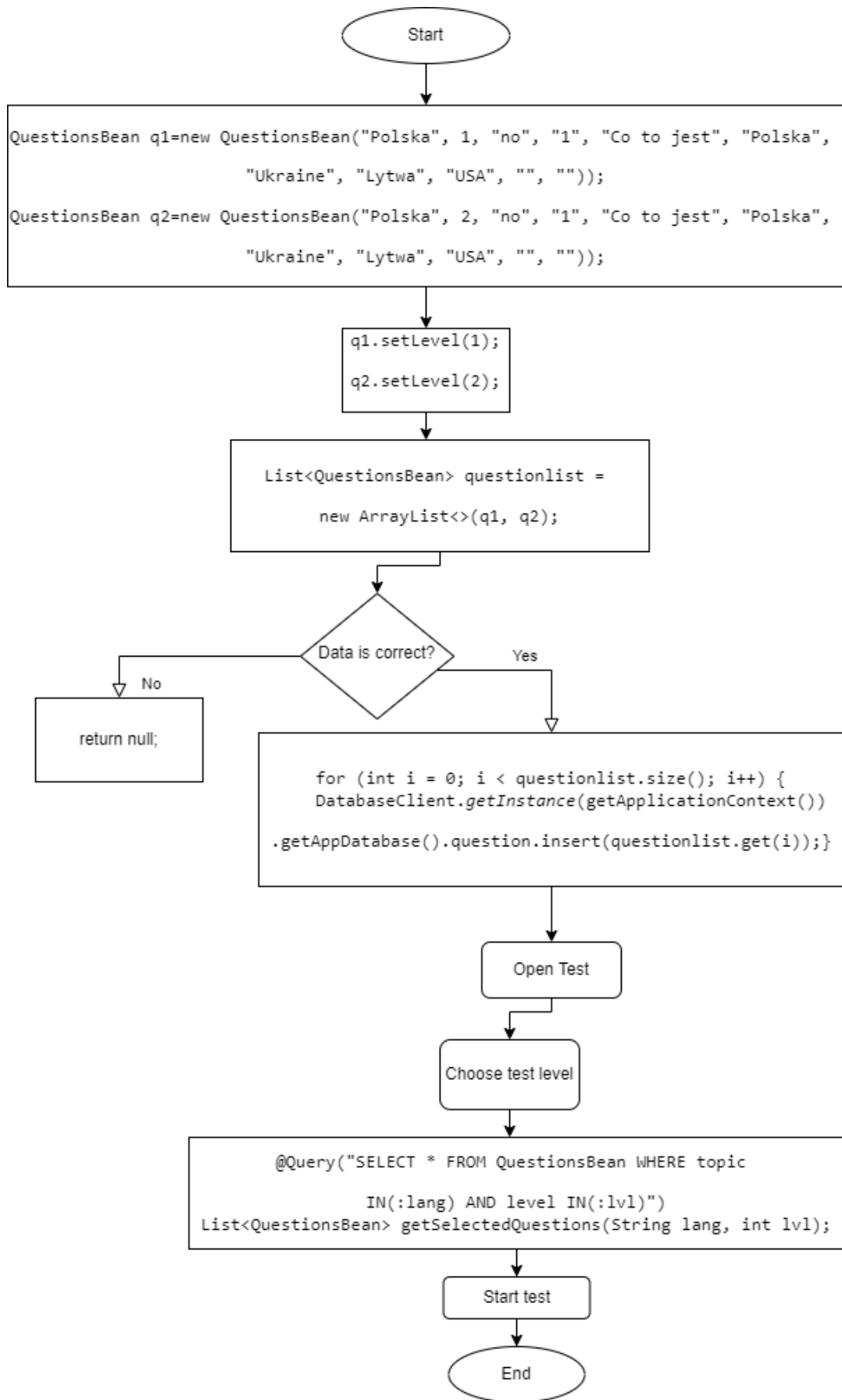


Рисунок 2.5 – Блок-схема алгоритму генерування та додавання в БД тестів

Метод роботи навчальної системи включає послідовність дій:

1. Авторизувавшись у системі, користувач отримує доступ до навчальних матеріалів, до блоків нових слів, до аудіо підтримки з озвучення слів і до блоку створення тестів, що включають в себе блоки змінних тесту.

2. На початку процесу створення тесту деякі блоки уже містить першу характеристику тесту та відповідні йому варіанти відповідей.

3. Після заповнення всіх полів тесту потрібно призначити об'єкту тест його рівень складності.

4. Згодом викликається метод зберігання даних до масиву.

5. Зберігання даних до бази даних. Якщо дані коректні – перейти до пункту 7, якщо ні – відображується повідомлення про помилку збереження тесту.

6. Зберегти дані до бази даних.

7. Для проходження новоствореного тесту потрібно використати функцію тестування.

8. Обиравши тему та рівень тестування, можна побачити створений тест.

9. Далі можна спостерігати отримування даних з бази даних та їх відображення за допомогою функції рандому (також використовується функція, яка спочатку фільтрує питання по рівнях).

10. Після чого починається початок тестування та відображення всіх тестів.

11. Збереження результатів тестування.

Запропонований метод створення мобільної навчальної системи для вивчення польської мови реалізує логічні зв'язки між компонентами для синхронізації модульних блоків з теоретичним і словниковим матеріалом, аудіо супроводом і тестами, що підвищує ефективність процесу вивчення нової мови.

2.4 Розробка моделі роботи навчальної системи

Для того щоб краще зрозуміти взаємодію користувача із системою для вивчення та тестування польської мови, було прийнято рішення створити модель роботи за допомогою метода User Flow [17] (рис. 2.6).

Згідно з діаграмою першим кроком користувача є вибір з головного меню методів роботи програми. Наступний крок користувача після вибору методу програми є проходження завдання, яке стосується обраного методу та показ результатів після проходження.

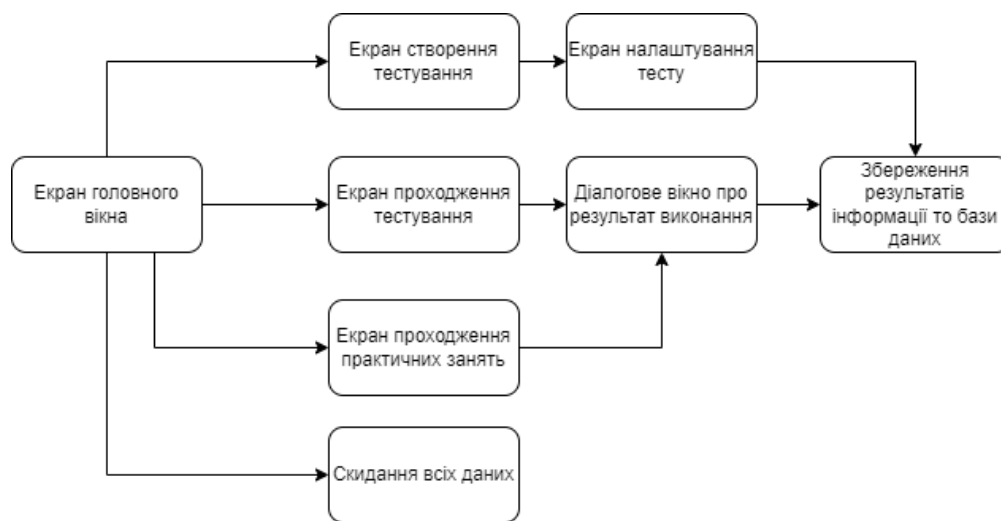


Рисунок 2.6 – Модель роботи навчальної системи

Як показано у моделі, при вході до web-системи користувачу потрібно зареєструватися або ж авторизуватися, якщо він уже має власний акаунт. При реєстрації нового акаунта дані користувача вносяться до бази даних.

При некоректному ввводі даних для реєстрації система виводить спеціальне повідомлення з вказанням на недопустиме значення. Аналогічно, при невдалій авторизації система виводить відповідне повідомлення.

Після успішної реєстрації/авторизації користувач переходить до навчальних матеріалів, де також має можливість перевірити свої знання за допомогою тестування.

2.5 Висновки

У другому розділі було розглянуто створення інтерфейсу додатку за допомогою Layout Editor. Було описано всі переваги Layout Editor для розробки додатку.

Було проаналізовано структуру графічного інтерфейсу „Pollaung”. Побудовано графічні схеми для головного вікна та меню, а також графічна схема інтерфейсу тестування.

Також було розроблено алгоритм створення та запису тестів до бази даних, який було представлено у вигляді блок-схем.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Варіантний аналіз та обґрунтування вибору мови програмування

З початком розробки будь-якого програмного забезпечення дуже важливим є питання вибору мови програмування. Для розробки даного додатку необхідно, щоб дана мова підходила для розробки під Android смартфони. Цю здатність підтримують такі мови як C++, C#, Java та Python. Розглянемо особливості кожної з них.

C++ – це компільована, апаратна мова. Він виконується після компіляції безпосередньо на ЦП (центральному процесорі) з прямим доступом до пам'яті та пристроїв [18].

Існує міф, що C++ більше не використовується, і це є правдою. Багатьох програмістів лякає зусилля, необхідне для того, щоб стати компетентним у C++. Інші мови стають модними, частково через цей страх.

C++ швидко падає та не розвивається, чому свідчить всесвітня статистика розвитку мов програмування серед програмістів. Дана мова мало використовується для розробки Android додатків, тому що вона використовується більше для розробки Windows. Існує дуже мала кількість бібліотек C++, для написання Android додатку.

C# – це сильна об'єктно-орієнтована мова, яка постачається з багатою стандартною бібліотекою, захищаючи програмістів від власної дурості, коли це можливо [19]. Це позбавляє функцій C та C++, таких як ручне управління пам'яттю та здебільшого доступ до низького рівня.

У C# є потужна система типів, а також збір сміття для позбавлення від невикористаної пам'яті. C# є внутрішньою частиною платформи .NET, тому сервер, на якому запускається програма, має базуватися на Windows. Це значно обмежує розробку на даній мові. C# менш гнучкий, оскільки переважно залежить від платформи .Net.

Python – це мова скриптів (scripting language). Він має обмежене використання, погано масштабується та має обмежені можливості. Коли

програмісти використовують Python для обчислювальних програм, таких як шифрування, машинне навчання або розпізнавання зображень, їх Python код використовує бібліотеки [20], написані більш швидкими мовами, такими як Fortran, C або C++.

Є багато “scripting language” (наприклад, Lua), які насправді кращі за Python. І хоч Python запускає програму швидше, але виконує код програми набагато повільніше. Вивчати Python легко, та все ж впровадження та використання Python може зайняти багато часу.

Найбільша проблема даної мови, це те що Python виконується динамічно, тобто python є інтерпретованою мовою, тому доводиться зустрічати більше помилок під час виконання.

Java – це суто об’єктно-орієнтована мова, також можна сказати що це комбінація C та C++ [21]. Більшість функцій C++ вже є в Java. Java використовує компілятор та інтерпретатор, тоді як C та C++ використовує лише компілятор.

Основна перевага Java – це підтримка веб-додатків та розробка під Android. Існує досить багато різних бібліотек та API для розробки Android додатків, написаних саме на Java. Це надійна, розповсюджуюча та захищена мова, а також підтримка Java в мережі, де це неможливо в C або C++.

На відміну від C++, під час компіляції Java вона не компілюється на платформу, а в незалежний від платформи байт-код (файли .class). Ці файли .class можуть виконуватися будь-якою операційною системою, будь-якою платформою, на якій запущено програму JVM.

Таким чином, програми, розроблені та скомпільовані на java на одній машині, можна виконувати на будь-якій іншій машині, якщо для цієї машини встановлено JVM.

Для програмістів, вже знайомих з поняттям модуля, абстрактними типами даних та об’єктно-орієнтованим програмуванням забере мінімальних зусиль, щоб опанувати дану мову.

Проаналізувавши мови програмування, було складено таблицю (табл. 3.1), яка демонструє відмінності Java, Python, C#, C++.

Таблиця 3.1 – Функціональні характеристики мов програмування

	Java	Python	C#	C++
Кросплатформеність	1	0	0	0
Автоматична збірка сміття	1	0	1	0
ООП	1	1	1	1
Поліморфізм	1	0	1	0
Швидкість роботи	1	0	0	1
Обсяг коду	1	1	0	0
Сумарний коефіцієнт	6	2	3	2

Проаналізувавши дані таблиці, Java краща за C++ та Python на 66,7% ($100\% - 2/6 * 100\% = 66,7\%$) та на 50% ($100\% - 3/6 * 100\% = 50\%$) краща ніж C#.

Отже, проаналізувавши відмінні ознаки Java, Python, C# та C++, було прийнято рішення, що Java найбільше підходить для виконання поставленої задачі.

3.2. Вибір середовища розробки та СКБД

З розвитком цифрових технологій актуальною стала проблема в зберіганні та обробці великих об'ємів даних, які невідомо зростають. Так, за прогнозами експертів міжнародної дослідницької компанії IDC вже за період з 2018 по 2023 рік обсяг усієї світової інформації, що зберігається на електронних носіях, повинна подвоїтись і становитиме близько 11,7 зеттабайт (1021 байт) [1].

Об'єм дискової пам'яті, необхідний для рішення типових задач у корпоративному сегменті, як правило, не перевищує 72 Тбайт (1012 байт),

однак для вирішення задач сфери великих даних необхідно до 300 Пбайт (1015 байт).

Згідно з іншим дослідження компанії IDC до 2025 року близько 80% усієї світової інформації становитиме корпоративна інформація, а тому зростатиме і відповідальність компаній за захист даних. Крім того, компанії висувають підвищені вимоги й до швидкості обробки даних. Таким чином, для задоволення зростаючих потреб користувачів необхідно обирати продуктивну модель організації баз даних.

За даними ресурсу реляційна модель даних все ще займає провідне місце, однак тренди свідчать про поступове зростання ролі нереляційних NoSQL баз даних, що зумовлено швидким розвитком сегменту великих даних, інтернету речей тощо, де дані є слабо структурованими, а головною вимогою є їх швидка обробка.

Реляційна модель даних базується на математичній теорії (теорія множин, теорія відношень, теорія логіки), тоді як нереляційні бази даних (БД) не мають єдиної математичної теорії.

Реляційна модель даних дозволяє забезпечити базі даних надійність, гнучкість, стабільність. Надійність моделі БД визначається за допомогою властивостей ACID (Atomicity, Consistency, Isolation, Durability): -

Нерозривність (Atomicity) найкраще характеризується висловом «все або нічого», тобто будь-яка транзакція виконується повністю, або не виконується взагалі.

Узгодженість (Consistency) передбачає, що будь-яка транзакція повинна зберігати узгодженість БД.

Ізольованість (Isolation) гарантує незалежне виконання транзакцій та недоступність проміжних результатів виконання незавершеної транзакції.

Стійкість (Durability) гарантує постійне зберігання в пам'яті результатів успішно завершених транзакцій, навіть після потенційних збоїв інформація має бути збережена до бази даних.

Коли набір даних організований за допомогою формально описаних таблиць, до елементів такої БД легко отримати доступ, створити нові дані, розширити структуру БД. Сьогодні такий тип зберігання інформації є найпоширенішим.

Помітними перевагами NoSQL бази даних також є простіші процедури шарингу (розподіл інформації по різних вузлах мережі) та реплікації (копіювання даних на інші сервери), які є невід’ємними складовими розподілених систем.

На відміну від реляційних баз даних, процес розробки NoSQL баз даних є простішим, оскільки не потрібно визначати схему даних та зв’язки між даними.

Однак, часто саме на етапі проектування визначаються потенційно слабкі місця обраного технічного рішення, що дозволяє вчасно відкоригувати стратегію розробки. Зображення прикладу реляційної бази даних знаходиться на рисунку 3.1.

ID	Name	Phone	Birthday
1	Andrew	222-31-31	1976
2	Bohdan	264-05-06	1990
3	Ivan	237-01-02	1989

Рисунок 3.1 – Приклад реляційної бази даних

За рахунок відсутності зв’язків між даними, не реляційні бази даних забезпечують значно вищу швидкість пошуку та видачі результатів, порівняно з реляційними, що є без сумніву їх основною перевагою.

Підсумовуючи, з огляду на дане завдання, краще буде використовувати реляційні бази даних, тобто MySQL або ж SQLite. Реляційні бази даних є добре

перевіреним інструментом, в той час як нереляційні бази даних є відносно новими рішеннями.

Нереляційні бази даних є вузькоспеціалізованими, найкраще підходять для організації неструктурованих даних, забезпечують лінійну масштабованість та високу швидкість обробки.

3.3 Програмна реалізація додатку

Для реалізації програмного функціоналу додатку для вивчення та тестування польської мови „Pollaung” було розроблено велику кількість алгоритмів. Основними серед них є наступні:

- 1) алгоритм змінення вікон додатку за допомогою меню;
- 2) алгоритм додавання власних тестів;
- 3) алгоритм заставки програмного додатку;
- 4) алгоритм генерування теоретичних відомостей для навчання;
- 5) алгоритм генерування тестів по теоретичним відомостям.

Першою дією, яку виконує додаток при запуску, є ініціалізація усіх бінів та змінних, що використовуються для початкового запуску програми, також іде запуск і підв'язка бази даних до додатку та заповнення її дефолтними відомостями та запитаннями.

У головному виконавчому файлі „MainActivity.class” викликається функція „onCreate”, яка запускає процес відкриття головного вікна за допомогою методу „super.onCreate(savedInstanceState)” – це стандартний метод запуску програми та встановлення інстансу. Клас „MainActivity.class” є нащадком класу „AppCompatActivity” і може використовувати всі його методи, та переоприділяти їх (рисунок 3.2).

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_home);
    toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    initView(savedInstanceState);
}
```

Рисунок 3.2 – Лістинг функції „onCreate”

Після використання методу „super.onCreate(savedInstanceState)”, викликається метод „setContentView(R.layout.activity_home)” який призначає вікно, яке буде видиме з початку роботи програми, також призначається toolbar, який має бути використовуваним за допомогою методу “setSupportActionBar(toolbar)”.

Автоматично, після запуску додатку та встановлення з’єднання із базою даних, викликається метод „initView(savedInstanceState)”, який налаштовує та синхронізує всі інші властивості програми, на її початку. Лістинг даної функції наведено на рисунку 3.3.

```
private void initView(Bundle savedInstanceState) {
    drawer = findViewById(R.id.drawer_layout);
    navigationView = findViewById(R.id.nav_view);
    drawerToggle = new ActionBarDrawerToggle(
        activity: this, drawer, toolbar, R.string.navigation_drawer_open, "Close navigation drawer");
    drawer.addDrawerListener(drawerToggle);
    drawerToggle.syncState();

    // declaring fragnav controller
    fragNavController = new FragNavController(getSupportFragmentManager(), R.id.content_frame);

    // declaring fragnav transaction options
    fragNavTransactionOptions = new FragNavTransactionOptions.Builder().customAnimations(R.anim.slide_in_from_right,
        R.anim.slide_out_to_left, R.anim.slide_in_from_left, R.anim.slide_out_to_right).build();

    // adding root fragments
    List<Fragment> fragments = new ArrayList<>();
    fragments.add(TopicSelectFrag.newInstance());

    // setting root fragments to fragnav controller
    fragNavController.setRootFragments(fragments);

    // setting fragnav transaction option to fragnav controller
    fragNavController.setDefaultTransactionOptions(fragNavTransactionOptions);

    // setting first fragment to display when we get on Homeactivity
    fragNavController.initialize(FragNavController.TAB1, savedInstanceState);

    //navigation drawer listener to navigate through activity and fragment from drawer
    navigationView.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener() {
```

Рисунок 3.3 – Лістинг функції „initView(savedInstanceState)”

Після виконання даних функцій, запускається головне вікно програми в якому є здатність відкривати меню програми. Реалізація алгоритму роботи меню програми є показана в класі „HomeActivity”. Лістинг даної функції наведено на рисунку 3.4.

```

public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {
    switch (menuItem.getItemId()) {
        case R.id.nav_home:
            if(!fragNavController.isRootFragment()){
                fragNavController.popFragment();
            }else {
                fragNavController.switchTab(FragNavController.TAB1);
            }
            break;

        case R.id.nav_aboutus:
            fragNavController.pushFragment(AboutUsFrag.newInstance());
            break;

        case R.id.nav_reset:
            deleteQuestions();
            break;

        case R.id.nav_add:
            Intent z = new Intent( packageContext: HomeActivity.this, AddQuestionActivity.class);
            startActivity(z);
            break;

        case R.id.nav_logout:
            SharedPreferences.setsharedText( context: HomeActivity.this, key: "logged", value: null);
            Intent i = new Intent( packageContext: HomeActivity.this, SignInActivity.class);
            i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
            startActivity(i);
            finish();
            break;

        case R.id.nav_study:
    
```

Рисунок 3.4 – Лістинг функції „onNavigationItemSelected”

Після запуску даного методу, для реалізації алгоритму роботи меню, було застосовано функцію „switch”, яка приймає значення „menuItem.getItemId()”, це є елемент меню.

Далі в кожному „case”, відловлюються елементи меню, які були передані в дану функцію, та виконуються згідно з їхніми значеннями. Оскільки значення дістаються з масиву елементів меню, то не має необхідності застосовувати дефолтне значення для „case”.

В першому випадку „case R.id.nav_home”, іде перевірка на те чи є переданий елемент, елементом головного вікна, якщо так то далі іде ще одна внутрішня перевірка, на те, чи він є головним фрагментом вікна чи ні. Якщо так, то відкривається головне вікно і воно призначається головним фрагментом,

якщо ж ні та запускається заставка програми, а потім фрагмент заставки, замінюється на фрагмент головного вікна.

Наступний елемент, це „R.id.nav_aboutus”, в цьому випадку відкривається інтерфейс з описом програми та розробника. Елемент „R.id.nav_reset” відповідає за виклик функції „deleteQuestions()”, яка відповідає за скидання всіх даних та налаштувань. Лістинг функції наведено на рисунку 3.5.

```
private void deleteQuestions() {

    class DeleteQues extends AsyncTask<Void, Void, Void> {

        @Override
        protected Void doInBackground(Void... voids) {
            DatabaseClient.getInstance(HomeActivity.this).getAppDatabase()
                .questionDao()
                .deleteAllQuestions();
            return null;
        }

        @Override
        protected void onPostExecute(Void aVoid) {
            super.onPostExecute(aVoid);
            Toast.makeText(context: HomeActivity.this, text: "All data is reset", Toast.LENGTH_SHORT).show();
            addquestions();
            fragNavController.replaceFragment(TopicSelectFrag.newInstance());
        }
    }

    DeleteQues dt = new DeleteQues();
    dt.execute();
}
```

Рисунок 3.5 – Лістинг функції „deleteQuestions()”

Метод „deleteQuestions()” має в собі клас „DeleteQues”, який наслідує клас „AsyncTask”. Даний клас викликає функцію „doInBackground” та „onPostExecute”, ці функції викликаються для того щоб перевірити всі наявні налаштування та дані які були внесені саме користувачем під час роботи з програмою. Далі знайдені дані та налаштування зі сторони користувача видаляться або замінюються на стандартні дані, які були ініціалізовані при першому запуску програми.

Елемент „R.id.nav_add” відповідає за виклик інтерфейсу, а з ним методу додавання власного тесту для користувача. Даний метод створює об’єкт класу „Intent”, в який передає інші класи, які будуть використовуватись в даному інтерфейсі програми: клас „HomeActivity.class” та клас „AddQuestionActivity.class”. Після чого запускається метод „startActivity(intent)”, в який передається екземпляр даного класу.

Елемент „R.id.nav_logout” відповідає за вихід з аккаунту користувача. Після виклику даного елемента за допомогою класу „CommonSharedPreference” відбувається вихід з системи.

Далі інтерфейс головного вікна змінюється на інтерфейс логування в додаток.

Також під час вилогювання з аккаунта, всі дані та налаштування, які були введені попереднім користувачем в його аккаунті зберігаються в базу даних, після чого видаляються з кеш-пам’яті додатку. Лістинг функції наведено на рисунку 3.6.

```

case R.id.nav_logout:
    | CommonSharedPreference.setsharedText(context: HomeActivity.this, key: "logged", value: null);
    Intent i = new Intent(packageContext: HomeActivity.this, SignInActivity.class);
    i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
    startActivity(i);
    finish();
    break;

```

Рисунок 3.6 – Лістинг елемента „R.id.nav_add”

Елемент „R.id.nav_test” відповідає за виклик інтерфейсу з тестами. Після цього викликається клас „WordAdapter”, який відповідає за розташування та вигляд слів та словосполучень.

Даний клас є нащадком класу „ArrayAdapter<Word>”, клас батько приймає в себе клас „Word”, який є класом для опису всіх слів та словосполучень, тобто класом entity.

Лістинг функції класу „WordAdapter” наведено на рисунку 3.7.

```

@OVERRIDE
public View getView(int position, View convertView, ViewGroup parent){
    Word currentWord= getItem(position);

    View listView= convertView;
    if( listView == null)
    {
        listView = LayoutInflater.from(getContext()).inflate(R.layout.listitem, parent, attachToRoot: false);
    }
    TextView miwokTextView = listView.findViewById(R.id.text_view_1);
    miwokTextView.setText(currentWord.getmiwokWordTranslation());
    TextView defaultTextView = listView.findViewById(R.id.text_view_2);
    defaultTextView.setText(currentWord.getDefaultWordTranslation());
    ImageView pictureImageView = listView.findViewById(R.id.imageView);
    int k = currentWord.getImageResourceId();
    if(k!=0)
    {
        pictureImageView.setImageResource(currentWord.getImageResourceId());
    }
    else
    {
        pictureImageView.setVisibility(View.GONE);
    }
    View linear_colour = listView.findViewById(R.id.text_container);
    int color = ContextCompat.getColor(getContext(), mBackgroundColourResource);
    linear_colour.setBackgroundColor(color);
    return listView;
}

```

Рисунок 3.7 – Лістинг класу „WordAdapter”

Спочатку іде ініціалізація поточного слово за допомогою конструктора класу „Word”, після чого перевіряється чи „listView” не є пустий. Якщо ж він не пустий то звідти достаються всі елементи, які там знаходились, і до них додається новостворений елемент.

Далі іде збірка новоствореного елемента, для того щоб він був відображений в „listView”. Спочатку додається переклад слова на українську мову, потім його переклад в польській мові. Потім іде перевірка чи є в даного слова малюнок в базі даних, якщо так, то він додається до слова, якщо ж ні то дається стандартний малюнок. Також після цього додається фон для даного

слова. Після цього новостворене слово додається до масиву, та даний масив повертається і записується в пам'ять.

За додавання власних тестів в додаток відповідає клас „AddQuestionActivity”. Даний клас наслідується від класу „AppCompatActivity” та імплементує інтерфейс „View.OnClickListener”. В цьому класі є безліч змінних, які стосуються тестів. Лістинг змінних класу „AddQuestionActivity” наведено на рисунку 3.8.

```
public class AddQuestionActivity extends AppCompatActivity implements View.OnClickListener {

    private Spinner topic_spinner;
    private Spinner level_spinner;
    private EditText question_edt;
    private EditText option_one_edt;
    private EditText option_two_edt;
    private EditText option_three_edt;
    private EditText option_four_edt;
    private RadioButton option_one_rb;
    private RadioButton option_two_rb;
    private RadioButton option_three_rb;
    private RadioButton option_four_rb;
    private RadioGroup correct_grp;
    private Button add_question_btn;

    private List<String> topics_list;
    private List<String> levels_list;
    private ArrayAdapter<String> topics_adapter;
    private ArrayAdapter<String> levels_adapter;
    private String topic_st, level_st, correct_st, question_no;
```

Рисунок 3.8 – Лістинг класу „AddQuestionActivity”

Даний клас при старті викликає метод „onCreate”, в якому викликається вікно з додаванням тесту. Далі використовується функція для призначення заголовку даного вікна, після чого для даного вікна ставиться можливість перегляду меню для вибору інших його елементів.

Приватний метод „initView” використовується для ініціалізації всіх полів для створення тестів:

- topic_spinner – заголовок тесту;
- level_spinner – рівень тесту;

- question_edt – запитання;
- option_one_edt – перша відповідь;
- option_two_edt – друга відповідь;
- option_three_edt – третя відповідь;
- option_four_edt – четверта відповідь;
- option_one_rb – опис на першу відповідь;
- option_two_rb – опис на другу відповідь;
- option_three_rb – опис на третю відповідь;
- option_four_rb – опис на четверту відповідь;
- correct_grp – правильна відповідь;

Лістинг методу „initView” наведено на рисунку 3.9.

```
private void initView() {  
    topic_spinner = (Spinner) findViewById(R.id.topic_spinner);  
    level_spinner = (Spinner) findViewById(R.id.level_spinner);  
    question_edt = (EditText) findViewById(R.id.question_edt);  
    option_one_edt = (EditText) findViewById(R.id.option_one_edt);  
    option_two_edt = (EditText) findViewById(R.id.option_two_edt);  
    option_three_edt = (EditText) findViewById(R.id.option_three_edt);  
    option_four_edt = (EditText) findViewById(R.id.option_four_edt);  
    option_one_rb = (RadioButton) findViewById(R.id.option_one_rb);  
    option_two_rb = (RadioButton) findViewById(R.id.option_two_rb);  
    option_three_rb = (RadioButton) findViewById(R.id.option_three_rb);  
    option_four_rb = (RadioButton) findViewById(R.id.option_four_rb);  
    correct_grp = (RadioGroup) findViewById(R.id.correct_grp);  
    add_question_btn = (Button) findViewById(R.id.add_question_btn);  
    add_question_btn.setOnClickListener(this);  
    option_one_rb.setChecked(true);  
    option_one_edt.addTextChangedListener(new MyTextWachter(option_one_edt));  
    option_two_edt.addTextChangedListener(new MyTextWachter(option_two_edt));  
    option_three_edt.addTextChangedListener(new MyTextWachter(option_three_edt));  
    option_four_edt.addTextChangedListener(new MyTextWachter(option_four_edt));  
  
    getTopics();  
}
```

Рисунок 3.9 – Лістинг методу „initView”

Після налаштування всіх полів, викликається метод „getTopics()”, він виводить на екран користувача інформацію про те що користувач коректно або не коректно ввів поле заголовок, для створення власно тексту.

Пізніше іде проходження по алгоритму даного методу, на кінці алгоритму викликається вже інший метод „getLevels()”, він виводить на екран користувача інформацію про те що користувач коректно або не коректно ввів поле рівень тесту, для створення власно тексту.

Після проходження по алгоритму методу „getLevels()”, на кінці алгоритму викликається вже інший метод „getQuestions”. В методі „getQuestions” передаються попередньо заповнені поля з заголовком та рівнем тесту. Потім іде проходження по тілі даного методу і читання всіх запитань згідно переданих параметрів.

За перевірку та запис усіх запитань відповідає метод „List<QuestionsBean> doInBackground(Void... voids)”, після завершення він повертає новостворений масив „List<QuestionsBean>”. Лістинг методу „doInBackground(Void... voids)” наведено на рисунку 3.10.

```
@Override
protected List<QuestionsBean> doInBackground(Void... voids) {

    List<QuestionsBean> taskList =
        DatabaseClient
            .getInstance(getApplicationContext()) DatabaseClient
            .getAppDatabase() AppDatabase
            .questionDao() QuestionDao
            .getSelectedQuestions(languagevalue, levelvalue);

    return taskList;
}
```

Рисунок 3.10 – Лістинг методу „initView”

Трішки пізніше після встановлення запитання, встановлюються відповіді на запитання. По стандарту можливо встановити до 4 відповідей на одне запитання, та обрати лише одну вірну відповідь. Це обмеження спростить програму в здатності викидати якісь помилки, а також полегшить завдання користувача відповідати на запитання, та навчатись швидше, що не менш важливо.

Пізніше, по проходженню всіх полів та їх заповненню викликається метод „submit”. Лістинг методу „submit” наведено на рисунку 3.11.

```
private void submit() {  
  
    // validations  
  
    if (TextUtils.isEmpty(question_edt.getText().toString().trim())) {  
        question_edt.setError("Please enter a question");  
        requestFocus(question_edt);  
    } else if (TextUtils.isEmpty(option_one_edt.getText().toString().trim())) {  
        option_one_edt.setError("Please enter value of option 1");  
        requestFocus(option_one_edt);  
    } else if (TextUtils.isEmpty(option_two_edt.getText().toString().trim())) {  
        option_two_edt.setError("Please enter value of option 2");  
        requestFocus(option_two_edt);  
    } else if (TextUtils.isEmpty(option_three_edt.getText().toString().trim())) {  
        option_three_edt.setError("Please enter value of option 3");  
        requestFocus(option_two_edt);  
    } else if (TextUtils.isEmpty(option_four_edt.getText().toString().trim())) {  
        option_four_edt.setError("Please enter value of option 4");  
        requestFocus(option_four_edt);  
    } else {  
        setdata();  
    }  
}
```

Рисунок 3.11 – Лістинг методу „submit”

Цей метод перевіряє чи всі відповіді на запитання введені коректно, якщо ж ні то користувач отримує повідомлення про те, поле якої відповіді введено не коректно та потребує змін.

Якщо ж всі поля з відповідями введені коректно, то викликається новий метод „setdata”, який кінцево створює тест та зберігає його до бази даних. На початку даного методу іде перевірка на правильну відповідь, в якому полі відповіді вона знаходиться. По закінченню перевірки створюється бін „QuestionsBean” з попередньо заповненими даними.

Розглянемо алгоритм генерування тестів по теоретичним відомостям. Для того щоб згенерувати тест, використовується клас „QuizActivity.class”.

Даний клас наслідується від класу „AppCompatActivity.class” та імплементує інтерфейс „View.OnClickListener”. В цьому класі є трішки змінних, які стосуються тестів. Лістинг змінних класу „AddQuestionActivity.class” наведено на рисунку 3.12.

```
public class QuizActivity extends AppCompatActivity implements View.OnClickListener {

    private RecyclerView quiz_Recyclerview;
    private String languagevalue;
    private int levelvalue;
    private Button btn_submit;
    private List<QuestionsBean> questionslist;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_quiz);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);

        levelvalue = getIntent().getIntExtra("lvl", defaultValue: 0);
        languagevalue = getIntent().getStringExtra("lang");
        setTitle("Poziom "+levelvalue); // setting t
        initView();
    }

    private void initView() {
        quiz_Recyclerview = findViewById(R.id.quiz_recyclerview);
        quiz_Recyclerview.setHasFixedSize(true);
        btn_submit = findViewById(R.id.btn_submit);
        btn_submit.setOnClickListener(this);
        getQuestions();
    }
}
```

Рисунок 3.12 – Лістинг класу „QuizActivity.class”

На початку викликається метод „onCreate”, який відповідає за створення та призначення робочого поля. Пізніше іде активація здатності перегляду меню, під час знаходження у інтерфейсі тестування, щоб користувач завжди міг перейти назад до вивчення матеріалів, або ж інших функцій з меню. Також в цьому методі призначаються доступні рівні, які користувач хоче пройти та теми по яким він буде проходити тестування.

Після цього викликається метод для ініціалізації приватного методу „initView()”. В ньому ж викликається публічний метод „getQuestions()”, який нічого не повертає, але виконує функцію збирання всіх запитань із бази даних, за допомогою методу „List < QuestionsBean > doInBackground (Void ... voids)”. Всі запитання для тесту збираються та фільтруються згідно з темою тесту та рівнем тесту. Лістинг методу „List < QuestionsBean > doInBackground (Void ... voids)” наведено на рисунку 3.13.

```
private void getQuestions() {
    // getting all the questions with selected topic and level
    class GetQuestions extends AsyncTask<Void, Void, List<QuestionsBean>> {

        @Override
        protected List<QuestionsBean> doInBackground(Void... voids) {
            List<QuestionsBean> taskList = DatabaseClient
                .getInstance(getApplicationContext()) DatabaseClient
                .getAppDatabase() AppDatabase
                .questionDao() QuestionDao
                .getSelectedQuestions(languagevalue, levelvalue);
            return taskList;
        }

        @Override
        protected void onPostExecute(List<QuestionsBean> questionsBeanList) {
            super.onPostExecute(questionsBeanList);

            // defining the linearlayoutmanager
            LinearLayoutManager layoutManager = new LinearLayoutManager(getApplicationContext());

            // setting linearlayoutmanager to the recyclerview
            quiz_Recyclerview.setLayoutManager(layoutManager);

            // custom adapter for recyclerview
            QuestionsAdapter qadapter =
                new QuestionsAdapter( applicationContext: QuizActivity.this, questionsBeanList);

            // setting the adapter to the recyclerview
            quiz_Recyclerview.setAdapter(qadapter);
        }
    }
}
```

Рисунок 3.13 – Лістинг класу „QuizActivity.class”

Останім розглянемо алгоритм заставки програмного додатку. Для того щоб згенерувати тест, використовується клас „Splashactivity.class”.

Даний клас наслідується від класу „AppCompatActivity.class”. В цьому класі є приватна змінна „Runnable runnable”, яка допомагає імплементувати багатопоточність в даний клас. Лістинг класу „ Splashactivity.class” наведено на рисунку 3.14.

```

ivity.java × Splashactivity.java ×
public class Splashactivity extends AppCompatActivity {
    private ImageView splash;
    private View decorView;
    private Handler handler;
    private Runnable runnable;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splashactivity);
        initView();
    }

    private void initView() {
        splash = findViewById(R.id.splash);
        Animation animation = AnimationUtils.loadAnimation(getBaseContext(), R.anim.right_to_left);
        splash.startAnimation(animation);
        animation.setAnimationListener(new Animation.AnimationListener() {
            @Override
            public void onAnimationStart(Animation animation) {

            }

            @Override
            public void onAnimationEnd(Animation animation) { startHandler(); }

            @Override
            public void onAnimationRepeat(Animation animation) {

            }
        });
    }
}

```

Рисунок 3.14 – Лістинг класу „ Splashactivity.class”

Даний клас має метод „startHandler()” який відповідає за запуск потоку з виводом головної заставки на екран перед запуском програми. Це потрібно для того щоб була проведена перевірка на те, чи користувач вже є

залогований в додаток чи ні, якщо ж є то його одразу переміщують на головну вкладку додатку, а якщо ні то він потрапляє на вкладку з логуванням.

Отже, у підрозділі 3.3 було описано програмний код основних алгоритмів роботи розробленого додатку для вивчення польської мови. Повний лістинг наведено у додатку Б.

3.4 Висновки

У даному розділі було проведено аналіз мов програмування та обґрунтування вибору мови для написання додатку, проведено детальний опис та порівняння середовища розробки для написання додатку з іншими аналогами. Було детально описано вибір системи управління базами даних для вирішення поставленої задачі.

В результаті даного аналізу було прийнято рішення використовувати мову програмування Java, середовище розробки Android Studio та СУБД SQLite для розробки програмного засобу для тестування та вивчення польської мови. Крім того, було проведено детальний опис алгоритмів основних програмних модулів додатку.

4 ТЕСТУВАННЯ ПРОГРАМИ

4.1 Аналіз методів тестування програмного забезпечення

Тестування програмного забезпечення [22] — це процес перевірки системи з метою виявлення будь-яких помилок, прогалин або відсутніх вимог порівняно з фактичною вимогою.

Тестування є основою кожного проекту, воно має місце в кожній частині життєвого циклу розробки системи. Використання поганої методології тестування призведе до виробництва нестабільного продукту, який буде коштувати більше грошей і часу на розробку [23]. Існує два основних види тестування: статичне та динамічне.

Статичне тестування — це вид тестування, при якому тестується якість програмного додатку який розробляється, код якого в процесі тестування не буде перевірятись та виконуватись [24]. Зачасту це перевірка вимог ПЗ або ж перевірка документації розроблюваного програмного додатку. Також даний вид тестування використовують на початковій стадії розробки програмного забезпечення.

Динамічне тестування — це вид тестування програмного забезпечення, мета якого переконатися, що програмне забезпечення працює відповідно до специфікацій і відповідає очікуванням користувачів [25]. Хоча ціль звучить досить просто, завдання включає в себе багато типів функціонального тестування, деяким з яких можна віддати перевагу чи пріоритет над іншими залежно від характеру програми та організації.

Залежно від рівня знання внутрішньої будови додатку, що тестується, виділяють такі види динамічного тестування [26]:

1) Модульні тести — саме вони є частиною процесу розробки програмного забезпечення, в якому невеликі частини програми, які називаються блоками, тестуються окремо. Модульне тестування можна виконувати вручну, але також воно є часто автоматизовано.

2) Тестування „білої скриньки” — це метод тестування програмного забезпечення, який перевіряє внутрішні структури або ж роботу програми, а не її функціональність. Під час тестування „білої скриньки” внутрішня перспектива системи, а також навички програмування необхідні і використовуються для розробки тестових випадків. Тестер вибирає вхідні дані для тесту через код і визначає відповідні вихідні дані. Це аналогічно тестуванню вузлів у ланцюзі, наприклад, внутрішньосхемне тестування.

3) Тестування „чорної скриньки” — це метод тестування програмного забезпечення, який перевіряє функціональність програми на відміну від її внутрішніх структур або роботи. Спеціальні знання коду внутрішньої структури програми та знання програмування загалом не потрібні. Тестові випадки будуються на основі специфікацій та вимог, тобто того, що програма повинна робити. Він використовує зовнішні описи програмного забезпечення, включаючи специфікації, вимоги та дизайн для отримання тестових випадків.

Ці тести можуть бути функціональними або нефункціональними, хоча зазвичай вони є саме функціональними. Конструктор тестів вибирає дійсні та недійсні вхідні дані та визначає правильний вихід. Немає знань про внутрішню структуру об'єкта тестування. Тестування чорної скриньки розглядає програмне забезпечення як „чорну скриньку”, тобто без будь-яких знань про внутрішню структуру коду.

4) Регресійне тестування — це метод тестування коли розробники фіксують новий код або змінюють функцію, яка запускає регресійні тести, щоб переконатися, що програмне забезпечення все ще працює належним чином. Регресійне тестування допомагає підтримувати стабільний продукт під час внесення в нього змін. Регресійні тести часто автоматизовані.

5) Димове тестування — це тестування верифікації, воно включає лише швидкі сценарії, які вказують на тестування основної частини продукту. Це виконується на початкових збірках продукту, перш ніж вони будуть випущені для ретельного тестування. Димове тестування є одним із важливих видів функціонального тестування.

Після визначення особливостей кожного із розглянутих методів тестування програмного забезпечення було вирішено використати методику «чорної скриньки», оскільки використання саме цієї методики тестування додатку для вивчення іноземної мови дозволить виявити помилки в логіці функціонування додатку та можливі неполадки у роботі його основних алгоритмів, чого не завжди можна досягти при використанні інших методів тестування.

4.2 Тестування розробленого програмного продукту

Тестування додатку для вивчення польської мови за методикою «чорної скриньки» передбачає перевірку правильності функціонування додатку при виконанні основних алгоритмів його використання та порівняння фактичного результату виконання із очікуваним. Тестування проводиться під час розробки додатку по технології BDD [27].

Такі алгоритми використання програмного додатку називають тест-кейсами. Для тестування розробленого додатку для вивчення польської мови було розроблено наступні тест-кейси:

Тест-кейс №1 –Проходження тестування:

1. Відкрити додаток “Pollaung”.
2. Зареєструватись або увійти в свій аккаунт
3. Натиснути кнопку ”Numery”.
4. Обрати свій рівень.
5. Обрати відповіді на запитання.

Очікуваним результатом даного тест-кейсу є отримання та вивід інформації з кількістю правильних, не правильних та не зазначених відповідей.

Після проходження рівня у користувача також має бути змога вирішити чи він буде переходити на наступний рівень чи просто закінчити тестування. Перейдемо до тесту та пройдемо його щоб перевірити результат.

Результат виконання тест-кейсу наведено на рисунку 4.1.

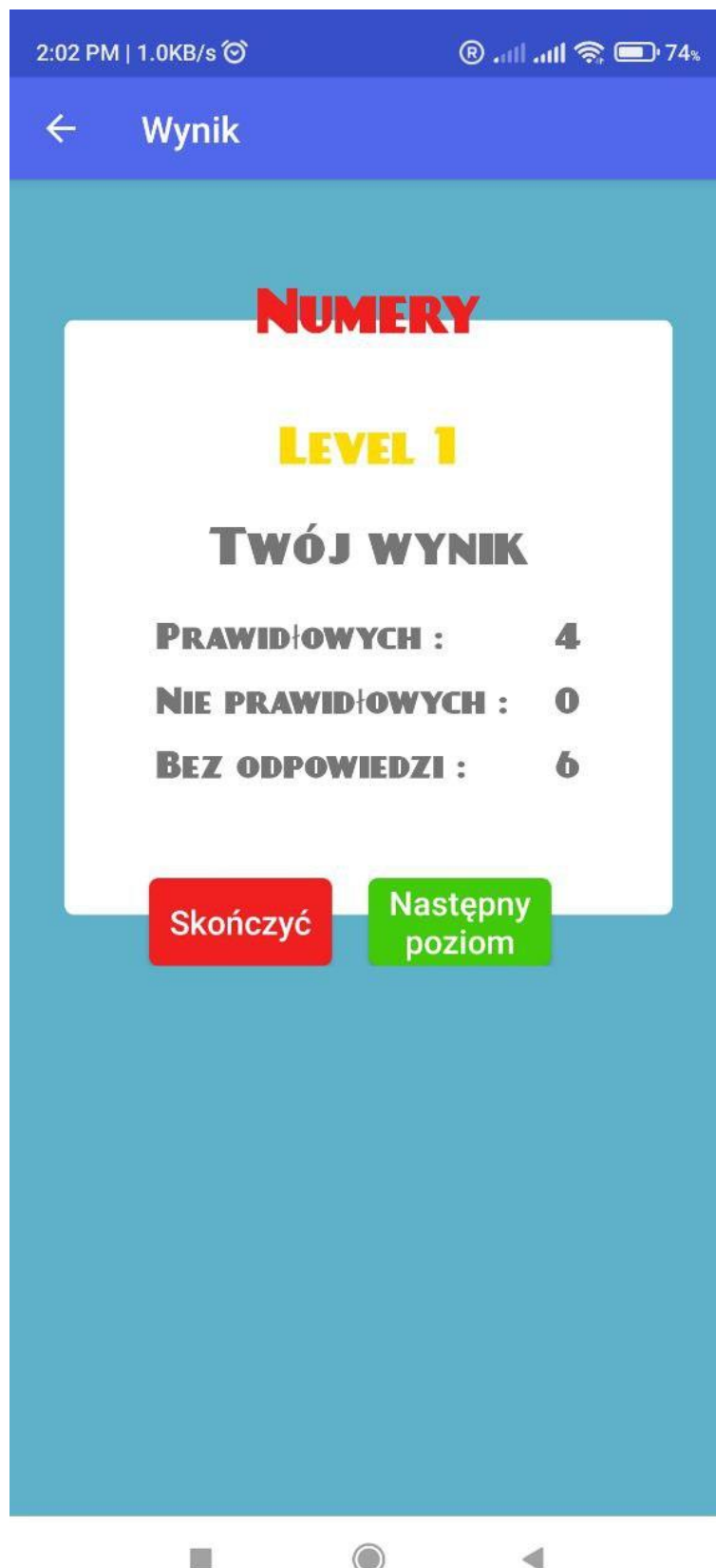


Рисунок 4.1 – Результат виконання тест-кейсу №1

Фактичний результат виконання відповідає очікованому, отже тест-кейс №1 успішно пройдено.

Тест-кейс №2 – Скидання налаштувань з аккаунту користувача:

1. Відкрити додаток “Pollaung”.
2. Зареєструватись або увійти в свій аккаунт
3. Натиснути кнопку “Zresetować dane” на панелі меню.
4. Перезайти в свій аккаунт.

Результат виконання тест-кейсу наведено на рисунку 4.2.

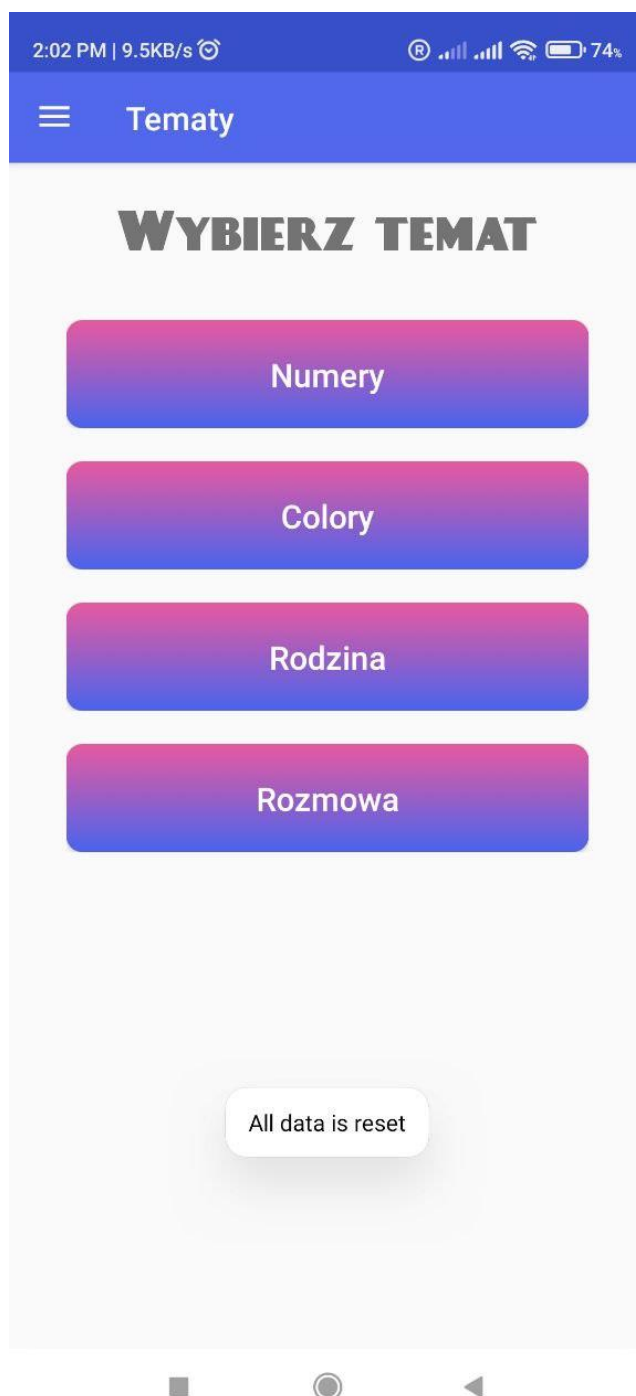


Рисунок 4.2 – Результат виконання тест-кейсу №2

Очікуваним результатом у даному випадку є скидання всіх налаштувань а також видалення результатів проходження тестів разом із створеними користувачем тестів.

Перед викликом даною функції була перевірка чи користувач знаходиться в своєму аккаунті. Після виклику функції на екрані з'являється інформація про те що всі налаштування видалені.

В результаті виконання тест кейсу додаток скинув всі дані які. Тобто, фактичний результат відповідає очікуваному, тому тест-кейс №2 успішно пройдено.

Тест-кейс №3 – Додавання користувачем власного тестування:

1. Відкрити додаток “Pollaung”.
2. Зареєструватись або увійти в свій аккаунт
3. Натиснути кнопку “Dodaj pytanie” на панелі меню.
4. Обрати тему та заповнити всі поля, які повинні бути заповненими.
5. Натиснути кнопку „Dodaj pytanie”

Даний тест є досить важким для тестування, оскільки тут може бути найбільше помилок, тому що саме користувач вводить дані.

Якщо користувач введе не коректні дані, після цього йому на екран має бути відображено повідомлення з інформацією про те що дані які він ввів є не вірними та їх потрібно замінити на інші дані.

Так як у даному тест-кейсі ми створюємо свій тест для обраної теми, новостворений тест з його даними повинен бути доданий до таблиці тестів в базі даних.

Для перевірки чи додався даний тест до бази потрібно зайти в тестування, та перейти до створеного тесту. Відображення тесту під час тестування свідчить що програма працює коректно.

Створення нового тесту разом з коректно заповненими даними наведено на рисунку 4.3. Результат виконання тест-кейсу наведено на рисунку 4.4.

Створення нового тесту разом з некоректно заповненими даними наведено на рисунку 4.5.

2:03 PM | 0.3KB/s

← Dodaj pytania

Tematy :

Numery

Poziomy :

Level 1

Co to?

jest

sous

wiem

my

Poprawna odpowiedź :

jest

sous

wiem

my

Dodaj pytanie

Рисунок 4.3 – Коректно заповнені дані

Q.11 Co to?

jest

sous

wiem

my

SKŁAŚĆ

Рисунок 4.4 – Результат виконання тест-кейсу №3

11:36 PM | 0.0KB/s | 54%

← Dodaj pytania

Tematy :

Numery

Poziomy :

Level 1

jdjsjsj

sjjs

jsjs

jdjs

Odpowiedz 4

Poprawna odpowiedź: Please enter value of option 4

sjjs

jsjs

jdjs

Dodaj pytanie

Рисунок 4.5 – Некоректно заповнені дані

В результаті виконання тест-кейсу новостворений тест додався до бази даних та успішно відображається при проходженні тестування. Тобто, фактичний результат відповідає очікованому, тому тест-кейс №3 успішно пройдено.

Тест-кейс №4 – Аудіосупровід при вивченні слів:

1. Відкрити додаток “Pollaung”;
2. Зареєструватись або увійти в свій аккаунт;
3. Натиснути кнопку “Czwiczenia” на панелі меню;
4. Обрати тему та натиснути на слово біля якого є значок програшу.

Очікуваним результатом у даному випадку є отримання від додатку голосового супровіду вивчення слова, тобто правильне звучання даного слова. Результат виконання тест-кейсу наведено на рисунку 4.6.

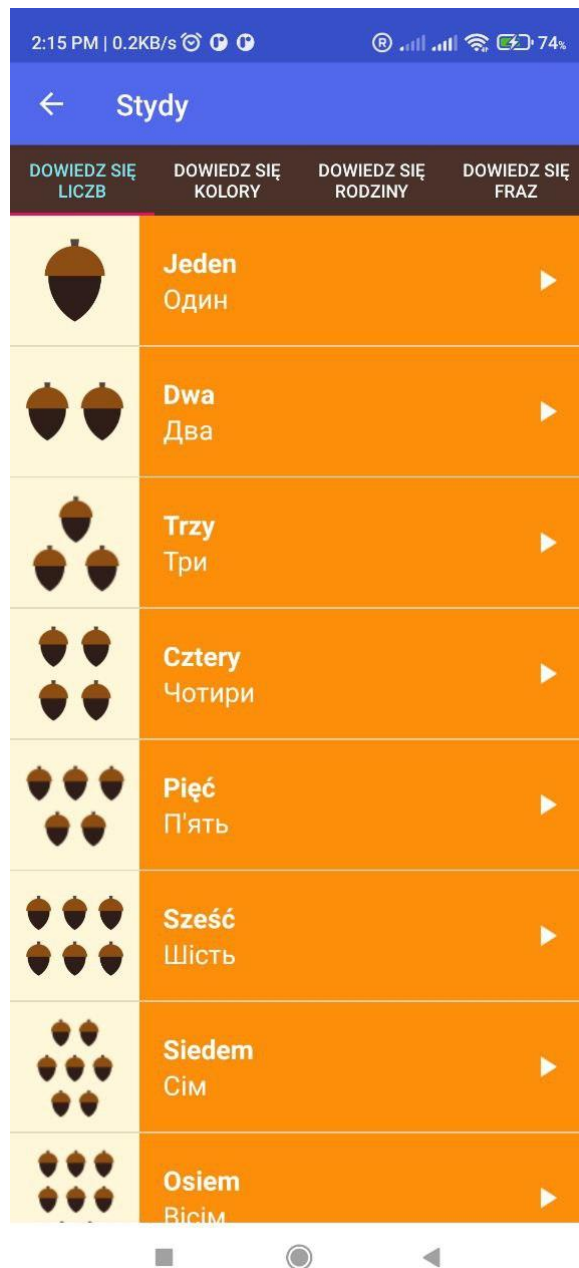


Рисунок 4.6 – Результат виконання тест-кейсу №4

Фактичний результат виконання даного тест-кейсу повністю відповідає очікуваному, отже тест-кейс №4 успішно пройдено.

При проведенні тестування програмного додатку було отримано повну відповідність фактичних результатів очікуваним. Помилки чи багів при роботі програми не виявлено, додаток працював в точності так, як було прогнозовано

відповідно до описаного функціоналу. У результаті доведено повну працездатність програмного додатку та його відповідність поставленому технічному завданню.

4.3 Розробка інструкції користувача

Так як „Pollaung” є додатком для андроїд пристроїв типу телефонів та планшетів, перед тим як його використовувати, потрібно установити його APK, або ж скачати його з мережі playmarket [28], як це показано на рисунку 4.7.

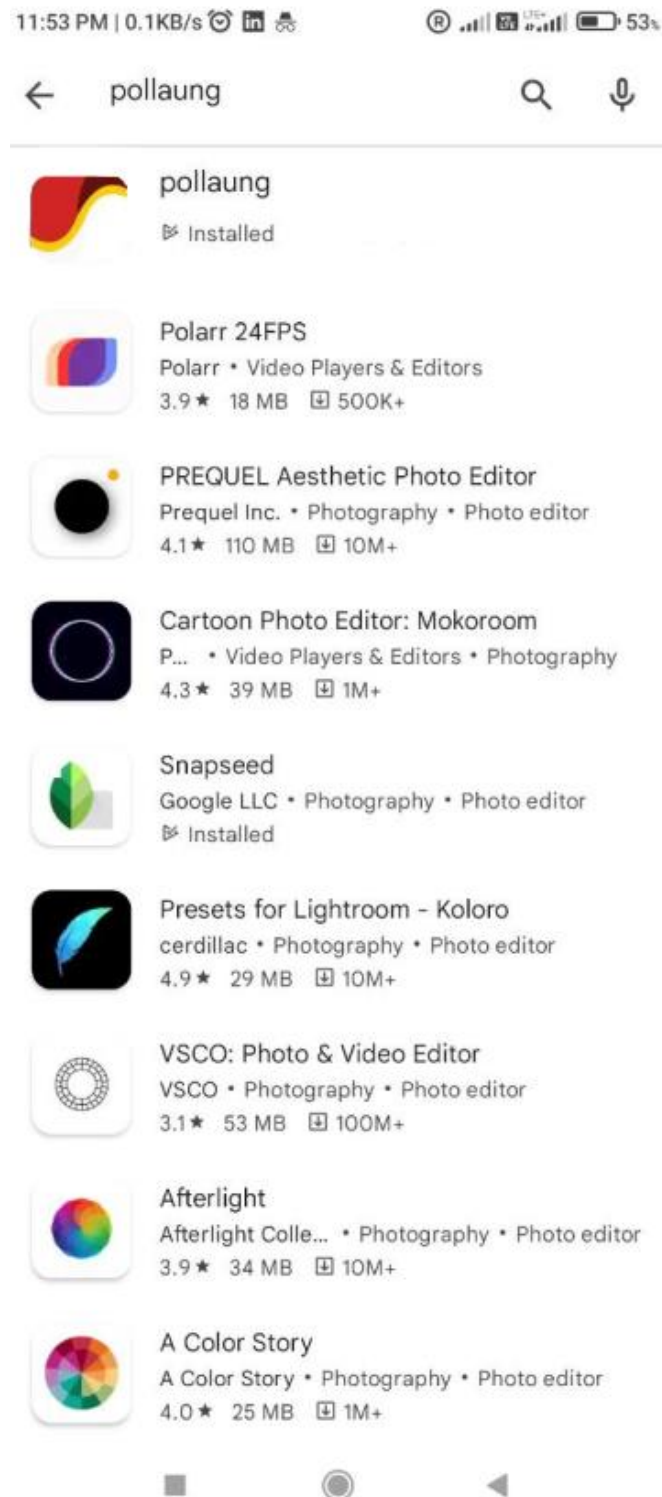


Рисунок 4.7 – Результат пошуку додатку в мережі playmarket

Після цього потрібно надати всі дозволи які попросить додаток при першому запуску, після чого він успішно запустить інтерфейс головного вікна додатку (рисунок 4.8).

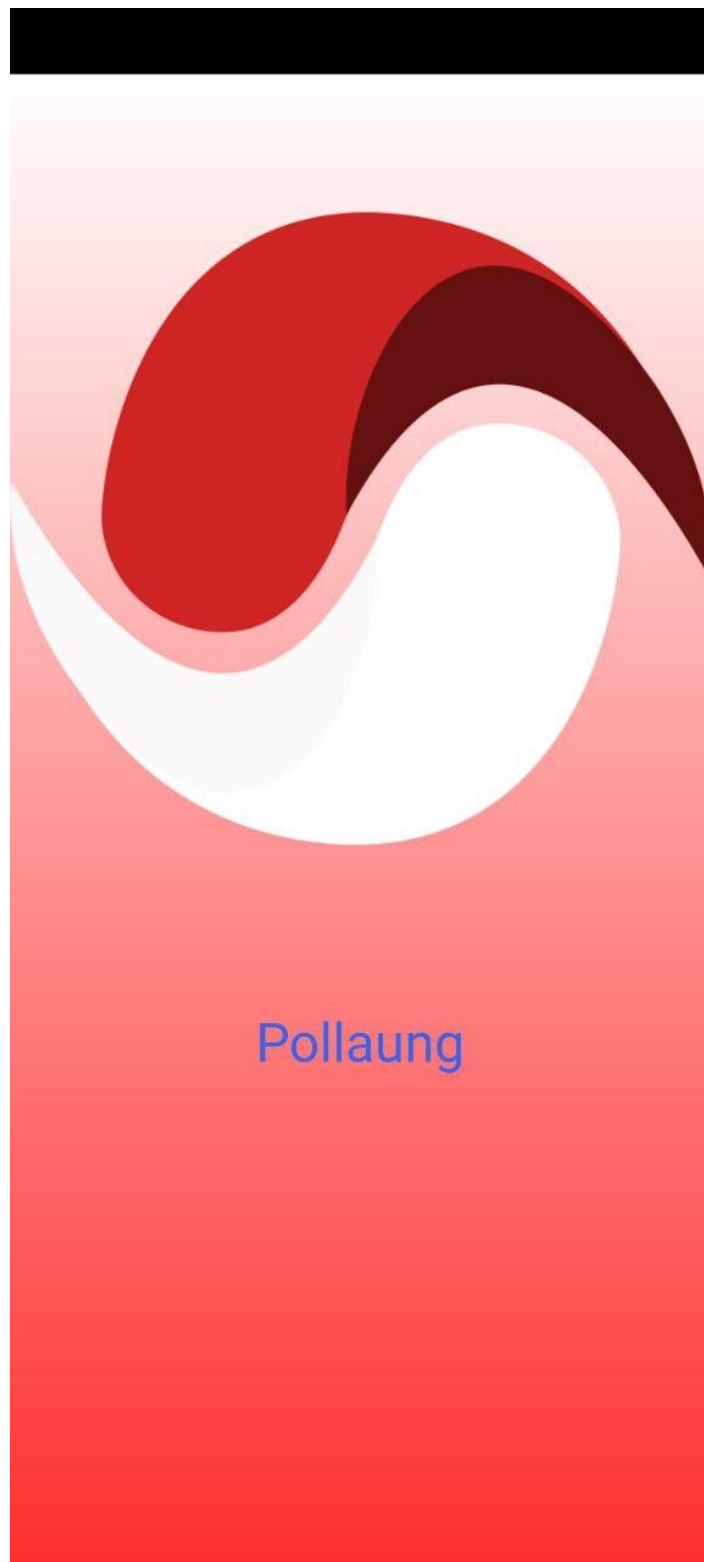


Рисунок 4.8 – Інтерфейс головного вікна додатку

Після запуску боту користувач має увійти у свій аккаунт або ж зареєструвати новий аккаунт, ввівши свою електронну адресу ім'я та пароль.

Приклад зображено на рисунку 4.9.



Рисунок 4.9 – Приклад логування

Користувач може переглядати всі навчальні матеріали в відповідній вкладці, яку можливо обрати з меню. В даній вкладці також всі матеріали поділені на теми, які можна міняти за допомогою гортання.

Приклад виклику даного функціоналу на рисунку 4.10.

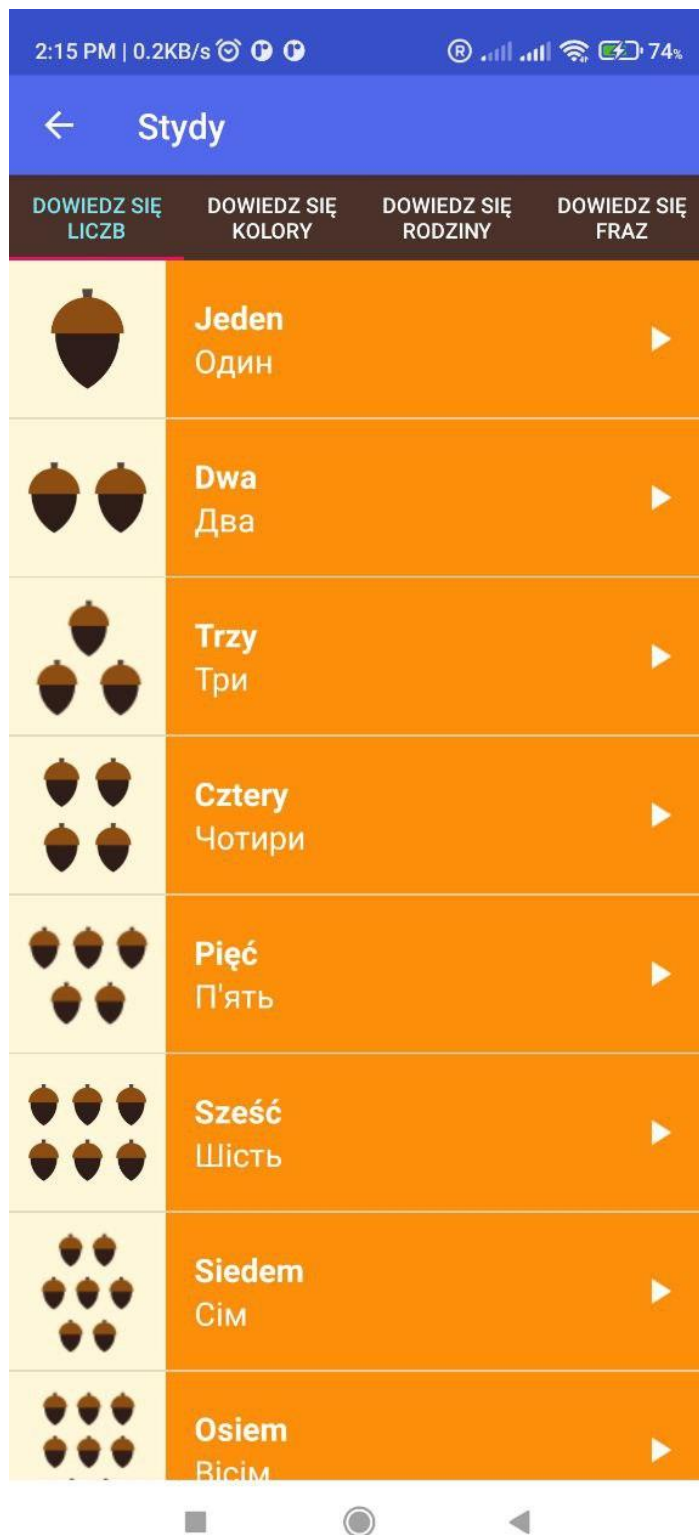


Рисунок 4.10 – Приклад інтерфейсу навчання

Також користувач має змогу проходити тестування по вивченим матеріалам, даний функціонал доступний в меню з відповідною назвою про тестування „Study”.

Приклад виклику даного вікна наведено на рисунку 4.11.

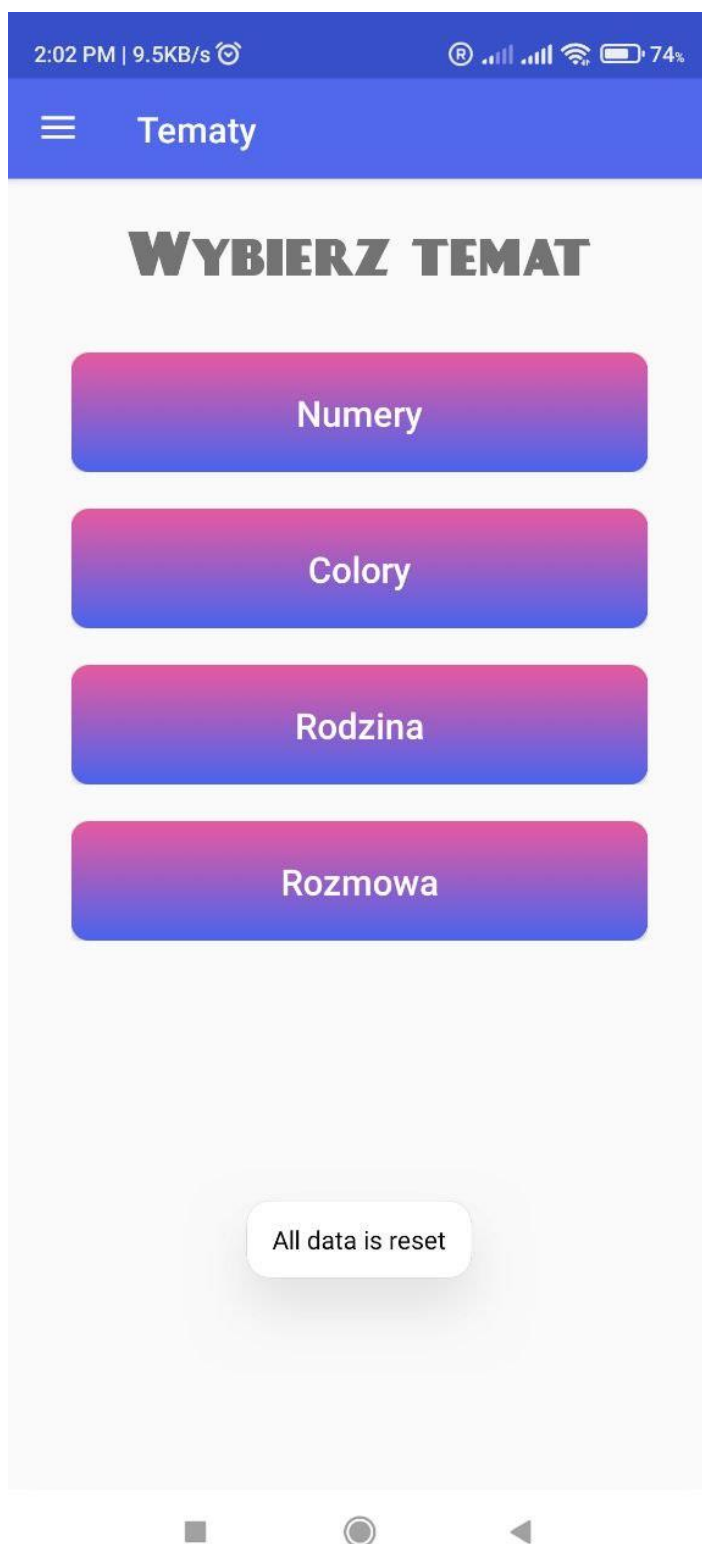


Рисунок 4.11 – Приклад інтерфейсу „Study”

Також в більш досвідченішого користувача є змога створювати та проходити власні тести, які він може додавати до вже існуючих тем. Приклад виклику даного вікна наведено на рисунку 4.12.

2:03 PM | 0.3KB/s

← Dodaj pytania

Tematy :

Numery

Poziomy :

Level 1

Co to?

jest

sous

wiem

my

Poprawna odpowiedź :

jest

sous

wiem

my

Dodaj pytanie

Рисунок 4.12 – Приклад створення тесту

Отже, було розроблено інструкцію користувача. В ній описані дії, які потрібно виконати для початку використання повного функціоналу додатку для вивчення польської мови та можливі варіанти взаємодії користувача із ним.

4.4 Вимоги до пристрою

Так як розроблений додаток працює на базі android, то для його використання користувачу потрібний встановлений на його пристрої даний додаток. Мінімальна та рекомендована конфігурації android пристрою наведено у таблицях 4.1 та 4.2.

Таблиця 4.1 – Мінімальна конфігурація

Тип процесора	процесор з тактовою частотою 1,6 ГГц
Об'єм оперативної пам'яті	4 ГБ
Місце на жорсткому диску	500 mb
Дисплей	5"
Операційна система	Android 5.0 version

Таблиця 4.2 – Рекомендована конфігурація

Тип процесора	процесор з тактовою частотою 2,4 ГГц
Об'єм оперативної пам'яті	6 ГБ
Місце на жорсткому диску	1Gb
Дисплей	5"
Операційна система	Android 6.0 version

4.5 Висновки

В даному розділі було проведено тестування програмного додатку для вивчення польської мови, для тестування обрано методику «чорної скриньки», яка передбачає перевірку функціональності додатку за допомогою тест-кейсів.

Під час тестування додатку було доведено його повну працездатність та відповідність поставленому технічному завданню. Розроблено інструкцію користувача, яка допоможе користувачу краще розуміти як працює додаток та як його потрібно використовувати, щоб досягти найбільшого результату у вивченні мови.

Крім того було визначено конфігурацію, мінімальну та рекомендовану, для пристрою на якому буде запускатись даний додаток, для правильної та ефективної його роботи.

ВИСНОВКИ

У бакалаврській дипломній роботі було розроблено програмний додаток для вивчення і тестування знань з польської мови під назвою „Pollaung”. Розроблений додаток підвищення ефективності вивчення користувачем нової для нього мови спілкування шляхом створення тестів на основі тестування найпопулярніших та найнеобхідніших іноземних слів в даній мові.

Було проведено аналіз існуючих додатків для вивчення польської мови. В результаті аналізу аналогів було висвітлено їхні основні недоліки. Проведено їхнє порівняння із власним програмним продуктом, в результаті чого було визначено доцільність розробки нового програмного додатку для вивчення і тестування знань з польської мови. Досліджено процеси вивчення, тестування та запам'ятовування нової інформації на основі перегляду, тестування та слухання нових для користувача слів за допомогою додатку. Виконано постановку задач розробки програмного продукту.

Було проведено аналіз засобів реалізації навчальних додатків, орієнтованих на використання принципів концепції Edutainment, при розробці власного навчального додатку. Було розроблено метод та модель роботи навчальної програми, мобільний додаток для ефективного вивчення польської мови з можливістю вивчення і тестування за вивченими матеріалами. Також було розроблено аудіо супровід під час вивчення мови, щоб покращити та збільшити швидкість вивчення мови. Розроблено основні модулі програми.

Було обґрунтовано вибір програмних засобів при розробці програми, обрано мову програмування Java, середовище розробки Android Studio та СУБД SQLite. В процесі розробки було описано програмну реалізацію основних модулів програми. Було розглянуто основні види та методи тестування, обрано методику «чорного ящика». Проведено тестування за допомогою ряду тест-кейсів, яке довело повну працездатність програмного додатку та його відповідність поставленому технічному завданню. Розроблено інструкцію користувача та вимоги до пристрою користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Shawn Loewen, Daniel R. Isbell, Zachary Sporn, The effectiveness of app-based language instruction for developing receptive linguistic knowledge and oral communicative ability 1 [Електронний ресурс] Режим доступу до ресурсу: <https://onlinelibrary.wiley.com/doi/10.1111/flan.12454>.
2. Ковтун Б. В. Порівняльна характеристика реляційних та NoSQL баз даних / Б. В. Ковтун, А. М. Манич, О. В. Романюк // Матеріали XLIX науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2020): збірник доповідей. – Вінниця: ВНТУ, 2020. – С. 1177-1179. – [Електронний ресурс] – Режим доступу: https://conferences.vntu.edu.ua/public/files/1/vntu_2020_netpub.pdf.
3. Войтко В.В. Навчальна система для вивчення і тестування з польської мови / В. В. Войтко, Г. Б. Ракитянська, С. В. Бевз, С. М. Бурбело, А. М. Манич // Матеріали Всеукраїнської науково-практичної інтернет-конференції "Молодь в науці: дослідження, проблеми, перспективи - 2022", Секція - Інформаційні технології та комп'ютерна інженерія. [Електронний ресурс] – Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2022/paper/viewFile/16201/13643>
4. Duolingo [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.duolingo.com>
5. Ashok Kumar. The duolingo english test: essential tips, strategies, practice, structure, grammar, spellings & vocabulary, 2020.
6. Speechling [Електронний ресурс] – Режим доступу до ресурсу: <https://speechling.com/>
7. Tandem [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tandem.net/>
8. Eric Brow. That's Edutainment! A Parent's Guide to Educational Software, 1995.
9. Andreas Oikonomou. Serious Games and Edutainment Applications Minhua Ma, 2011.

10. Bryan D. Svencer. Edutainment: Entertainment in the K-12 Classroom, 2012.
11. Jeff Hawkins. A Thousand Brains: A New Theory of Intelligence, 2021.
12. I. E. Hewitt. Edutainment: How to Teach Language with Fun and Games, 1998.
13. Melvia Miller. Edutainment Game Book: A Guide for Instructors to Spice, 2009.
14. John Horton. Android Programming for Beginners: Build in-depth, full-featured Android apps starting from zero programming experience, 2021.
15. Ian Darwin. Android Cookbook: Problems and Solutions for Android Developers, 2011.
16. Neil Smyth. Android Studio 4.2 Development Essentials - Java Edition: Developing Android Apps Using Android Studio 4.2, Java and Android Jetpack, 2021.
17. Rob Whitaker. Developing Inclusive Mobile Apps: Building Accessible Apps for iOS and Android, 2020.
18. Josh Lospinoso. C++ Crash Course: A Fast-Paced Introduction, 2019.
19. Joe Mayo. C# Cookbook: Modern Recipes for Professional Developers, 2021.
20. Andrew Warner. Python for Absolute Beginners: A Step by Step Guide to Learn Python Programming from Scratch, with Practical Coding Examples and Exercises, 2020.
21. Allen Downey. Think Java: How to Think Like A Computer Scientist, 2020.
22. Knott Daniel. Hands-On Mobile App Testing: A Guide for Mobile Testers and Anyone Involved in the Mobile App Business, 2020.
23. Software testing [Электронный ресурс] – Режим доступа до ресурсу: https://en.wikipedia.org/wiki/Software_testing
24. Smilgin Radosław. Zawód tester. Od decyzji do zdobycia doświadczenia, 2018.

25. Cem Kaner. Lessons Learned in Software Testing: A Context-Driven Approach, 2002.

26. Основи тестування [Електронний ресурс] – Режим доступу до ресурсу: <https://qalight.ua/baza-znaniy/shho-take-testuvannya-programnogo-zabezpechennya>.

27. John Smart BDD in Action: Behavior-Driven Development for the Whole Software Lifecycle, 2014.

28. “Pollaung” [Електронний ресурс] – Режим доступу до ресурсу: <https://play.google.com/store/search?q=pollaung&c=apps&gl=UA>

ДОДАТКИ

Додаток А – Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедру ІЗ

д.т.н., проф.

_____ О. Н. Романюк

“_25_” березня 2022 року

Технічне завдання

**на бакалаврську дипломну роботу «Розробка програми для вивчення і
тестування знань з польської мови» за спеціальністю
121 – Інженерія програмного забезпечення**

Керівник бакалаврської дипломної роботи:

_____ к.т.н., доцент Г. Б. Ракитянська

“_25_” березня 2022 року

Виконав:

_____ студент гр. ЗПІ-186 А. М. Манич

“_25_” березня 2022 року

Вінниця – 2022 року

1. Найменування та галузь застосування

Бакалаврська дипломна робота: «Розробка програми для вивчення і тестування знань з польської мови».

Галузь застосування – навчання.

2. Підстава для розробки.

Підставою для виконання бакалаврської дипломної роботи (БДР) є індивідуальне завдання на БДР та наказ №66 від «24» березня 2022 р. ректора по ВНТУ про закріплення тем БДР.

3. Мета та призначення розробки.

Метою дослідження є підвищення ефективності вивчення користувачем польської мови шляхом розробки та використання спеціалізованої мобільної навчальної системи з теоретичним матеріалом і тестами, що дозволить підвищити рівень володіння новою мовою.

Призначення роботи – розробка та програмна реалізація програми для вивчення і тестування знань з польської мови.

4. Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись БДР.

1. Andreas Oikonomou. Serious Games and Edutainment Applications Minhua Ma, 2011.

2. Eric Brow. That's Edutainment! A Parent's Guide to Educational Software, 1995.

3. Rob Whitaker. Developing Inclusive Mobile Apps: Building Accessible Apps for iOS and Android, 2020.

4. Войтко В.В. Навчальна система для вивчення і тестування з польської мови / В. В. Войтко, Г. Б. Ракитянська, С. В. Бевз, С. М. Бурбело, А. М. Манич // Матеріали Всеукраїнської науково-практичної інтернет-конференції "Молодь

в науці: дослідження, проблеми, перспективи - 2022", Секція - Інформаційні технології та комп'ютерна інженерія. [Електронний ресурс] – Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2022/paper/viewFile/16201/13643>.

5. Технічні вимоги

Система управління базами даних – SQLite; мова запитів – SQL; бібліотека для розробки графічного інтерфейсу – Layout Editor; середовище розробки – Android Studio 2022; мова програмування – Java, Kotlin; бібліотека для тестування: Junit

6. Конструктивні вимоги

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

1. Пояснювальна записка до БДР;
2. Технічне завдання;
3. Лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки:

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання і вибір методу вирішення поставленої задачі дослідження	26.03.2022 - 29.03.2022	Вик.
2	Вибір архітектури програмного компоненту	01.04.2022 - 09.04.2022	Вик.
3	Розробка структури графічного інтерфейсу „Pollaung”	10.04.2022 - 19.04.2022	Вик.
4	Вибір середовища та мови розробки	20.04.2022 - 24.04.2022	Вик.
5	Розробка модулів програми	27.04.2022 - 11.05.2022	Вик.
6	Тестування програми	11.05.2022 - 15.05.2022	Вик.
7	Оформлення матеріалів до захисту БДР	16.05.2022 - 28.06.2022	Вик.

10. Порядок контролю та прийняття

Виконання етапів бакалаврської дипломної роботи контролюється керівником згідно з графіком виконання роботи.

Прийняття бакалаврської дипломної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком.

Додаток Б – Перевірка на плагіат

ПРОТОКОЛ ПЕРЕВІРКИ БАКАЛАВРСЬКОЇ ДИПЛОМНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка програми для вивчення і тестування знань з польської мови

Тип роботи: БДР

Підрозділ : кафедра програмного забезпечення, ФІТКІ

Науковий керівник: Г. Б. Ракитянська

Оригінальність	92.2%
Схожість	7.8%

Аналіз звіту подібності

■ **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.**

Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Черноволик Г. О.

Ознайомлені з повним звітом подібності, який був згенерований системою Unichesk

Автор роботи _____

Манич А. М.

Керівник роботи _____

Ракитянська Г. Б.

Додаток В – Лістинг програми

```

package com.crypto.quizapp.Activities;

import ...

public class HomeActivity extends AppCompatActivity {

    private Toolbar toolbar;
    private DrawerLayout drawer;
    private NavigationView navigationView; // Navigation drawer
    private ActionBarDrawerToggle drawerToggle;
    private NavController fragNavController; // NavController library controller
    private NavController.OnBackPressedDispatcherDelegate fragNavControllerDelegate; // NavController delegate

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        initView(savedInstanceState);
    }

    private void initView(Bundle savedInstanceState) {
        drawer = findViewById(R.id.drawer_layout);
        navigationView = findViewById(R.id.nav_view);
        drawerToggle = new ActionBarDrawerToggle(
            activity, drawer, toolbar, "Open navigation drawer", R.string.navigation_drawer_close);
        drawer.addDrawerListener(drawerToggle);
        drawerToggle.syncState();

        // declaring NavController controller
        fragNavController = new NavController(getSupportFragmentManager(), R.id.content_frame);

        // declaring NavController delegate
        fragNavControllerDelegate = new NavController.OnBackPressedDispatcherDelegate() {
            @Override
            public void onBackPressed() {
                // adding root fragments
                List<Fragment> fragments = new ArrayList<>();
                fragments.add(TopicSelectFrag.newInstance());

                // setting root fragments to NavController controller
                fragNavController.setRootFragments(fragments);

                // setting NavController delegate to NavController controller
                fragNavController.setOnBackPressedDispatcherDelegate(fragNavControllerDelegate);

                // setting first fragment to display when we get on HomeActivity
                fragNavController.initialize(NavigationView.Tab1, savedInstanceState);

                // navigation drawer listener to navigate through activity and fragment from drawer
                navigationView.setNavigationItemSelectedListener((menuItem) -> {
                    switch (menuItem.getItemId()) {
                        case R.id.nav_home:
                            // checking if the fragment is root fragment or not, if not then pop the fragment from stack,
                            // if it is root fragment then switch the fragment to First fragment in stack

                            if(!fragNavController.isRootFragment()){
                                fragNavController.popFragment();
                            }else {
                                fragNavController.switchTab(NavigationView.Tab1);
                            }
                            break;

                        case R.id.nav_aboutus:
                            // push fragment in the stack
                            fragNavController.pushFragment(AboutUsFrag.newInstance());
                            break;

                        case R.id.nav_study:
                            Intent z1 = new Intent( packageContext, HomeActivity.this, StudyActivity.class);
                            startActivity(z1);
                            break;

                        case R.id.nav_reset:
                            // clearing user answers
                            deleteQuestions();
                            break;
                    }
                });
            }
        };
    }
}

```

Рисунок Б.1 – Лістинг класу HomeActivity

```

package com.crypto.quizapp.Activities;

import ...

public class AddQuestionActivity extends AppCompatActivity implements View.OnClickListener {

    private Spinner topic_spinner;
    private Spinner level_spinner;
    private EditText question_edt;
    private EditText option_one_edt;
    private EditText option_two_edt;
    private EditText option_three_edt;
    private EditText option_four_edt;
    private RadioButton option_one_rb;
    private RadioButton option_two_rb;
    private RadioButton option_three_rb;
    private RadioButton option_four_rb;
    private RadioGroup correct_grp;
    private Button add_question_btn;

    private List<String> topics_list;
    private List<String> levels_list;
    private ArrayAdapter<String> topics_adapter;
    private ArrayAdapter<String> levels_adapter;
    private String topic_st, level_st, correct_st, question_no;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_question);
        setTitle("Dodaj pytanie");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);
        initView();
    }

    private void initView() {
        topic_spinner = (Spinner) findViewById(R.id.topic_spinner);
        level_spinner = (Spinner) findViewById(R.id.level_spinner);
        question_edt = (EditText) findViewById(R.id.question_edt);
        option_one_edt = (EditText) findViewById(R.id.option_one_edt);
        option_two_edt = (EditText) findViewById(R.id.option_two_edt);
        option_three_edt = (EditText) findViewById(R.id.option_three_edt);
        option_four_edt = (EditText) findViewById(R.id.option_four_edt);
        option_one_rb = (RadioButton) findViewById(R.id.option_one_rb);
        option_two_rb = (RadioButton) findViewById(R.id.option_two_rb);
        option_three_rb = (RadioButton) findViewById(R.id.option_three_rb);
        option_four_rb = (RadioButton) findViewById(R.id.option_four_rb);
        correct_grp = (RadioGroup) findViewById(R.id.correct_grp);
        add_question_btn = (Button) findViewById(R.id.add_question_btn);
        add_question_btn.setOnClickListener(this);
        option_one_rb.setChecked(true);
        option_one_edt.addTextChangedListener(new MyTextWatcher(option_one_edt));
        option_two_edt.addTextChangedListener(new MyTextWatcher(option_two_edt));
        option_three_edt.addTextChangedListener(new MyTextWatcher(option_three_edt));
        option_four_edt.addTextChangedListener(new MyTextWatcher(option_four_edt));

        getTopics();
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.add_question_btn:
                submit();
                break;
        }
    }

    private void submit() {
        // validations

        if (TextUtils.isEmpty(question_edt.getText().toString().trim())) {
            question_edt.setError("Please enter a question");
            requestFocus(question_edt);
        } else if (TextUtils.isEmpty(option_one_edt.getText().toString().trim())) {
            option_one_edt.setError("Please enter value of option 1");
        }
    }
}

```

Рисунок Б.2 – Лістнінг класу AddQuestionActivity

```

package com.crypto.quizapp.Activities;

import ...

public class LevelSelectActivity extends AppCompatActivity implements View.OnClickListener {

    private LinearLayout linearLayout;
    private String language;
    private Button score_board;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_levelselect_activity);
        setTitle("Poziomy"); // setting the title on toolbar
        getSupportActionBar().setDisplayHomeAsUpEnabled(true); // enabling the back button on toolbar
        getSupportActionBar().setDisplayHomeAsUpEnabled(true); // enabling the back button on toolbar

        language = getIntent().getStringExtra("lang"); // getting the selected topic value from previous fragment
        initView();
    }

    private void initView() {
        linearLayout = findViewById(R.id.lvl_linear);
        score_board = findViewById(R.id.btn_scoreboard);
        score_board.setOnClickListener(this);
        getQuestions();
    }

    private void getQuestions() {

        // getting levels

        class GetQuestions extends AsyncTask<Void, Void, List<QuestionsBean>> {

            @Override
            protected List<QuestionsBean> doInBackground(Void... voids) {
                List<QuestionsBean> questionsBeanList = DatabaseClient
                    .getInstance(getApplicationContext()) DatabaseClient
                    .getAppDatabase() AppDatabase
                    .questionDao() QuestionDao
                    .getLevels(language);
                return questionsBeanList;
            }

            @Override
            protected void onPostExecute(List<QuestionsBean> questionsBeanList) {
                super.onPostExecute(questionsBeanList);
                setdata(questionsBeanList);
            }
        }

        GetQuestions getQuestions = new GetQuestions();
        getQuestions.execute();
    }

    private void setdata(List<QuestionsBean> questionsBeanList) {

        // dynamic layouts
        // clear linear layout before adding view to the root layout

        linearLayout.removeAllViews();
        int lvl = 0;
        for (int i = 0; i < questionsBeanList.size(); i++) {
            if (questionsBeanList.get(i).getLevel() > lvl) {

                lvl = questionsBeanList.get(i).getLevel();

                // layout inflater to add views
                LayoutInflater inflater = (LayoutInflater) getSystemService(
                    Context.LAYOUT_INFLATER_SERVICE);

                // layout parameter needed for views
                LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(
                    LinearLayout.LayoutParams.MATCH_PARENT, LinearLayout.LayoutParams.WRAP_CONTENT);
                params.setMargins(left: 20, top: 20, right: 50, bottom: 30);
            }
        }
    }
}

```

Рисунок Б.3 – Лістнінг класу LevelSelectActivity

```

package com.crypto.quizapp.Activities;

import ...

public class QuizActivity extends AppCompatActivity implements View.OnClickListener {

    private RecyclerView quiz_Recyclerview;
    private String languagevalue;
    private int levelvalue;
    private Button btn_submit;
    private List<QuestionsBean> questionslist;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_quiz);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);

        levelvalue = getIntent().getIntExtra("lvl", 0);
        languagevalue = getIntent().getStringExtra("lang");
        setTitle("Poziom "+levelvalue); // setting title on toolbar
        initView();
    }

    private void initView() {
        quiz_Recyclerview = findViewById(R.id.quiz_recyclerview);
        quiz_Recyclerview.setHasFixedSize(true);
        btn_submit = findViewById(R.id.btn_submit);
        btn_submit.setOnClickListener(this);
        getQuestions();
    }

    private void getQuestions() {
        // getting all the questions with selected topic and level
        class GetQuestions extends AsyncTask<Void, Void, List<QuestionsBean>> {

            @Override
            protected List<QuestionsBean> doInBackground(Void... voids) {
                List<QuestionsBean> taskList = DatabaseClient
                    .getInstance(getApplicationContext()) DatabaseClient
                    .getAppDatabase() AppDatabase
                    .questionDao() QuestionDao
                    .getSelectedQuestions(languagevalue, levelvalue);
                return taskList;
            }

            @Override
            protected void onPostExecute(List<QuestionsBean> questionsBeanList) {
                super.onPostExecute(questionsBeanList);

                // defining the linearlayoutmanager
                LinearLayoutManager layoutManager = new LinearLayoutManager(getApplicationContext());

                // setting linearlayoutmanager to the recyclerview
                quiz_Recyclerview.setLayoutManager(layoutManager);

                // custom adapter for recyclerview
                QuestionsAdapter qadapter =
                    new QuestionsAdapter(applicationContext: QuizActivity.this, questionsBeanList);

                // setting the adapter to the recyclerview
                quiz_Recyclerview.setAdapter(qadapter);
            }
        }

        GetQuestions getQuestions = new GetQuestions();
        getQuestions.execute();
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.btn_submit:
                // moving to the result activity

```

Рисунок Б.4 – Лістнінг класу QuizActivity

```

package com.crypto.quizapp.Activities;

import ..

public class ResultActivity extends AppCompatActivity implements View.OnClickListener {
    private TextView language_txt;
    private TextView lvl_txt;
    private TextView correct_txt;
    private TextView wrong_txt;
    private TextView not_ans_txt;
    private Button btn_finish, btn_nextlvl;
    private String language;
    private int level;
    private int correct_count;
    private int wrong_count;
    private int not_ans_count;
    private int total_count;
    private int passing_marks;
    private List<Integer> levels_list;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_result);
        setTitle("Wynik"); // setting the toolbar title
        getSupportActionBar().setDisplayHomeAsUpEnabled(true); // enabling the toolbar back button
        getSupportActionBar().setDisplayShowHomeEnabled(true); // enabling the toolbar back button

        language = getIntent().getStringExtra( name: "Lang");
        level = getIntent().getIntExtra( name: "Lvl", defaultValue: 0);

        initView();
    }

    private void initView() {
        language_txt = findViewById(R.id.lang_txt);
        lvl_txt = findViewById(R.id.lvl_count);
        correct_txt = findViewById(R.id.correct_count);
        wrong_txt = findViewById(R.id.wrong_count);
        not_ans_txt = findViewById(R.id.not_answered_count);
        btn_finish = findViewById(R.id.finish_btn);
        btn_finish.setOnClickListener(this);
        btn_nextlvl = findViewById(R.id.next_lvl_btn);
        btn_nextlvl.setOnClickListener(this);
        getQuestions();
    }

    private void getQuestions() {
        // getting the questions

        class GetQuestions extends AsyncTask<Void, Void, List<QuestionsBean>> {

            @Override
            protected List<QuestionsBean> doInBackground(Void... voids) {
                List<QuestionsBean> questionsBeanList = DatabaseClient
                    .getInstance(getApplicationContext()) DatabaseClient
                    .getAppDatabase() AppDatabase
                    .questionDao() QuestionDao
                    .getSelectedQuestions(language, level);
                return questionsBeanList;
            }

            @Override
            protected void onPostExecute(List<QuestionsBean> questionsBeans) {
                super.onPostExecute(questionsBeans);
                getResult(questionsBeans);
            }
        }

        GetQuestions getQuestions = new GetQuestions();
        getQuestions.execute();
    }

    private void updateQuestions(final List<QuestionsBean> questionsBeanList) {
        // updating the score and setting the next level to enable

        class updateQuestions extends AsyncTask<Void, Void, Void> {

```

Рисунок Б.5 – Лістнінг класу ResultActivity

```

package com.crypto.quizapp.Activities;

import ...

public class ScoreboardActivity extends AppCompatActivity {

    private LinearLayout linearLayout;
    private String language;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scoreboard);
        setTitle("Tablica wyników"); // setting the title on toolbar
        getSupportActionBar().setDisplayHomeAsUpEnabled(true); // enabling the back button on toolbar
        getSupportActionBar().setDisplayHomeAsUpEnabled(true); // enabling the back button on toolbar

        language = getIntent().getStringExtra("lang"); // getting the selected topic value from previous fr
        initView();
    }

    private void initView() {
        linearLayout = findViewById(R.id.lv1_Score);
        getQuestions();
    }

    private void getQuestions() {

        // getting levels and their scores

        class GetQuestions extends AsyncTask<Void, Void, List<QuestionsBean>> {

            @Override
            protected List<QuestionsBean> doInBackground(Void... voids) {
                List<QuestionsBean> questionsBeanList = DatabaseClient
                    .getInstance(getApplicationContext()) DatabaseClient
                    .getAppDatabase() AppDatabase
                    .questionDao() QuestionDao
                    .getLevels(language);
                return questionsBeanList;
            }

            @Override
            protected void onPostExecute(List<QuestionsBean> questionsBeanList) {
                super.onPostExecute(questionsBeanList);
                setdata(questionsBeanList);
            }
        }

        GetQuestions getQuestions = new GetQuestions();
        getQuestions.execute();
    }

    private void setdata(List<QuestionsBean> tasks) {

        // dynamic layouts
        // clear all view before adding view to the root layout

        linearLayout.removeAllViews();
        int lv1 = 0;
        for (int i = 0; i < tasks.size(); i++) {
            if (tasks.get(i).getLevel() > lv1) {
                lv1 = tasks.get(i).getLevel();

                // layout inflater
                LayoutInflater inflater = (LayoutInflater) getSystemService(
                    Context.LAYOUT_INFLATER_SERVICE);

                // layout parameters
                LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(
                    LinearLayout.LayoutParams.MATCH_PARENT, LinearLayout.LayoutParams.WRAP_CONTENT);
                params.setMargins(left: 20, top: 20, right: 50, bottom: 30);

                // parent layout
                final View view = inflater.inflate(R.layout.score_item_lay, root: null);
                final LinearLayout layout1 = (LinearLayout) view.findViewById(R.id.lv1_linear_lay);
                layout1.setLayoutParams(params);

                TextView textView = view.findViewById(R.id.lv1_score_txt);
                String score = getLevelScore(tasks.get(i).getLevel()); // getting score according to each level
            }
        }
    }
}

```

Рисунок Б.6 – Лістніг класу ScoreBarActivity

```

package com.crypto.quizapp.Activities;

import ...

public class SignInActivity extends AppCompatActivity implements View.OnClickListener {

    private Button btn_signin;
    private LinearLayout btn_signup;
    private EditText email_edt;
    private EditText pass_edt;

    @RequiresApi(api = Build.VERSION_CODES.M)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signin);
        getSupportActionBar().hide(); // hide toolbar
        getWindow().setStatusBarColor(this.getColor(R.color.Lightpink));
        initView();
    }

    private void initView() {
        email_edt = findViewById(R.id.email_edt);
        pass_edt = findViewById(R.id.pass_edt);
        btn_signin = findViewById(R.id.btn_signin);
        btn_signin.setOnClickListener(this);
        btn_signup = findViewById(R.id.signup_btn_signin);
        btn_signup.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.signup_btn_signin:
                // moving from SignInActivity to SignupActivity

                Intent i = new Intent( packageContext SignInActivity.this, SignupActivity.class);
                startActivity(i);
                break;
            case R.id.btn_signin:
                submit();
                break;
        }
    }

    private void submit() {

        // validations

        if (!validateEmail()) {
            return;
        } else if (!validatepassword()) {
            return;
        } else {
            signInUser();
        }
    }

    private void signInUser() {

        final String email = email_edt.getText().toString().trim();
        final String pass = pass_edt.getText().toString().trim();

        // Checking if the user exist in the database

        class SignInUser extends AsyncTask<Void, Void, Boolean> {

            @Override
            protected Boolean doInBackground(Void... voids) {

                boolean usr_exist = DatabaseClient.getInstance(SignInActivity.this).getAppDatabase()
                    .usersDao()
                    .checkUser(email, pass);

                return usr_exist;
            }
        }
    }
}

```

Рисунок Б.7 – Лістнінг класу SignInActivity


```

package com.crypto.quizapp.Beans;

import ...

@Entity
public class QuestionsBean implements Serializable {

    @PrimaryKey(autoGenerate = true)
    private int id;

    @ColumnInfo(name = "topic")
    private String topic;

    @ColumnInfo(name = "level")
    private int level;

    @ColumnInfo(name = "cleared")
    private String cleared;

    @ColumnInfo(name = "questionno")
    private String questionno;

    @ColumnInfo(name = "question")
    private String question;

    @ColumnInfo(name = "option1")
    private String option1;

    @ColumnInfo(name = "option2")
    private String option2;

    @ColumnInfo(name = "option3")
    private String option3;

    @ColumnInfo(name = "option4")
    private String option4;

    @ColumnInfo(name = "correctoption")
    private String correctoption;

    @ColumnInfo(name = "useranswer")
    private String useranswer;

    @ColumnInfo(name = "SelectedRadioButton")
    private int SelectedRadioButton;

    public QuestionsBean(String topic, int level, String cleared, String questionno, String question, String option1, Str
        this.topic = topic;
        this.level = level;
        this.cleared = cleared;
        this.questionno = questionno;
        this.question = question;
        this.option1 = option1;
        this.option2 = option2;
        this.option3 = option3;
        this.option4 = option4;
        this.correctoption = correctoption;
        this.useranswer = useranswer;
    }

    public QuestionsBean() {

    }

    public String getCleared() { return cleared; }

    public void setCleared(String cleared) { this.cleared = cleared; }

    public int getLevel() { return level; }

    public void setLevel(int level) { this.level = level; }

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getTopic() { return topic; }

    public void setTopic(String topic) { this.topic = topic; }
}

```

Рисунок Б.8 – Лістнінг класу QuestionBean

```
package com.crypto.quizapp.Beans;

import androidx.room.ColumnInfo;
import androidx.room.Entity;
import androidx.room.PrimaryKey;

import java.io.Serializable;

@Entity
public class UsersBean implements Serializable {

    @PrimaryKey(autoGenerate = true)
    private int id;

    @ColumnInfo(name = "name")
    private String name;

    @ColumnInfo(name = "email")
    private String email;

    @ColumnInfo(name = "password")
    private String password;

    public UsersBean() {
    }

    public UsersBean(String name, String email, String password) {
        this.name = name;
        this.email = email;
        this.password = password;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```

Рисунок Б.9 – Лістнінг класу UserBean

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context=".Activities.AddQuestionActivity">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginHorizontal="28dp"
                android:layout_marginTop="18dp"
                android:text="Tematy :>
                android:textColor="@color/black"
                android:textSize="16sp" />

            <Spinner
                android:id="@+id/topic_spinner"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginHorizontal="28dp"
                android:layout_marginTop="28dp"
                android:background="@drawable/spinner_bg" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginHorizontal="28dp"
                android:layout_marginTop="18dp"
                android:text="Poziomy :>
                android:textColor="@color/black"
                android:textSize="16sp" />

            <Spinner
                android:id="@+id/level_spinner"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginHorizontal="28dp"
                android:layout_marginTop="28dp"
                android:background="@drawable/spinner_bg" />

            <EditText
                android:id="@+id/question_edt"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginHorizontal="28dp"
                android:layout_marginTop="18dp"
                android:hint="Pytanie" />

            <EditText
                android:id="@+id/option_one_edt"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginHorizontal="28dp"
                android:layout_marginTop="18dp"
                android:hint="Odpowiedz 1" />

            <EditText
                android:id="@+id/option_two_edt"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginHorizontal="28dp"
                android:layout_marginTop="18dp"
                android:hint="Odpowiedz 2" />
        </LinearLayout>
    </ScrollView>
</LinearLayout>

```

Рисунок Б.10 – Лістнінг класу addQuestion.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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="#5fb8c9"
    android:orientation="vertical"
    tools:context=".Activities.ResultActivity">

    <com.crypto.quizapp.helper.CustomTextView
        android:id="@+id/lang_txt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
        android:elevation="4dp"
        android:text="Pollaung"
        android:textColor="@color/red"
        android:textSize="30sp"
        android:textStyle="bold" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginHorizontal="30dp"
        android:layout_marginTop="-15dp"
        android:background="@drawable/bg_white_round_small_corner"
        android:gravity="center"
        android:orientation="vertical"
        android:paddingBottom="70dp">

        <com.crypto.quizapp.helper.CustomTextView
            android:id="@+id/lvl_count"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="50dp"
            android:text="Poziom"
            android:textColor="@color/yellow"
            android:textSize="25sp" />

        <com.crypto.quizapp.helper.CustomTextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:text="Twoj wynik"
            android:textSize="25sp" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginHorizontal="50dp"
            android:layout_marginTop="20dp"
            android:orientation="horizontal">

            <com.crypto.quizapp.helper.CustomTextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Prawidłowych :"
                android:textSize="18sp" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:gravity="end">

                <com.crypto.quizapp.helper.CustomTextView
                    android:id="@+id/correct_count"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="10"
                    android:textSize="18sp" />

            </LinearLayout>
        </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"

```

Рисунок Б.11 – Лістнінг класу resultactivity.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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="@drawable/gradient"
    android:orientation="vertical"
    tools:context=".Activities.SignUpActivity">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
        android:src="@drawable/Logo_small" />

    <EditText
        android:id="@+id/email_edt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginHorizontal="30dp"
        android:layout_marginTop="30dp"
        android:background="@drawable/edit_text_bakground"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:maxLines="1"
        android:paddingHorizontal="10dp"
        android:paddingVertical="10dp" />

    <EditText
        android:id="@+id/pass_edt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginHorizontal="30dp"
        android:layout_marginTop="20dp"
        android:background="@drawable/edit_text_bakground"
        android:hint="Hasło"
        android:inputType="textPassword"
        android:paddingHorizontal="10dp"
        android:paddingVertical="10dp" />

    <EditText
        android:id="@+id/name_edt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginHorizontal="30dp"
        android:layout_marginTop="20dp"
        android:background="@drawable/edit_text_bakground"
        android:hint="Imię"
        android:inputType="textPersonName"
        android:paddingHorizontal="10dp"
        android:paddingVertical="10dp" />

    <Button
        android:id="@+id/btn_signup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="end"
        android:layout_marginHorizontal="20dp"
        android:layout_marginTop="30dp"
        android:background="@drawable/button_bakground"
        android:paddingHorizontal="30dp"
        android:text="Zarejestruj się"
        android:elevation="10dp"
        android:textAllCaps="false"
        android:textSize="16sp" />

    <LinearLayout
        android:id="@+id/signin_btn_signup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginHorizontal="20dp"
        android:layout_marginTop="20dp"
        android:orientation="horizontal"
        android:gravity="center">

```

Рисунок Б.12 – Лістнінг класу signUpActivity.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/forgot_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:background="@drawable/dialog_bg"
        android:orientation="vertical"
        android:padding="10dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="30dp"
            android:orientation="vertical">

            <EditText
                android:id="@+id/topic_edt"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Тема" />

            <Button
                android:id="@+id/add_question_btn"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Додати"
                android:background="@drawable/bg_white_round_small_corner"
                android:backgroundTint="@color/green"
                android:textAllCaps="false"
                android:layout_gravity="end"
                android:textSize="16sp"
                android:textColor="@color/white"
                android:paddingHorizontal="30dp"
                android:layout_marginTop="10dp"
                android:layout_marginHorizontal="20dp" />

        </LinearLayout>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:gravity="right">

        <ImageView
            android:id="@+id/close"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginRight="10dp"
            android:background="@drawable/cross_icon_small" />

    </LinearLayout>
</RelativeLayout>

```

Рисунок Б.13 – Лістнінг класу forgotName.xml

```

<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (C) 2016 The Android Open Source Project
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<!-- Layout for a single list item -->
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="88dp"
android:background="@color/tan_background"
android:minHeight="88dp">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="88dp"
        android:layout_height="88dp" />
    <RelativeLayout
        android:id="@+id/text_container"
        android:layout_width="match_parent"
        android:layout_height="88dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_toRightOf="@id/imageView"
        android:orientation="vertical"
        android:paddingLeft="16dp">
        <TextView
            android:id="@+id/text_view_1"
            android:layout_width="match_parent"
            android:layout_height="44dp"
            android:layout_weight="1"
            android:gravity="bottom"
            android:textAppearance="?android:textAppearanceMedium"
            android:textColor="@android:color/white"
            android:textStyle="bold"
            tools:text="Lutti"
            android:background="?android:attr/selectableItemBackground"/>
        <TextView
            android:id="@+id/text_view_2"
            android:layout_width="match_parent"
            android:layout_height="44dp"
            android:layout_below="@id/text_view_1"
            android:layout_weight="1"
            android:gravity="top"
            android:textAppearance="?android:textAppearanceMedium"
            android:textColor="@android:color/white"
            tools:text="one"
            android:background="?android:attr/selectableItemBackground"/>
        <ImageView
            android:layout_width="24dp"
            android:layout_height="24dp"
            android:layout_alignParentRight="true"
            android:layout_centerVertical="true"
            android:layout_marginRight="16dp"
            android:src="@drawable/ic_play" />
    </RelativeLayout>
</RelativeLayout>

```

Рисунок Б.14 – Лістнінг класу login.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/xtview_ques_no"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0.1"
        android:textSize="20sp"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="10dp"/>

    <TextView
        android:id="@+id/xtview_ques"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="55dp"
        android:text="What is java?"
        android:textSize="20sp"/>

    <RadioGroup
        android:id="@+id/answer_radiogroup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/xtview_ques"
        android:layout_marginLeft="50dp"
        android:layout_marginTop="10dp">

        <RadioButton
            android:id="@+id/option1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="option1"
            android:background="@drawable/edit_text_bakground"/>

        <RadioButton
            android:id="@+id/option2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="option2"
            android:background="@drawable/edit_text_bakground"
            android:layout_marginTop="5dp"/>

        <RadioButton
            android:id="@+id/option3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="option3"
            android:background="@drawable/edit_text_bakground"
            android:layout_marginTop="5dp"/>

        <RadioButton
            android:id="@+id/option4"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="option4"
            android:background="@drawable/edit_text_bakground"
            android:layout_marginTop="5dp"/>

    </RadioGroup>

</RelativeLayout>

```

Рисунок Б.15 – Лістнінг класу signIn.xml


```
<?xml version="1.0" encoding="utf-8"?>

<resources>

    <color name="colorPrimary">#5168EC</color>

    <color name="colorPrimaryDark">#384FCA</color>
    <color name="colorAccent">#D81B60</color>

    <color name="yellow">#FFD803</color>

    <color name="black">#030303</color>

    <color name="white">#FFFFFF</color>
    <color name="transparentwhite">#8FFFFFFF</color>

    <color name="lightpink">#9E2261</color>

    <color name="red">#F02020</color>

    <color name="blue">#3764F1</color>

    <color name="green">#42C989</color>

    <color name="light_green">#9942C989</color>

    <color name="purple">#97898AC4</color>

    <color name="light_White">#D6D3D3</color>

    <color name="light_grey">#5c888888</color>

    <color name="primary_color">#4A312A</color>

    <!-- Primary dark color of the app (shown in the status bar) -->
    <color name="primary_dark_color">#2F1D1A</color>

    <!-- Background color for app -->
    <color name="tan_background">#FFF7DA</color>

    <!-- Background color for the numbers category -->
    <color name="category_numbers">#FD8E09</color>

    <!-- Background color for the family members category -->
    <color name="category_family">#379237</color>

    <!-- Background color for the colors category -->
    <color name="category_colors">#8888A0</color>

    <!-- Background color for the phrases category -->
    <color name="category_phrases">#16AFCA</color>

</resources>
```

Рисунок Б.16 – Лістнінг класу colors.xml

```
package com.crypto.quizapp.helper;

import android.annotation.SuppressLint;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Typeface;
import android.util.AttributeSet;
import androidx.appcompat.widget.AppCompatEditText;

@SuppressLint("AppCompatCustomView")
public class CustomEditText extends AppCompatEditText {

    // custom EditText with custom font

    public CustomEditText(Context context) {
        super(context);

        Typeface face=Typeface.createFromAsset(context.getAssets(), path: "DAYPBL.ttf");

        this.setTypeface(face);
    }

    public CustomEditText(Context context, AttributeSet attrs) {
        super(context, attrs);

        Typeface face=Typeface.createFromAsset(context.getAssets(), path: "DAYPBL.ttf");

        this.setTypeface(face);
    }

    public CustomEditText(Context context, AttributeSet attrs, int defStyle) {
        super(context, attrs, defStyle);

        Typeface face=Typeface.createFromAsset(context.getAssets(), path: "DAYPBL.ttf");

        this.setTypeface(face);
    }

    protected void onDraw (Canvas canvas) {

        super.onDraw(canvas);
    }
}
```

Рисунок Б.17 – Лістнінг класу CustomEditText

```

package com.crypto.quizapp.helper;

import android.content.Context;

import android.content.SharedPreferences;

import android.preference.PreferenceManager;

import com.google.gson.Gson;

import com.google.gson.reflect.TypeToken;

import java.io.File;
import java.lang.reflect.Type;

public class CommonSharedPreference {

    static SharedPreferences.Editor editor;

    // setting a value in shared preferences
    public static void setsharedText(Context context, String key, String value) {

        SharedPreferences preferences = context.getSharedPreferences(context.getPackageName(), context.MODE_PRIVATE);
        SharedPreferences.Editor prefsEditor = preferences.edit();
        prefsEditor.putString(key, value);
        prefsEditor.apply();

    }

    // getting a value from shared preferences
    public static String getsharedText(Context context, String key) {

        SharedPreferences preferences = context.getSharedPreferences(context.getPackageName(), context.MODE_PRIVATE);
        String value = preferences.getString(key, "");
        return value;

    }

    // deleting all the shared preference
    public static void deletesharedpref(Context context) {

        File sharedPreferenceFile = new File( pathname: "/data/data/" + context.getPackageName() + "/shared_prefs/");
        File[] listFiles = sharedPreferenceFile.listFiles();
        for (File file : listFiles) {
            file.delete();
        }
        SharedPreferences sharedPref = PreferenceManager.getDefaultSharedPreferences(context);
        editor = sharedPref.edit().clear();
        editor.commit();

        // if not wanting the fresh start
        SharedPreferences preferences = PreferenceManager.getDefaultSharedPreferences(context);
        SharedPreferences.Editor prefsEditor = preferences.edit();
        prefsEditor.putString( "firstTime", "yes");
        prefsEditor.commit();

    }

}

```

Рисунок Б.18 – Лістинг класу CommonSharedPerformance

```
package com.crypto.quizapp.helper;

import android.annotation.SuppressLint;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Typeface;
import android.util.AttributeSet;

import androidx.appcompat.widget.AppCompatTextView;

@SuppressLint("AppCompatCustomView")
public class CustomTextView extends AppCompatTextView {

    // custom TextView with custom font

    public CustomTextView(Context context) {
        super(context);

        Typeface face=Typeface.createFromAsset(context.getAssets(), "path: \"DAYPBL.ttf\"");

        this.setTypeface(face);
    }

    public CustomTextView(Context context, AttributeSet attrs) {
        super(context, attrs);

        Typeface face=Typeface.createFromAsset(context.getAssets(), "path: \"DAYPBL.ttf\"");

        this.setTypeface(face);
    }

    public CustomTextView(Context context, AttributeSet attrs, int defStyle) {
        super(context, attrs, defStyle);

        Typeface face=Typeface.createFromAsset(context.getAssets(),
            "path: \"DAYPBL.ttf\"");

        this.setTypeface(face);
    }

    protected void onDraw (Canvas canvas) {
        super.onDraw(canvas);
    }
}
```

Рисунок Б.19 – Лістнінг класу CustomTextView

Додаток Г – Графічна частина

ГРАФІЧНА ЧАСТИНА

**РОЗРОБКА ПРОГРАМИ ДЛЯ ВИВЧЕННЯ І ТЕСТУВАННЯ ЗНАНЬ З
ПОЛЬСЬКОЇ МОВИ**

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА
КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ
КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Бакалаврська дипломна робота на тему:

«Розробка програми для вивчення і тестування з польської мови»

Автор: ст. групи ЗП-186 Маніч А. М.

Науковий керівник: к.т.н., доц. каф. ПЗ Ракитянська Г.Б

Вінниця - 2022

Рисунок Г.1 – Титульний слайд

Актуальність теми

Зараз, як ніколи, стрімкий ріст інформаційних та комп'ютерних технологій обумовлює розвиток і поширення інформаційних ресурсів. Мобільні пристрої, а з ними і мобільні додатки, досить міцно увійшли в життя сучасної людини. В сьогоденні важко уявити людину, яка б замість того, щоб шукати якусь інформацію в інтернеті, буде відкривати для цього підручник або енциклопедію. Оскільки програми для вивчення мови стають все доступнішими, люди, які хочуть вивчити нову мову, звертаються за допомогою до своїх мобільних пристроїв. В інтернеті існує множина платних і безкоштовних програм для вивчення майже всіх мов світу. Ці програми дозволяють користувачам вивчати нову мову з будь-якого місця в будь-який час, як онлайн, так і офлайн. Також вони є досить ефективними для тих, хто вивчає нову мову для подорожей та спілкування на простому рівні, оскільки в таких програмах часто є слова або ж фрази, необхідні для туристів, які знаходяться в іншій країні вперше.

Рисунок Г.2 – Актуальність теми

Мета, об'єкт та предмет дослідження

- ❖ Метою дослідження є підвищення ефективності вивчення користувачем польської мови шляхом розробки та використання спеціалізованої мобільної навчальної системи з теоретичним матеріалом і тестами, що дозволить підвищити рівень володіння новою мовою.
- ❖ Об'єкт дослідження – процеси створення мобільної навчальної системи для вивчення польської мови.
- ❖ Предмет дослідження – засоби програмної реалізації мобільного застосунку для вивчення польської мови.

Рисунок Г.3 – Мета, об'єкт та предмет дослідження

Задачі дослідження

- ❖ Визначити найбільш ефективний метод вивчення іноземної мови;
- ❖ Розглянути можливі шляхи вирішення поставленої задачі;
- ❖ Розробити метод і модель навчальної системи;
- ❖ Провести варіантний аналіз та обґрунтувати вибір програмного забезпечення для реалізації розроблюваного програмного засобу.
- ❖ Розробити інтерфейс власного програмного продукту;
- ❖ Розробити модулі із завданнями;
- ❖ Реалізувати аудіо супровід при вивченні мови;
- ❖ Розробити алгоритм генерування тестів для вивчення мови;
- ❖ Розробити алгоритм генерування текстових та графічних компонентів для зручного вивчення нових слів та словосполучень;
- ❖ Провести тестування розробленого програмного продукту.

Рисунок Г.4 – Задачі дослідження

Новизна отриманих результатів

Подальшого розвитку отримав метод створення мобільної навчальної системи для вивчення польської мови, який, на відміну від існуючих, реалізує логічні зв'язки між компонентами для синхронізації модульних блоків з теоретичним і словниковим матеріалом, аудіо супроводом і тестами, що підвищує ефективність процесу вивчення нової мови.

Подальшого розвитку отримала модель мобільної навчальної системи для вивчення польської мови, яка, на відміну від існуючих, орієнтована на покращенні навчального процесу шляхом реалізації комплексного функціоналу для вивчення основ польської мови з поєднанням словникового матеріалу, аудіо супроводу та тестів, що дозволить реалізувати універсальний підхід до розробки доступного навчального Android додатку і забезпечить тренувальні потреби користувача в процесі вивчення нової мови.




Рисунок Г.5 – Новизна отриманих результатів

Практична цінність отриманих результатів

Практична цінність одержаних результатів полягає у створенні мобільної навчальної системи для вивчення польської мови, що дозволить опанувати нову мову користувачам Android застосунків.




Рисунок Г.6 – Практична цінність отриманих результатів

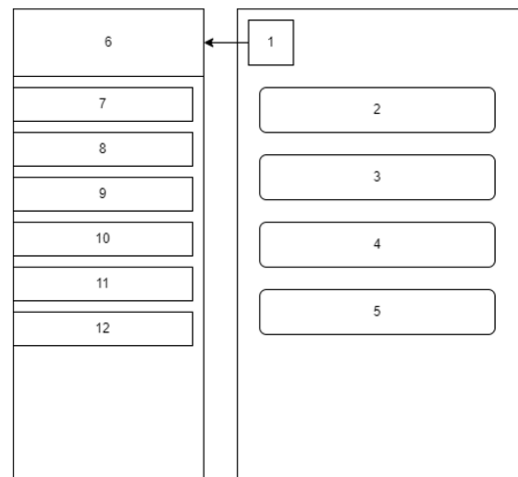
Порівняльний аналіз аналогів

Критерій	Duolingo	Speechling	iTalki	Tandem	Pollang
Тестування вивчаємої мови	1	1	0	0	1
Словник зі словами	0	0	0	0	1
Безкоштовність повного функціоналу	0	1	0	0	1
Аудіовідтворення іноземних слів	1	1	0	0	1
Спілкування з іншими учасниками	0	0	1	1	0
Підсумковий результат	2	3	1	1	4

Рисунок Г.7 – Порівняльний аналіз аналогів

Графічна схема головного вікна додатку

1. Кнопка „Menu”.
2. Кнопка „Проходження тестування”.
3. Кнопка „Проходження тестування”.
4. Кнопка „Проходження тестування”.
5. Кнопка „Проходження тестування”.
6. Назва додатку.
7. Пункт «Головна сторінка».
8. Пункт «Інформація».
9. Пункт «Очистити всі дані».
10. Пункт «Вивчення матеріалів».
11. Пункт «Створити власний тест».
12. Пункт «Видогуватись».



БІЛОР-СХЕМЛІ

Рисунок Г.8 – Графічна схема головного вікна додатку

Метод роботи навчальної системи

Метод роботи навчальної системи включає послідовність дій:

1. Авторизувавшись у системі, користувач отримує доступ до навчальних матеріалів, до блоків нових слів, до аудіо підтримки з озвучення слів і до блоку створення тестів, що включають в себе блоки змінних тесту.
2. На початку процесу створення тесту деякі блоки уже містять першу характеристику тесту та відповідні йому варіанти відповідей.
3. Після заповнення всіх полів тесту потрібно призначити об'єкту тест його рівень складності.
4. Згодом викликається метод зберігання даних до масиву.
5. Зберігання даних до бази даних. Якщо дані коректні – перейти до пункту 7, якщо ні – відображується повідомлення про помилку збереження тесту.
6. Зберегти дані до бази даних.

Рисунок Г.9 – Метод роботи навчальної системи

Метод роботи навчальної системи (продовження)

7. Для проходження новоствореного тесту потрібно використати функцію тестування.
8. Обиравши тему та рівень тестування, можна побачити створений тест.
9. Далі можна спостерігати отримування даних з бази даних та їх відображення за допомогою функції рандому (також використовується функція, яка спочатку фільтрує питання по рівнях).
10. Після чого починається початок тестування та відображення всіх тестів.
11. Збереження результатів тестування.

Рисунок Г.10– Метод роботи навчальної системи (продовження)

Блок-схема алгоритму генерування та додавання в БД тестів

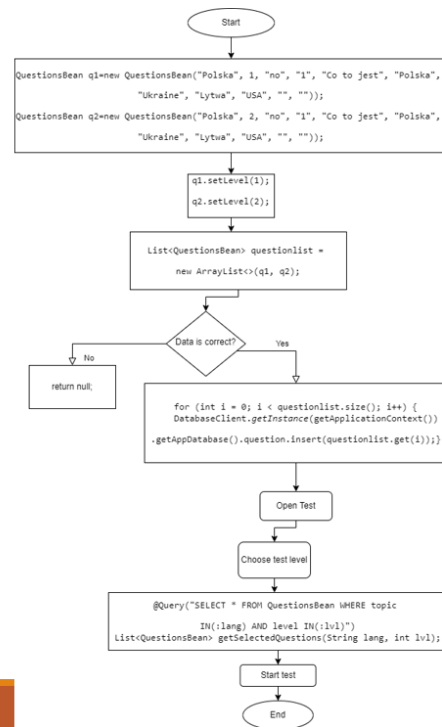


Рисунок Г.11 – Блок-схема алгоритму генерування та додавання в БД тестів

Блок-схема моделі роботи навчальної системи

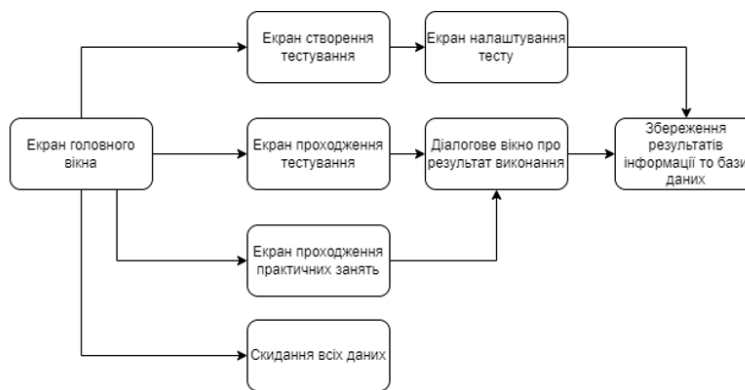


Рисунок Г.12 – Блок-схема моделі роботи навчальної системи

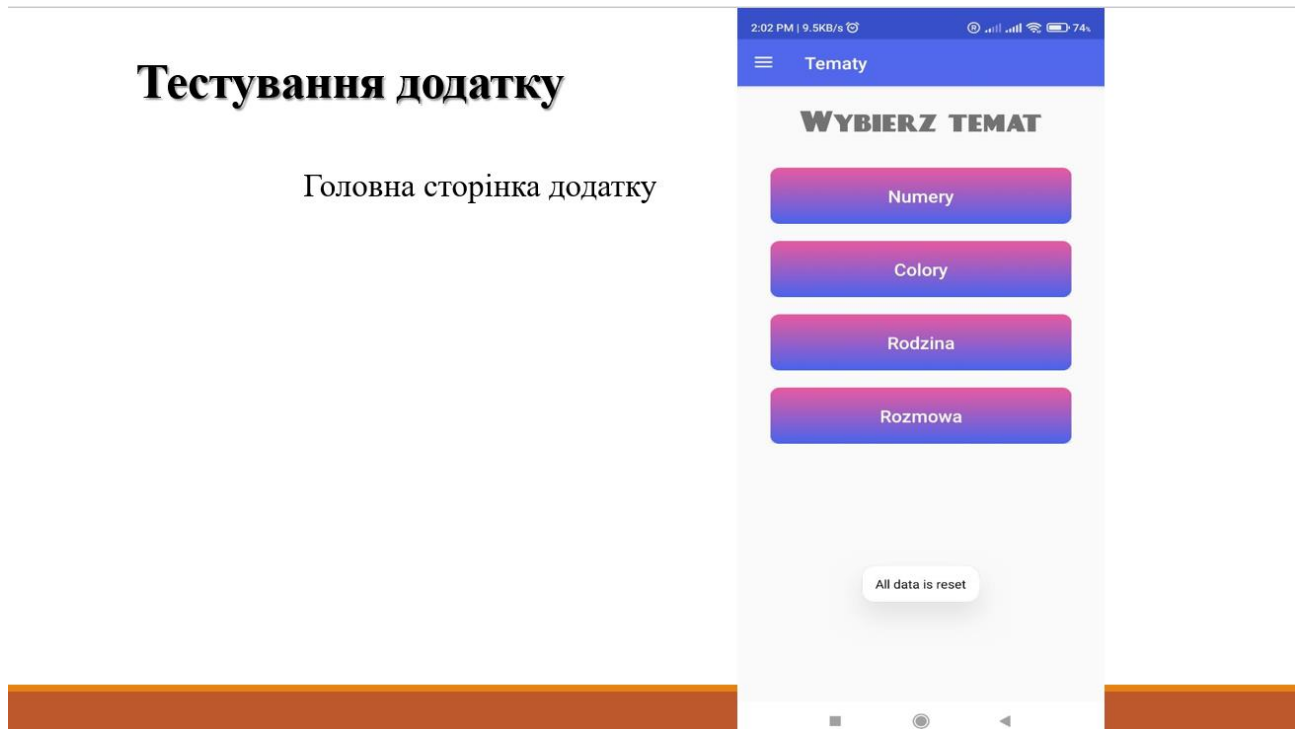


Рисунок Г.13 – Тестування головної сторінки додатку

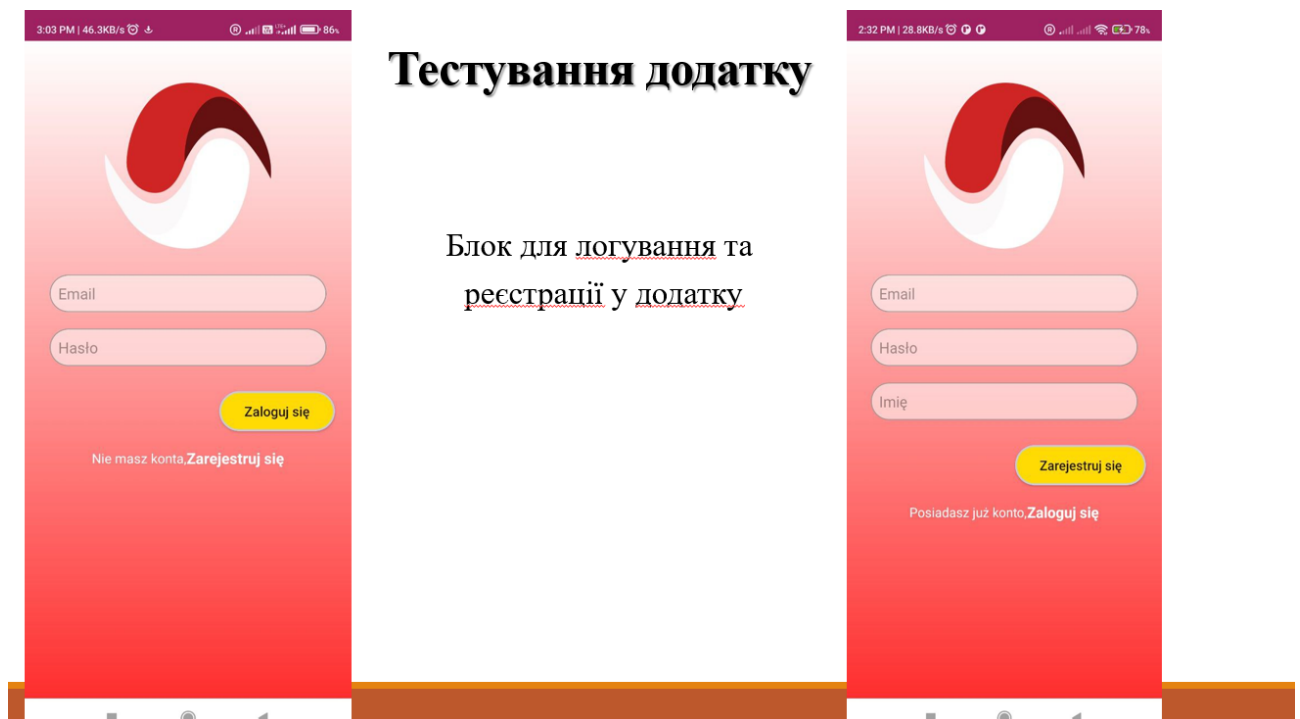


Рисунок Г.14 – Блок для логуювання та реєстрації у додатку

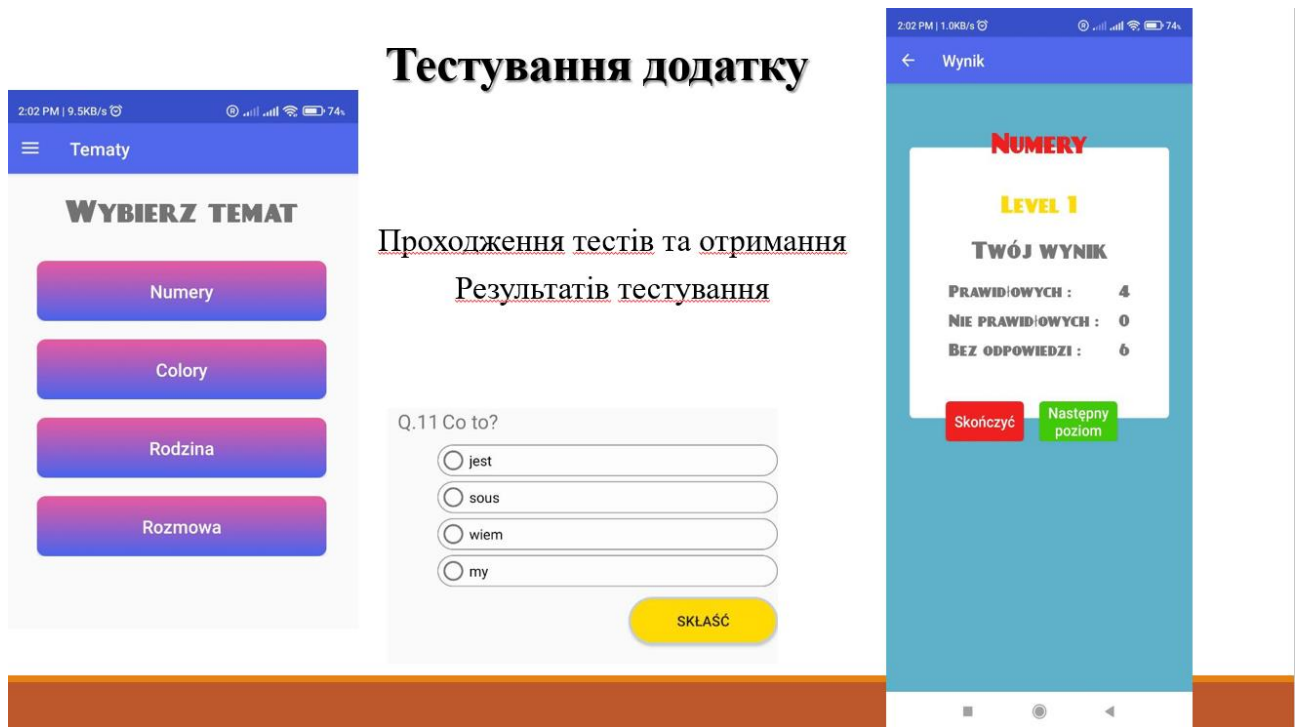


Рисунок Г.15 – Проходження тестів та отримання
Результатів тестування

Тестування додатку

Проходження теоретичних матеріалів з
аудіосупровідом

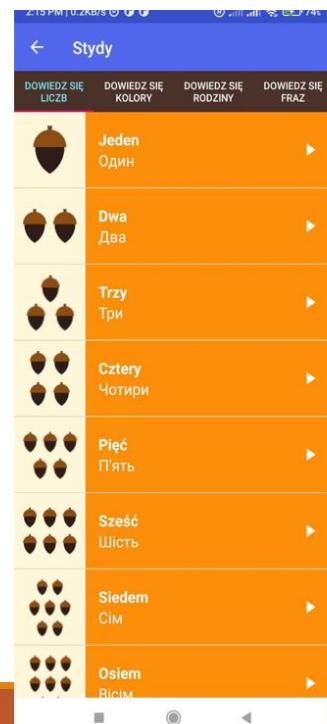


Рисунок Г.16 – Проходження теоретичних матеріалів з аудіосупровідом

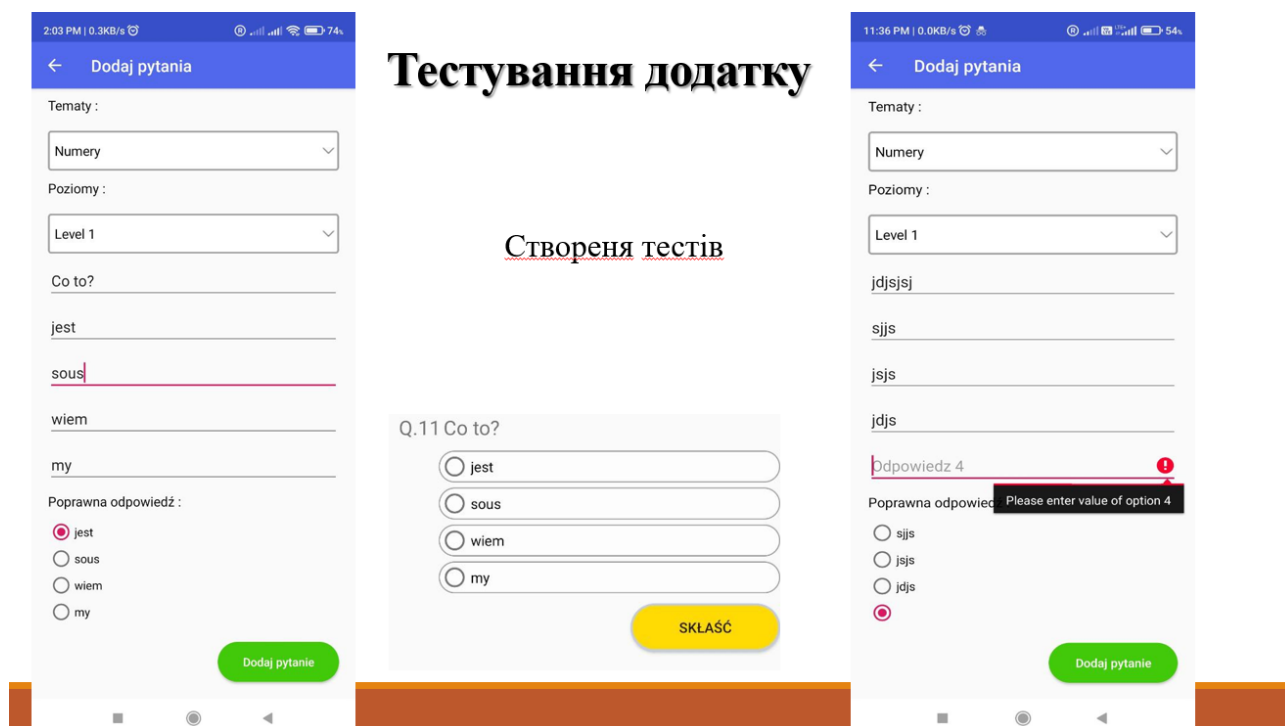


Рисунок Г.17 – Створення тестів

Апробація результатів роботи

Результати роботи доповідалися на Всеукраїнській науково-практичній інтернет-конференції "Молодь в науці: дослідження, проблеми, перспективи - 2022", Секція - Інформаційні технології та комп'ютерна інженерія.

Наукова публікація:

Войтко В.В. Навчальна система для вивчення і тестування з польської мови / В. В. Войтко, Г. Б. Ракитянська, С. В. Бевз, С. М. Бурбело, А. М. Манич // Матеріали Всеукраїнської науково-практичної інтернет-конференції "Молодь в науці: дослідження, проблеми, перспективи - 2022", Секція - Інформаційні технології та комп'ютерна інженерія. [Електронний ресурс] – Режим доступу:

https://conferences.vntu.edu.ua/index.php/mm/mm_2022/paper/viewFile/16201/13643

Рисунок Г.18 – Апробація та публікація результатів роботи