Міністерство освіти і науки України
Університет митної справи та фінансів

Факультет інноваційних технологій
Кафедра комп'ютерних наук та інженерії програмного забезпечення

Кваліфікаційна робота бакалавра

на тему: «Розробка мобільного додатку для вивчення англійської мови в інтегрованому середовищі Android Studio»

Виконав: студент групи _____ІПЗ20-2_____

Спеціальність 121 «Інженерія програмного забезпечення»

_____Баталов Владислав Андрійович_____

(прізвище та ініціали)

Керівник к.т.н., доцент Чупілко Т.А.___

Рецензент Університет митної справи та фінансів,

(місце роботи)

доцент кафедри кібербезпеки та інформаційних технологій_____

(посада)

к.т.н. Савченко Ю.В.

(науковий ступінь, вчене звання, прізвище та ініціали)

Дніпро – 2024

АНОТАЦІЯ

Баталов В.А. Розробка мобільного додатку для вивчення англійської мови в інтегрованому середовищі Android Studio.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за спеціальністю 121 «Інженерія програмного забезпечення». – Університет митної справи та фінансів, Дніпро, 2024.

Дана кваліфікаційна робота присвячена розробці мобільного додатку для вивчення англійської мови в інтегрованому середовищі Android Studio. Розглянуті методи та технічні засоби, необхідні для реалізації додатку, сформульовані вимоги до програмного забезпечення. Було проведено проектування архітектури та інтерфейсу користувача, а також розробку та тестування додатку. Отримані результати мають практичне значення, оскільки забезпечують автоматизацію та полегшення процесу вивчення англійської мови, надаючи користувачам зручний та ефективний інструмент для покращення мовних навичок. Розроблений додаток є сучасним та використовує передові технології мобільної розробки, такі як Android Studio та Kotlin, і є важливим кроком у покращенні доступності та якості навчальних ресурсів з англійської мови.

Розроблена система має клієнтську та серверну частини. Клієнтська частина - це мобільний додаток на Android, який отримує дані від користувача та відправляє їх на сервер для аналізу. На сервері використовується Firebase для аутентифікації користувачів, зберігання граматичних вправ та тестових питань у Realtime Database. Додатково система інтегрується із зовнішнім API для отримання визначень слів.

Ключові слова: розробка, мобільний додаток, англійська мова, тестування, Android Studio, Kotlin.

ABSTRACT

Batalov V.A. Development of a Mobile Application for Learning English in the Integrated Environment of Android Studio.

Qualification work for a bachelor's degree in speciality 121 «Software Engineering». – University of Customs and Finance, Dnipro, 2024.

This bachelor's thesis is dedicated to the development of a mobile application for learning English in the integrated environment of Android Studio. The methods and technical means necessary for the implementation of the application are considered, and the software requirements are formulated. The architecture and user interface design were carried out, as well as the development and testing of the application. The obtained results are of practical significance as they ensure the automation and facilitation of the English language learning process, providing users with a convenient and effective tool for improving language skills. The developed application is modern and uses advanced mobile development technologies such as Android Studio and Kotlin, and is an important step in improving the accessibility and quality of English language learning resources.

The developed system has client and server parts. The client part is an Android mobile application that receives data from the user and sends it to the server for analysis. The server uses Firebase for user authentication, storage of grammatical exercises, and test questions in Realtime Database. Additionally, the system integrates with an external API to obtain word definitions.

Keywords: development, mobile application, English language, testing, Android Studio, Kotlin.

ЗМІСТ

ВСТУП

Актуальність дослідження. Англійська мова є однією з найпоширеніших мов світу та відіграє провідну роль у міжнародному спілкуванні, бізнесі, науці, технологіях та культурі. Уміння вільно володіти англійською мовою стає невід'ємним елементом особистісного та професійного розвитку в сучасному глобалізованому світі.

Незважаючи на те, що англійська мова широко викладається в навчальних закладах, багато людей стикаються зі значними труднощами у процесі її вивчення. Традиційні методи навчання, такі як підручники, аудіозаписи та заняття з викладачем, не завжди є достатньо ефективними та зручними для всіх категорій тих, хто навчається. Вони можуть бути обмежені часом, місцем або темпом навчання.

У той же час, популярність мобільних пристроїв, таких як смартфони та планшети, стрімко зростає в усьому світі. Мобільні технології відкривають нові можливості для інтерактивного та персоналізованого навчання, дозволяючи користувачам вивчати англійську мову у зручний для них час і темп, а також максимально пристосовувати процес до їхніх індивідуальних потреб та рівня знань.

Розробка якісного мобільного додатку для вивчення англійської мови, який поєднує сучасні методики викладання, інтерактивні вправи, словники та можливості відстеження прогресу, є актуальною проблемою. Такий додаток може значно полегшити та прискорити процес вивчення мови, забезпечуючи зручний доступ до навчальних матеріалів у будь-який час і в будь-якому місці.

Інтегроване середовище Android Studio, яке використовується для створення додатків на платформі Android, пропонує широкий спектр інструментів та бібліотек, що дозволяють реалізувати сучасний та функціональний мобільний додаток для вивчення англійської мови. Такий додаток може стати потужним інструментом для підвищення рівня володіння

англійською мовою та сприяти розширенню можливостей для особистісного та професійного зростання користувачів.

Метою роботи є розробка сучасного та зручного у використанні мобільного додатку для вивчення англійської мови в інтегрованому середовищі Android Studio.

Методи дослідження: аналіз вимог до розроблюваного програмного забезпечення, проектування архітектури додатку, програмування із застосуванням мови Kotlin та інструментів Android Studio, тестування та налагодження розробленого додатку.

У відповідності до поставленої мети в кваліфікаційній роботі ставились та вирішувались наступні завдання дослідження:

1. Проаналізувати методи та технічні засоби, що застосовуються для розробки, та обрати необхідні для реалізації мобільного додатку для вивчення англійської мови в інтегрованому середовищі Android Studio.
2. Розробити вимоги до програмного забезпечення для мобільного додатку для вивчення англійської мови в інтегрованому середовищі Android Studio на основі аналізу переваг та недоліків існуючих систем.
3. Спроектувати та реалізувати нову систему на основі аналізу потреб користувачів.
4. Провести тестування системи.

Об'єктом дослідження є процес розробки мобільних додатків для вивчення англійської мови.

Предметом дослідження є програмний комплекс, реалізований у вигляді мобільного додатку для вивчення англійської мови в інтегрованому середовищі Android Studio.

Практичне значення одержаних результатів – автоматизувати та полегшити процес вивчення англійської мови за допомогою розробленого мобільного додатку, який забезпечує зручний та ефективний інструментарій для покращення мовних навичок користувачів.

Структура роботи:

Розділ 1 Дослідження предметної області. Постановка завдань дослідження. У даному розділі буде виконано пошук та аналіз публікацій стосовно теми розробки мобільного додатку для вивчення англійської мови.

Розділ 2 Аналіз засобів реалізації мобільного додатку для вивчення англійської мови. В даному розділі буде проаналізовано та обрано технології для реалізації мобільного додатку для вивчення англійської мови.

Розділ 3 Розробка мобільного додатку для вивчення англійської мови в інтегрованому середовищі Android Studio. В даному розділі буде спроектовано та розроблено мобільний додаток для вивчення англійської мови.

В результаті виконання кваліфікаційної роботи було набуто фахові компетентності та підтверджені наступні результати навчання, які відповідають Освітньо-Професійній програмі за спеціальністю 121«Інженерія програмного забезпечення»:

К01. Здатність до абстрактного мислення, аналізу та синтезу.

К02. Здатність застосовувати знання у практичних ситуаціях.

К03. Здатність спілкуватися державною мовою як усно, так і письмово.

К06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

К13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення;

К14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування;

К15. Здатність розробляти архітектури, модулі та компоненти програмних систем;

К16. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.

К19. Володіння знаннями про інформаційні моделі даних та системи, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних;

К22. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя;

К26. Здатність до алгоритмічного та логічного мислення.

ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПР05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об’єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.

ПР06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПР08. Вміти розробляти людино-машинний інтерфейс.

ПР12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.

ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

ПР14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

ПР21. Знати, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв’язуваних прикладних завдань та створюваних програмних систем.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ

1.1. Аналіз публікацій щодо розробки мобільного додатку

В епоху глобалізації та стрімкого розвитку інформаційних технологій англійська мова стала невід'ємною частиною життя сучасної людини. За даними Британської Ради, близько 1,5 мільярда людей у світі вивчають або використовують англійську мову як іноземну. Вільне володіння англійською відкриває нові можливості для кар'єрного зростання, міжнародного спілкування та доступу до величезного обсягу інформації. У цьому контексті мобільні додатки для вивчення англійської мови стають все більш затребуваними, оскільки пропонують зручний, гнучкий та інтерактивний спосіб опанування нових мовних навичок.

За даними аналітичної компанії App Annie, глобальний ринок освітніх мобільних додатків у 2022 році досяг 38,5 мільярдів доларів, демонструючи безперервне зростання попиту на цифрові рішення для навчання. Тому аналіз та вдосконалення мобільних додатків для вивчення англійської мови є актуальним питанням для забезпечення якісної мовної освіти та задоволення потреб сучасних користувачів.

У статті "Розробка мобільного додатку для вивчення англійської мови" [1] автори відзначають зростаючу популярність мобільних додатків для вивчення англійської мови серед користувачів смартфонів. Вони наголошують на необхідності створення багатофункціонального, зрозумілого та доступного додатку для вивчення англійської.

Автори висвітлюють загальну концепцію такого додатку, яка включає:

1. Можливість вивчати англійську в офлайн-режимі.
2. Різнобічне вивчення (слова, граматика).
3. Наявність мотиваційного блоку.
4. Простий дизайн інтерфейсу.

Стаття підкреслює переваги мобільних додатків для навчання, такі як портативність, зручність та доступність у будь-який час та будь-якому місці [1].

Наводяться статистичні дані про зростання кількості користувачів смартфонів (близько 75% людей володіють смартфонами) та попит на освітні мобільні додатки (ринок освітніх додатків досяг 38,5 мільярдів доларів у 2022 році) [1].

Автори аналізують різні типи існуючих мобільних додатків для вивчення англійської: спеціалізовані додатки для курсів та загальні додатки. На основі проведеного аналізу вони прийшли до рішення створити додаток з можливістю кастомізованого вибору місця початку та продовження навчання [1].

У статті представлено пропонований інтерфейс головної сторінки розроблюваного додатку.

У публікації "Дидактичний потенціал мобільних застосунків для вивчення англійської мови як іноземної" [2] автори Н.М. Блинова, О.В. Кирилова та М.В. Долженко проводять огляд та аналіз мобільних додатків, доступних в Google Play, які призначені для вивчення англійської мови студентами-нефілологами в Україні. Публікація охоплює кілька ключових питань:

1. Актуальність використання мобільних технологій у навчанні (m-learning) та вивченні іноземних мов (MALL - mobile assisted language learning). Автори наводять огляд попередніх досліджень у цій галузі, підкреслюючи зростаючу роль мобільних додатків в освіті [2].

2. Аналіз 30 мобільних додатків з Google Play, призначених для вивчення англійської мови. Автори класифікують їх за різними критеріями: вік цільової аудиторії, рівень мовних навичок, експертність розробників, кількість завантажень тощо [2].

3. Детальний розгляд 8 відібраних застосунків, які можуть бути корисними для студентів 1 курсу рівнів elementary та pre-intermediate. Для кожного додатку наведено опис функціоналу, особливостей, переваг та недоліків [2].

4. Виявлення спільних рис проаналізованих додатків, таких як реалізація концепції "edutainment" (поєднання навчання та ігрової форми), акцент на граматиці та лексиці, використання аудіо з голосом носіїв мови, інтерактивні вправи та тести [2].

5. Обговорення дидактичного потенціалу мобільних додатків як допоміжного матеріалу у викладанні та вивченні англійської мови, а також їх ролі у формуванні мовних компетенцій студентів [2].

Автори роблять висновок, що мобільні додатки не замінюють основний підручник, але є корисним сучасним дидактичним матеріалом, зрозумілим для студентів [2]. Вони закликають до подальших досліджень у галузі m-learning та використання мобільних технологій у викладанні інших дисциплін.

Загалом, ця публікація є ґрунтовним оглядом наявних мобільних застосунків для вивчення англійської мови та їх дидактичних можливостей в українському освітньому контексті.

Публікація [3] присвячена актуальній проблемі використання мобільних додатків у процесі вивчення англійської мови. Автори розглядають мобільне навчання як важливий освітній інструмент і сучасний напрям в освіті, характерною ознакою якого є створення освітнього середовища [3]. Зазначається, що завданням викладача є допомогти студентам вибрати необхідні ресурси, які можуть максимально сприяти вивченню мови, індивідуалізуючи процес навчання.

На думку авторів, практичне застосування мобільних додатків має великий потенціал для підвищення ефективності процесу вивчення іноземних мов, але водночас інтеграція роботи з додатками в структурі практичного заняття вносить певні проблеми. Тому автори вважають, що мобільні додатки можуть бути більш ефективно використані для організації та інтенсифікації самостійної роботи студентів (переважно позааудиторної) [3].

У публікації наведено короткий огляд мобільних додатків для навчання англійської мови, таких як Sounds Right, Sounds: Pronunciation App, додатки BBC, Learn English Grammar, Johnny Grammar's Word Challenge, EnglishGrammarTest,

MyWordBook та інші. Автори відзначають, що ці додатки можуть ефективно використовуватися для самостійної роботи студентів з метою розвитку різних мовних навичок: аудіювання, вимови, граматики, лексики [3].

Загалом, автори роблять висновок, що широкий спектр та різноманітність існуючих мобільних навчальних ресурсів дозволяють вибирати додатки відповідно до індивідуальних потреб та рівня підготовки студентів. Практика використання додатків засвідчує їх перевагу над традиційними методами навчання з огляду на інтенсифікацію самостійної діяльності, індивідуалізацію, підвищення пізнавальної активності та мотивації студентів до вивчення іноземної мови [3].

1.2. Аналіз методів розробки мобільного додатку

Розробка мобільного додатку для вивчення англійської мови вимагає ретельного вибору методів та технологій, щоб забезпечити високу якість, ефективність та зручність користування. У цьому розділі було розглянуто різні підходи та інструменти, які можуть бути використані для створення такого додатку.

Першим кроком у розробці мобільного додатку є вибір цільової платформи та середовища розробки. Найпопулярнішими мобільними операційними системами є Android від Google та iOS від Apple. У цій роботі ми зосередимося на розробці додатку для платформи Android, оскільки вона має найбільшу частку ринку та відкритий вихідний код, що дозволяє більш гнучку й економічну розробку.

Для розробки додатків на Android найбільш поширеним інструментом є Android Studio - офіційна інтегрована середа розробки (IDE) від Google. Android Studio пропонує потужний набір інструментів для написання коду, налагодження, тестування та розгортання додатків. Він підтримує кілька мов програмування, таких як Java та Kotlin.

Kotlin є відносно новою мовою програмування, розробленою JetBrains, яка все частіше використовується для розробки додатків на Android. Порівняно з Java, Kotlin пропонує більш сучасний, лаконічний та безпечний синтаксис, що підвищує продуктивність розробника та знижує ймовірність появи помилок. Крім того, Kotlin повністю сумісний з Java, що дозволяє інтегрувати існуючі Java-бібліотеки та фреймворки в нові проекти на Kotlin.

У контексті розробки мобільного додатку для вивчення англійської мови, Kotlin може бути кращим вибором, ніж Java, завдяки своїй лаконічності, безпеці та підтримці сучасних парадигм програмування, таких як функціональне програмування та розширення функціоналу.

Існує кілька основних методів, які використовуються в розробці програмного забезпечення, і кожен із них має свої переваги та недоліки. Розглянемо найпоширеніші методи розробки та їх застосування для створення мобільного додатку для вивчення англійської мови.

Агільні або гнучкі методології стали дуже популярними в останні роки, особливо при розробці програмного забезпечення для мобільних пристроїв. Ці методології базуються на ітеративному підході, де розробка відбувається короткими циклами, що називаються ітераціями або спринтами. Кожна ітерація включає в себе планування, розробку, тестування та перегляд проекту.

Одним із найвідоміших гнучких методів є Scrum. Scrum передбачає наявність Scrum-команди, яка складається з Product Owner (власника продукту), Scrum Master (фасилітатора процесу) та розробників. Product Owner відповідає за визначення вимог до продукту та їх пріоритизацію, а Scrum Master забезпечує дотримання процесу Scrum та усуває перешкоди в роботі команди. Розробники виконують роботу відповідно до плану ітерації (спринту).

Перевагами використання гнучких методологій при розробці мобільного додатку для вивчення англійської мови є:

1. Швидка адаптація до змін вимог замовника або користувачів.
2. Можливість оперативно коригувати функціональність додатку на основі зворотного зв'язку від користувачів.

3. Регулярні релізи нових версій додатку, що дозволяє швидко отримувати відгуки та покращувати продукт.

4. Більш тісна співпраця між членами команди та замовниками.

5. Гнучкість та можливість швидкого реагування на непередбачувані ситуації.

З іншого боку, недоліками гнучких методологій можуть бути:

1. Необхідність постійного залучення замовника або представника користувачів до процесу розробки.

2. Складність планування та оцінки вартості проекту на початковому етапі, оскільки вимоги можуть змінюватися.

3. Потреба в досвідчених та кваліфікованих розробниках, які можуть ефективно працювати в гнучкому середовищі.

Загалом, гнучкі методології добре підходять для розробки мобільних додатків, оскільки вони дозволяють швидко реагувати на зміни вимог та тенденцій у сфері мобільних технологій, а також забезпечують тісну співпрацю між розробниками, замовниками та кінцевими користувачами.

Каскадна модель, також відома як лінійна або послідовна модель, є одним із найстаріших і найбільш поширених методів розробки програмного забезпечення. Ця модель передбачає, що процес розробки складається з послідовних фаз, де кожна наступна фаза починається лише після завершення попередньої. Основними фазами каскадної моделі є:

1. Збір вимог та їх аналіз.

2. Проектування системи та архітектури.

3. Реалізація (програмування).

4. Інтеграція та тестування.

5. Впровадження та експлуатація.

6. Супровід.

У каскадній моделі перехід до наступної фази відбувається лише після завершення поточної фази. Це забезпечує чітку структуру процесу розробки та дозволяє уникнути невизначеностей і непорозумінь.

Переваги використання каскадної моделі при розробці мобільного додатку для вивчення англійської мови:

1. Чітке визначення вимог та планування на початковому етапі, що забезпечує стабільність та передбачуваність проекту.
2. Добре структурований процес розробки з чітким розподілом обов'язків та відповідальності між учасниками проекту.
3. Простота у відстеженні прогресу та контролі над проектом.
4. Відповідність вимогам стандартів та норм, що важливо в освітньому програмному забезпеченні.

Однак, каскадна модель також має певні недоліки:

1. Низька гнучкість та складність у внесенні змін на пізніх стадіях розробки.
2. Відсутність зворотного зв'язку від користувачів до завершення проекту, що може призвести до невідповідності їх очікуванням.
3. Можливий простій ресурсів на певних фазах розробки через жорстку послідовність етапів.
4. Складність у визначенні всіх вимог на початковому етапі, особливо для інноваційних проектів.

Незважаючи на ці недоліки, каскадна модель все ще широко використовується у розробці програмного забезпечення, особливо в проектах з чітко визначеними та стабільними вимогами, де необхідно дотримуватися жорстких стандартів та норм.

Ітеративна модель з проміжним плануванням (Iterative with Intermediate Design) Ітеративна модель з проміжним плануванням є гібридним підходом, який поєднує переваги ітеративного розвитку з чітким проектуванням на кожній ітерації. Ця модель передбачає розбиття розробки на кілька циклів або ітерацій, кожна з яких включає такі фази:

1. Планування та визначення вимог для поточної ітерації.
2. Аналіз вимог та проектування архітектури для поточної ітерації.

3. Реалізація (програмування) функціоналу згідно з вимогами поточної ітерації.

4. Тестування реалізованого функціоналу.

5. Оцінка та інтеграція проміжного рішення.

На відміну від класичної ітеративної моделі, де процес проектування відбувається лише на початковому етапі, в цій на відміну від класичної ітеративної моделі, де процес проектування відбувається лише на початковому етапі, в цій моделі проектування виконується перед кожною ітерацією. Це дозволяє забезпечити більш ретельне проектування та уникнути ризиків, пов'язаних із внесенням змін на пізніх стадіях розробки.

Переваги використання ітеративної моделі з проміжним плануванням при розробці мобільного додатку для вивчення англійської мови:

1. Гнучкість та можливість внесення змін у вимоги на кожній ітерації на основі зворотного зв'язку від користувачів або замовників.

2. Регулярні проміжні релізи додатку, що дозволяє отримувати своєчасний відгук та коригувати подальший розвиток продукту.

3. Поступове розширення функціоналу додатку з урахуванням пріоритетів та потреб користувачів.

4. Зниження ризиків за рахунок ретельного проектування на кожній ітерації.

5. Можливість раннього залучення користувачів та отримання їх реакції на проміжні версії додатку.

Недоліками даної моделі можуть бути:

1. Підвищені витрати на проектування через необхідність виконувати його на кожній ітерації.

2. Складність у визначенні загальної картини проекту на початковому етапі, оскільки повний функціонал додатку не визначено.

3. Потреба у високій кваліфікації розробників, здатних ефективно працювати в ітеративному режимі.

Незважаючи на ці недоліки, ітеративна модель з проміжним плануванням є популярним підходом у розробці мобільних додатків, оскільки забезпечує необхідну гнучкість та можливість коригувати напрямок розвитку продукту відповідно до потреб користувачів та ринкових тенденцій.

Спіральна модель (Spiral) Спіральна модель розробки програмного забезпечення є ітераційною моделлю, яка поєднує елементи гнучкої та планової розробки. Ця модель була запропонована Баррі Боемом у 1988 році та базується на ітераційному підході з акцентом на аналіз ризиків.

У спіральній моделі процес розробки складається з кількох ітерацій або циклів, кожен з яких включає чотири основні фази:

1. Планування: На цій фазі визначаються цілі та завдання поточної ітерації, проводиться аналіз ризиків та планування способів їх зменшення.
2. Аналіз ризиків: Детальний аналіз ризиків, пов'язаних із поточною ітерацією, та оцінка їх впливу на проект.
3. Розробка та тестування: Реалізація функціоналу згідно з планом ітерації, включаючи проектування, програмування та тестування.
4. Оцінка: Оцінка результатів поточної ітерації та прийняття рішення про перехід до наступної ітерації або завершення проекту.

Кожна наступна ітерація базується на результатах попередньої, враховуючи накопичені знання та досвід.

Переваги використання спіральної моделі при розробці мобільного додатку для вивчення англійської мови:

1. Гнучкість та можливість адаптації до змін вимог на кожній ітерації.
2. Акцент на управлінні ризиками, що дозволяє виявляти та усувати потенційні проблеми на ранніх стадіях розробки.
3. Можливість залучення користувачів на різних етапах розробки для отримання зворотного зв'язку.
4. Регулярні проміжні релізи додатку, що дозволяє відстежувати прогрес та вносити необхідні зміни.

Недоліками спіральної моделі можуть бути:

1. Підвищені витрати на управління проектом через складність координації між ітераціями.

2. Необхідність високої кваліфікації розробників та досвіду в управлінні ризиками.

3. Можлива складність у визначенні критеріїв завершення проекту через ітераційний характер розробки.

Спіральна модель добре підходить для розробки складних та ризикованих проектів, де важливо ретельно аналізувати та управляти ризиками. У випадку мобільного додатку для вивчення англійської мови, ця модель може бути корисною, оскільки дозволяє гнучко реагувати на зміни вимог та тенденції у сфері освітніх технологій, а також забезпечує управління ризиками, пов'язаними з інтеграцією різних компонентів додатку (навчальний контент, ігрові елементи, зовнішні API тощо).

Крім розглянутих основних моделей, існують також деякі інші підходи та методи, які можуть бути використані при розробці мобільного додатку для вивчення англійської мови. Наприклад, модель розробки за функціями (Feature-Driven Development, FDD) або модель розробки, орієнтована на компоненти (Component-Based Software Engineering, CBSE). Однак вони менш поширені та мають більш вузьку сферу застосування.

Вибір відповідної моделі або методології розробки залежить від багатьох факторів, таких як розмір проекту, вимоги до якості, наявні ресурси, кваліфікація команди розробників, терміни виконання та бюджет. Для успішної реалізації мобільного додатку для вивчення англійської мови необхідно ретельно проаналізувати всі ці фактори та обрати найбільш підходящу модель або комбінацію моделей, які забезпечать ефективну розробку, врахування вимог користувачів та відповідність освітнім стандартам.

Для забезпечення якісної, масштабованої та легкої в підтримці архітектури додатку, рекомендується використовувати сучасні архітектурні шаблони, такі як Model-View-ViewModel (MVVM) або Model-View-Presenter (MVP). Ці шаблони

допомагають відокремити логіку подання від бізнес-логіки та забезпечити кращу модульність та повторне використання коду.

MVVM є популярним шаблоном для додатків Android, де ViewModel виступає проміжною ланкою між моделлю даних (Model) та інтерфейсом користувача (View). ViewModel забезпечує підготовку та управління даними для подання, а також підтримує циклічність через збереження стану після повороту екрану або інших змін конфігурації.

Крім того, для забезпечення кращої організації коду та підвищення його модульності, рекомендується використовувати архітектурні компоненти Android, такі як Room для роботи з базами даних, LiveData та ViewModel для управління життєвим циклом і даними, а також Dagger для впровадження залежностей.

Для забезпечення онлайн-функціоналу, такого як аутентифікація користувачів, синхронізація даних та доступ до он-лайн ресурсів, додаток повинен мати можливість взаємодіяти з віддаленими серверами та хмарними сервісами. Для цього можна використовувати бібліотеки, такі як Retrofit для взаємодії з REST API або Firebase для розробки бекенду в якості сервісу (BaaS).

Firebase є потужною хмарною платформою від Google, яка пропонує різноманітні інструменти та сервіси для розробки мобільних додатків, включаючи аутентифікацію користувачів, хмарну базу даних Realtime Database, хмарне сховище файлів Cloud Storage та інші функції. Використання Firebase може значно спростити процес розробки та дозволити зосередитися на основній логіці додатку, делегуючи важливі завдання хмарній інфраструктурі.

Для ефективного вивчення англійської мови важливо забезпечити високоякісну роботу з аудіо- та відеоматеріалами. Android надає кілька API та бібліотек для роботи з мультимедійним контентом.

Для відтворення аудіо можна використовувати MediaPlayer або ExoPlayer - бібліотеку з відкритим вихідним кодом від Google для відтворення мультимедійного контенту. ExoPlayer забезпечує кращу підтримку різних форматів мультимедіа та має більше можливостей для налаштування порівняно з MediaPlayer.

Для синтезу мовлення з тексту можна використовувати API Text-to-Speech від Google, який дозволяє додатку виводити текстові повідомлення у вигляді синтезованого мовлення. Цей функціонал може бути корисним для вправ з аудіювання та вимови.

Для відображення відеоматеріалів можна використовувати VideoView або TextureView разом з MediaPlayer або ExoPlayer. Крім того, для більш складних випадків використання відео можна розглянути стороннє рішення, таке як ExoPlayer.

Для забезпечення високої якості розробленого додатку важливо приділити увагу тестуванню на різних рівнях, включаючи модульне тестування (unit testing), тестування інтерфейсу користувача (UI testing) та інтеграційне тестування, застосовувати підходи безперервної інтеграції та безперервного розгортання.

Android Studio надає інструменти для налаштування та виконання автоматизованих тестів, такі як JUnit для модульного тестування та Espresso для тестування інтерфейсу користувача.

Також потрібно враховувати оптимізацію продуктивності та використання ресурсів пристрою, а також питання доступності та локалізації для забезпечення зручності користування додатком для широкого кола користувачів.

Успішне виконання поставлених задач дозволить створити якісний, ефективний та зручний у використанні мобільний додаток для вивчення англійської мови, який відповідатиме сучасним вимогам та тенденціям у галузі мобільного навчання та опанування іноземних мов.

Ретельно спроектований та розроблений мобільний додаток буде корисним інструментом як для індивідуального навчання, так і для використання в освітніх закладах як допоміжний ресурс. Він дозволить студентам та всім, хто вивчає англійську мову, ефективно розвивати свої мовні навички, підвищувати рівень володіння мовою та розширювати можливості для кар'єрного зростання та міжнародного спілкування.

Крім того, успішна реалізація проекту може стати важливим кроком у розвитку мобільних технологій для освіти та популяризації концепції mLearning (мобільного навчання) в Україні. Це відкриває перспективи для подальших досліджень та розробок у галузі створення інноваційних цифрових навчальних ресурсів для різних предметних областей.

Модульне тестування дозволяє перевірити окремі компоненти та функції додатку, забезпечуючи їх коректну роботу в ізоляції. Тестування інтерфейсу користувача імітує взаємодію користувача з додатком, перевіряючи правильність відображення інформації та реагування на користувацькі дії.

Крім того, для забезпечення якості та стабільності додатку, рекомендується використовувати безперервну інтеграцію (Continuous Integration, CI) та безперервне розгортання (Continuous Deployment, CD). Ці практики дозволяють автоматизувати процеси збірки, тестування та розгортання додатку, що значно підвищує ефективність розробки та полегшує виявлення та виправлення помилок на ранніх стадіях.

Під час розробки мобільного додатку важливо приділяти увагу оптимізації продуктивності та ефективного використання ресурсів пристрою, таких як пам'ять, процесор та батарея. Це особливо актуально для додатків, орієнтованих на вивчення мови, оскільки вони можуть включати обробку великих обсягів даних та виконувати ресурсомісткі операції.

Для оптимізації продуктивності слід дотримуватися рекомендацій Google щодо розробки продуктивних додатків на Android. Це включає використання фонових потоків для виконання тривалих операцій, оптимізацію циклів життєвого циклу компонентів додатку, ефективне кешування даних та уникнення непотрібних обчислень або операцій з даними.

Крім того, важливо оптимізувати використання пам'яті, уникаючи витоків пам'яті та зменшуючи розмір використовуваних ресурсів, таких як зображення та мультимедійні файли. Для цього можна використовувати бібліотеки для стиснення зображень, лінивне завантаження ресурсів та кешування даних.

Для забезпечення широкого охоплення користувачів та зручності використання додатку важливо врахувати питання доступності та локалізації.

Доступність передбачає створення інтерфейсу користувача та функціоналу додатку, зручного для використання людьми з обмеженими можливостями, такими як порушення зору, слуху або рухової активності.

Android надає інструменти та рекомендації для розробки доступних додатків, включаючи підтримку спеціальних служб доступності, налаштування контрастності та розмірів елементів інтерфейсу та забезпечення доступності за допомогою голосових команд.

Локалізація полягає в адаптації додатку до різних мов та культурних особливостей. Це включає переклад тексту інтерфейсу, підтримку різних форматів дат, часу та чисел, а також врахування специфічних для кожної культури особливостей, таких як напрямок читання та порядок сортування. Android Studio надає інструменти для локалізації, що полегшують процес створення багатомовних додатків.

Дослідження методів розробки мобільних додатків для вивчення англійської мови виявило необхідність ретельного підходу до вибору платформи, середовища розробки, архітектури, технологій та інструментів.

Для додатку з вивчення англійської мови важливо забезпечити високу якість локалізації, особливо для тексту вправ, інструкцій та пояснень. Це може включати залучення професійних перекладачів або носіїв мови для перевірки та редагування перекладених матеріалів.

На основі проведеного дослідження можна сформулювати основні задачі для реалізації у даній кваліфікаційній роботі:

1. Проаналізувати вимоги та визначити функціональні можливості мобільного додатку для вивчення англійської мови.
2. Спроектувати архітектуру та структуру додатку, обравши відповідні шаблони проектування та архітектурні рішення.
3. Розробити інтерфейс користувача з урахуванням принципів зручності, доступності та локалізації.

4. Реалізувати основні функції додатку, такі як навчальні вправи, тести, робота з аудіо та відео.

5. Забезпечити взаємодію додатку з хмарними сервісами для аутентифікації користувачів, зберігання даних та доступу до онлайн-ресурсів.

6. Провести тестування додатку на різних рівнях та забезпечити його належну продуктивність та оптимізацію використання ресурсів пристрою.

7. Розробити документацію, керівництво користувача та інструкції для забезпечення зручного використання додатку.

8. Провести апробацію та впровадження розробленого додатку для отримання зворотного зв'язку від користувачів та вдосконалення його функціоналу.

Вибір оптимального методу розробки мобільного додатку для вивчення англійської мови вимагає ретельного аналізу всіх факторів та вимог проекту. У багатьох випадках може бути доцільним використання комбінації різних методів або гібридних підходів, які поєднують переваги кількох моделей. Наприклад, проект може розпочатися з використання каскадної моделі для ретельного планування та визначення вимог, а потім перейти до ітеративної моделі з проміжним плануванням для подальшої розробки та інтеграції зворотного зв'язку від користувачів.

Гнучкі методології, такі як Scrum, стали дуже популярними в розробці програмного забезпечення, особливо для мобільних пристроїв. Їх перевагами є швидка адаптація до змін вимог, регулярні релізи нових версій, тісна співпраця між командою розробників та замовниками, а також гнучкість і можливість швидкого реагування на непередбачувані ситуації. Однак, використання гнучких методологій вимагає постійного залучення замовника або представника користувачів, що може бути складним у випадку освітніх проектів. Крім того, на початковому етапі важко точно оцінити вартість проекту, оскільки вимоги можуть змінюватися.

Каскадна модель, навпаки, забезпечує чітку структуру процесу розробки та дозволяє уникнути невизначеностей і непорозумінь. Її перевагами є ретельне

планування на початковому етапі, добре структурований процес розробки та простота у відстеженні прогресу. Однак, ця модель має низьку гнучкість і складність у внесенні змін на пізніх стадіях розробки, а також відсутність зворотного зв'язку від користувачів до завершення проекту.

Ітеративна модель з проміжним плануванням є гібридним підходом, який поєднує переваги ітеративного розвитку з чітким проектуванням на кожній ітерації. Це забезпечує гнучкість та можливість внесення змін у вимоги на основі зворотного зв'язку від користувачів, регулярні проміжні релізи додатку та поступове розширення функціоналу. Водночас, така модель вимагає підвищених витрат на проектування та високої кваліфікації розробників.

Спіральна модель розробки програмного забезпечення є ітераційною моделлю, яка поєднує елементи гнучкої та планової розробки з акцентом на управління ризиками. Її перевагами є гнучкість, можливість залучення користувачів на різних етапах розробки та регулярні проміжні релізи додатку. Однак, ця модель вимагає підвищених витрат на управління проектом, високої кваліфікації розробників та досвіду в управлінні ризиками.

Незалежно від обраного методу, успіх розробки мобільного додатку для вивчення англійської мови значною мірою залежить від ефективної співпраці всіх учасників процесу: замовників, розробників, експертів з англійської мови та освітніх технологій, а також кінцевих користувачів. Регулярне спілкування, обмін інформацією та відкритість до зворотного зв'язку є ключовими факторами для створення якісного та корисного продукту, який відповідатиме потребам користувачів та сприятиме ефективному вивченню англійської мови.

При виборі методу розробки мобільного додатку для вивчення англійської мови необхідно враховувати кілька ключових факторів:

1. Масштаб та складність проекту: для більш масштабних і складних проектів краще підходять ітераційні методи, такі як гнучкі методології або спіральна модель, які забезпечують необхідну гнучкість та можливість управління ризиками.

2. Вимоги до якості та відповідність освітнім стандартам: каскадна модель може бути більш підходящою, якщо проект вимагає дотримання жорстких стандартів та норм, характерних для освітнього програмного забезпечення.

3. Наявні ресурси та досвід команди розробників: для успішного впровадження гнучких методологій або спіральної моделі потрібні висококваліфіковані розробники з відповідним досвідом роботи в ітераційному режимі.

4. Терміни та бюджет проекту: ітераційні методи, як правило, вимагають більших витрат на управління проектом, тоді як каскадна модель може бути більш економічно ефективною для проектів із чітко визначеними вимогами.

5. Ступінь залучення замовників та користувачів: якщо передбачається тісна співпраця з замовниками та регулярне отримання зворотного зв'язку від користувачів, перевагу варто віддати гнучким методологіям або ітераційним моделям з проміжним плануванням.

Загалом, розробка якісного мобільного додатку для вивчення англійської мови вимагає ретельного вибору та інтеграції різних методів, технологій та інструментів. Правильний підхід до архітектури, тестування, оптимізації, доступності та локалізації дозволить створити додаток, який буде корисним, ефективним та зручним для широкого кола користувачів.

1.3. Висновки до першого розділу

У першому розділі було проведено аналіз та дослідження предметної області, а саме розробки мобільного додатку для вивчення англійської мови. Були розглянуті існуючі публікації, методи та підходи до створення таких додатків.

Аналіз публікацій показав зростаючу актуальність та попит на мобільні додатки для вивчення іноземних мов, зокрема англійської. Автори наголошують на перевагах використання мобільних технологій у навчанні, таких як

портативність, зручність, доступність, індивідуалізація та підвищення мотивації студентів.

Було рекомендовано використовувати платформу Android та середовище розробки Android Studio, а також мову програмування Kotlin, яка забезпечує більшу ефективність, безпеку та сучасні парадигми програмування порівняно з Java. У ході аналізу було розглянуто кілька основних методів, серед яких гнучкі методології (Agile), каскадна модель (Waterfall), ітеративна модель з проміжним плануванням (Iterative with Intermediate Design) та спіральна модель (Spiral).

Для забезпечення якісної архітектури додатку пропонується використовувати сучасні архітектурні шаблони, такі як MVVM або MVP, а також архітектурні компоненти Android, такі як Room, LiveData, ViewModel та Dagger. Для взаємодії з мережею та хмарними сервісами рекомендується використовувати бібліотеки, такі як Retrofit або Firebase,

Для роботи з мультимедійним контентом запропоновано використовувати API MediaPlayer, ExoPlayer, Text-to-Speech від Google та інші інструменти.

Обраний метод розробки мобільного додатку для вивчення англійської мови відіграватиме ключову роль у забезпеченні успішної реалізації проекту, врахуванні вимог користувачів та відповідності освітнім стандартам. Ретельний аналіз факторів та ефективне поєднання різних методологій дозволить створити якісний та корисний продукт, який сприятиме підвищенню доступності та ефективності вивчення англійської мови за допомогою сучасних мобільних технологій.

РОЗДІЛ 2
АНАЛІЗ ЗАСОБІВ РЕАЛІЗАЦІЇ МОБІЛЬНОГО ДОДАТКУ

2.1. Вибір програмних засобів для реалізації проекту

На сучасному ринку мобільних пристроїв домінують дві основні операційні системи: Android від Google та iOS від Apple. Обидві платформи мають свої переваги та недоліки, тому вибір між ними є важливим рішенням для розробників мобільних додатків.

Android - це відкрита мобільна операційна система, заснована на ядрі Linux, яка була розроблена Google. Вона характеризується відкритим вихідним кодом, що дозволяє виробникам пристроїв вносити модифікації та пристосовувати її до своїх потреб. Android надає велику гнучкість у налаштуванні та персоналізації, а також пропонує широкий вибір пристроїв від різних виробників з різними цінами та характеристиками. Це зробило Android найпопулярнішою мобільною операційною системою у світі, займаючи близько 71% світового ринку смартфонів станом на 2022 рік.

З іншого боку, iOS є закритою операційною системою, розробленою виключно для пристроїв Apple, таких як iPhone, iPad та iPod Touch. На відміну від Android, iOS має більш жорстку інтеграцію між апаратним та програмним забезпеченням, що забезпечує вищу продуктивність та оптимізацію. iOS також відома своїм зручним та інтуїтивним користувацьким інтерфейсом, а також високим рівнем безпеки та конфіденційності. Однак iOS доступна лише на пристроях Apple, що обмежує її доступність та гнучкість порівняно з Android.

Для реалізації мобільного додатку для вивчення англійської мови в рамках цієї кваліфікаційної роботи було обрано платформу Android з кількох причин:

1. Широка поширеність: Android є найпопулярнішою мобільною операційною системою у світі, що забезпечує доступ до широкої аудиторії користувачів. Оскільки додаток для вивчення англійської мови орієнтований на

масового споживача, Android є найбільш доцільним вибором з точки зору охоплення цільової аудиторії.

2. Відкритий вихідний код: Android має відкритий вихідний код, що робить її більш доступною для розробників та дозволяє адаптувати її під специфічні потреби проекту. Це також сприяє кращій інтеграції з різними бібліотеками та фреймворками з відкритим вихідним кодом.

3. Низька вартість розробки: Розробка для Android зазвичай є більш економічно вигідною порівняно з iOS, оскільки не потребує дорогих ліцензій та спеціального обладнання від Apple. Це особливо важливо для проектів з обмеженим бюджетом, таких як кваліфікаційна робота.

4. Гнучкість та налаштування: Android дозволяє користувачам персоналізувати свої пристрої відповідно до їхніх уподобань та потреб. Це може бути корисним для додатку з вивчення англійської мови, оскільки користувачі можуть налаштувати додаток відповідно до свого рівня знань, інтересів та стилю навчання.

5. Різноманітність пристроїв: Android підтримується величезною кількістю пристроїв різних виробників, з різними розмірами екранів, характеристиками та цінами. Це дозволяє охопити ширшу аудиторію користувачів з різними потребами та бюджетами.

6. Сумісність з хмарними сервісами: Android тісно інтегрований з хмарними сервісами Google, такими як Firebase, Google Play Services та Google Cloud Platform. Ці сервіси пропонують потужні інструменти для розробки, аналітики, хмарного зберігання даних та інших функцій, які можуть бути корисними для реалізації додатку для вивчення англійської мови.

Традиційно для розробки додатків на Android використовувалася мова програмування Java. Проте в останні роки компанія Google офіційно підтримала альтернативну мову Kotlin, яка була розроблена компанією JetBrains.

Java - це широко поширена та перевірена часом мова програмування, яка використовується в різних галузях, включаючи розробку мобільних додатків, веб-застосунків, настільних програм та корпоративних систем. Вона відома

своєю стабільністю, безпекою та великою екосистемою бібліотек і фреймворків. Однак Java часто критикується за громіздкий та застарілий синтаксис, а також відсутність деяких сучасних можливостей, таких як лямбда-вирази та підтримка функціонального програмування.

Kotlin, з іншого боку, є відносно новою мовою програмування, яка була розроблена JetBrains як альтернатива Java для Android-розробки. Вона була створена з метою підвищення продуктивності розробників, зменшення кількості шаблонного коду та усунення деяких недоліків Java. Kotlin є більш лаконічною, безпечною та сучасною мовою, яка підтримує функціональне програмування, розширення функціоналу та інші передові концепції.

Незважаючи на те, що Java залишається широко використовуваною та підтримуваною мовою для Android-розробки, Kotlin все частіше стає кращим вибором для нових проектів з кількох причин:

1. Підвищена продуктивність: Kotlin вимагає менше коду для виконання тих самих завдань, що і Java, завдяки своєму лаконічному та виразному синтаксису. Це підвищує продуктивність розробників та полегшує підтримку коду.

2. Безпека: Kotlin має вбудовані механізми безпеки, такі як нульова безпека, що зменшує ймовірність виникнення помилок, пов'язаних з обробкою null-значень.

3. Сумісність з Java: Kotlin повністю сумісний з Java, що дозволяє використовувати існуючі Java-бібліотеки та фреймворки в нових проектах на Kotlin, а також інтегрувати Kotlin-код в існуючі Java-проекти.

4. Функціональне програмування: Kotlin підтримує концепції функціонального програмування, такі як лямбда-вирази, функції вищого порядку та розширення функціоналу, що робить код більш чистим та лаконічним.

5. Сучасний синтаксис: Kotlin має сучасний та зрозумілий синтаксис, який запозичив кращі практики з інших сучасних мов програмування, таких як Scala та Swift. Це робить код більш читабельним та легшим для розуміння.

6. Підтримка Google: Google офіційно підтримує Kotlin як мову для розробки додатків на Android, що свідчить про її перспективність та довгострокову підтримку в екосистемі Android.

7. Велика спільнота: Незважаючи на те, що Kotlin є відносно новою мовою, вона швидко набуває популярності серед розробників Android, утворюючи велику та активну спільноту, яка сприяє розвитку бібліотек, інструментів та навчальних ресурсів.

Для розробки додатків на платформі Android найбільш поширеним та рекомендованим інструментом є Android Studio - офіційна інтегрована середа розробки (IDE) від Google. Android Studio була спеціально розроблена для створення високоякісних додатків на Android, забезпечуючи зручне та інтуїтивне середовище для написання коду, налагодження, тестування та розгортання.

Android Studio пропонує широкий спектр потужних інструментів та функцій для полегшення процесу розробки додатків на Android, включаючи:

1. Інтелектуальний редактор коду: Редактор коду Android Studio забезпечує підсвічування синтаксису, автозавершення коду, перевірку помилок під час введення та інші функції, які підвищують продуктивність розробників.

2. Інструменти для налагодження: Android Studio включає потужний інструмент для налагодження (Debugger), який дозволяє відстежувати та аналізувати виконання додатку, встановлювати точки зупинки, переглядати значення змінних та виконувати крок за кроком.

3. Емулятор Android: Вбудований емулятор Android дозволяє запускати та тестувати додатки на віртуальному пристрої з різними конфігураціями та версіями Android, без необхідності використання фізичного пристрою.

4. Інструменти профілювання: Android Studio пропонує інструменти для профілювання продуктивності додатку, включаючи профілювання CPU, пам'яті, мережевого трафіку та енергоспоживання, що допомагає виявляти та усувати вузькі місця.

5. Підтримка різних мов програмування: Android Studio підтримує кілька мов програмування, таких як Java, Kotlin, C++ та інші, забезпечуючи гнучкість у виборі мови для розробки.

6. Інтеграція з системами контролю версій: Android Studio тісно інтегрований з популярними системами контролю версій, такими як Git, що полегшує спільну роботу над проектами та відстеження змін у коді.

7. Можливості побудови та розгортання: Android Studio забезпечує зручні інструменти для збірки та підписування додатків, а також розгортання їх на пристрої або в магазини додатків, такі як Google Play.

Завдяки своїм потужним можливостям, зручному інтерфейсу та тісній інтеграції з екосистемою Android, Android Studio є ідеальним вибором середовища розробки для створення мобільного додатку для вивчення англійської мови в рамках даної кваліфікаційної роботи.

Екосистема Android пропонує величезну кількість бібліотек та фреймворків, які можуть значно полегшити та прискорити процес розробки додатків. Ці інструменти забезпечують готові рішення для різноманітних завдань, таких як розробка інтерфейсу користувача, робота з мережею, обробка мультимедіа, забезпечення безпеки та багато іншого. Використання належних бібліотек та фреймворків дозволяє уникнути написання надлишкового коду та зосередитися на основній логіці додатку.

Для розробки сучасного та зручного інтерфейсу користувача для додатку з вивчення англійської мови можна розглянути такі бібліотеки:

1. Material Design: Офіційні компоненти та бібліотеки від Google, які реалізують керівні принципи Material Design - сучасної концепції дизайну, орієнтованої на створення інтуїтивних та естетичних інтерфейсів.

2. ConstraintLayout: Потужний та гнучкий макет, що спрощує процес розміщення елементів інтерфейсу та забезпечує відповідне масштабування на різних екранах і орієнтаціях.

3. RecyclerView: Висококонфігуруємий віджет списку, який забезпечує ефективне відображення та прокручування великих наборів даних, таких як списки слів або фраз.

Для інтеграції додатку з хмарними сервісами, такими як Firebase, та взаємодії з мережевими ресурсами можна використовувати такі бібліотеки:

1. Retrofit: Потужна бібліотека для взаємодії з REST API, яка забезпечує зручний інтерфейс та автоматичне перетворення даних між об'єктами Java/Kotlin та форматами даних, такими як JSON.

2. Firebase SDK: Офіційний набір бібліотек від Google для інтеграції з різними сервісами Firebase, такими як аутентифікація користувачів, хмарна база даних Realtime Database, хмарне сховище файлів Cloud Storage та інші.

Під час розробки мобільного додатку для вивчення англійської мови важливо ретельно вибирати бібліотеки та фреймворки, враховуючи їхню стабільність, продуктивність, популярність у спільноті розробників та відповідність вимогам проекту. Правильний вибір допоміжних інструментів може значно полегшити процес розробки, підвищити якість коду та прискорити виведення продукту на ринок.

Firebase є потужною хмарною платформою, розробленою Google, яка пропонує комплексний набір інструментів та сервісів для полегшення розробки високоякісних мобільних додатків. Ця платформа забезпечує готові рішення для різноманітних завдань, таких як аутентифікація користувачів, хмарна база даних, аналітика, хмарні повідомлення та багато іншого.

Використання Firebase може значно спростити процес розробки мобільного додатку для вивчення англійської мови, зменшуючи навантаження на розробників та дозволяючи їм зосередитися на основній логіці додатку, а не на реалізації допоміжної інфраструктури.

Firebase пропонує широкий спектр сервісів та інструментів, серед яких:

1. Authentication: Сервіс аутентифікації Firebase дозволяє легко інтегрувати різні методи аутентифікації користувачів, такі як електронна

пошта/пароль, соціальні мережі (Google, Facebook, Twitter), анонімна аутентифікація та інші.

2. Realtime Database: Хмарна NoSQL база даних Firebase Realtime Database забезпечує автоматичну синхронізацію даних у реальному часі між додатками та хмарою, що полегшує обмін даними між кількома клієнтами.

3. Cloud Firestore: Масштабована та потужна хмарна NoSQL база даних, оптимізована для зберігання та синхронізації структурованих даних на різних платформах.

4. Cloud Storage: Сервіс для зберігання та доставки контенту, такого як зображення, відео, аудіо та інші файли, який забезпечує безпечне зберігання та масштабованість.

5. Cloud Functions: Платформа для розгортання серверного коду, яка дозволяє запускати функції на основі подій або за розкладом, що корисно для виконання фонових завдань або реагування на події з бази даних.

6. Hosting: Сервіс для розгортання та хостингу веб-сайтів, статичних ресурсів та мікросервісів на безпечній та масштабованій інфраструктурі Google.

7. Аналітика: Потужні інструменти для відстеження подій, поведінки користувачів, відгуків та проблем у додатках, що допомагає покращити якість продукту та досвід користувачів.

Використання Firebase в додатку для вивчення англійської мови може включати використання аутентифікації для забезпечення безпечного доступу користувачів, Realtime Database або Cloud Firestore для зберігання даних користувачів та їхнього прогресу, Cloud Storage для зберігання мультимедійних файлів (аудіо, відео), Cloud Functions для виконання фонових завдань та аналітику для відстеження використання додатку та виявлення можливих проблем.

Слід зазначити, що Firebase пропонує детальну документацію, навчальні матеріали та приклади коду, що полегшує інтеграцію цих сервісів у додаток для Android.

2.2. Засоби для реалізації мобільного додатку

Під час розробки мобільного додатку для вивчення англійської мови важливо обрати належні бібліотеки та фреймворки, які полегшать та прискорять процес розробки, а також забезпечать високу якість та продуктивність кінцевого продукту. Використання готових рішень дозволяє уникнути написання надлишкового коду та зосередитися на основній логіці додатку.

Для реалізації мобільного додатку для вивчення англійської мови буде застосований комплексний підхід, який включає використання різноманітних технологій, фреймворків та бібліотек. Основна ціль – створити зручний, інтерактивний та ефективний додаток, який буде адаптований до потреб користувачів та забезпечить їм належний рівень навчання англійської мови.

Розробка буде здійснюватися на платформі Android, що дозволить максимально охопити цільову аудиторію користувачів. Для створення додатку буде використане інтегроване середовище розробки Android Studio, яке є офіційним інструментом від Google та забезпечує потужну підтримку для розробки додатків на Android. Android Studio пропонує зручний інтерфейс, інтелектуальний редактор коду, емулятор для тестування додатків, інструменти для налагодження та профілювання, а також інтеграцію з системами контролю версій та можливості для розгортання додатків.

Мовою програмування для реалізації додатку буде обрана Kotlin – сучасна, безпечна та лаконічна мова, яка офіційно підтримується Google для розробки додатків на Android. Kotlin повністю сумісна з Java та пропонує багато переваг, таких як функціональне програмування, нульова безпека, розширення функціоналу та підвищену продуктивність розробників. Використання Kotlin дозволить написати більш читабельний, лаконічний та підтримуваний код, що спростить процес розробки та подальшу підтримку додатку.

Для створення сучасного та зручного інтерфейсу користувача в додатку для вивчення англійської мови можна використати такі бібліотеки:

1. Material Design: Офіційні компоненти та бібліотеки від Google, що реалізують керівні принципи Material Design - сучасної концепції дизайну, орієнтованої на створення інтуїтивних та естетичних інтерфейсів. Material Design забезпечує єдиний стиль та стандартний набір віджетів для створення привабливих та зрозумілих інтерфейсів.

2. ConstraintLayout: Потужний та гнучкий макет, що спрощує процес розміщення елементів інтерфейсу та забезпечує відповідне масштабування на різних екранах і орієнтаціях. ConstraintLayout дозволяє описувати розміщення віджетів за допомогою системи обмежень, що робить його зручним для створення складних макетів.

3. RecyclerView: Висококонфігуруємий віджет списку, який забезпечує ефективне відображення та прокручування великих наборів даних, таких як списки слів або фраз. RecyclerView оптимізований для роботи з великими колекціями даних, забезпечуючи плавну прокрутку та ефективне використання пам'яті [8].

Для взаємодії з мережевими ресурсами та інтеграції з хмарними сервісами можна використовувати такі бібліотеки:

1. Retrofit: Потужна бібліотека для взаємодії з REST API, яка забезпечує зручний інтерфейс та автоматичне перетворення даних між об'єктами Java/Kotlin та форматами даних, такими як JSON. Retrofit спрощує роботу з мережевими запитами, абстрагуючи низькорівневі деталі та дозволяючи зосередитися на логіці додатку [9].

2. Firebase SDK: Офіційний набір бібліотек від Google для інтеграції з різними сервісами Firebase, такими як аутентифікація користувачів, хмарна база даних Realtime Database, хмарне сховище файлів Cloud Storage та інші. Firebase SDK забезпечує зручний інтерфейс для взаємодії з хмарними сервісами Firebase, дозволяючи швидко інтегрувати їх у додаток.

Firebase є потужною хмарною платформою, розробленою Google, яка пропонує комплексний набір інструментів та сервісів для полегшення розробки високоякісних мобільних додатків. Ця платформа забезпечує готові рішення для

різноманітних завдань, таких як аутентифікація користувачів, хмарна база даних, аналітика, хмарні повідомлення та багато іншого [10].

Використання Firebase може значно спростити процес розробки мобільного додатку для вивчення англійської мови, зменшуючи навантаження на розробників та дозволяючи їм зосередитися на основній логіці додатку, а не на реалізації допоміжної інфраструктури.

Firebase пропонує широкий спектр сервісів та інструментів, серед яких:

1. Authentication: Сервіс аутентифікації Firebase дозволяє легко інтегрувати різні методи аутентифікації користувачів, такі як електронна пошта/пароль, соціальні мережі (Google, Facebook, Twitter), анонімна аутентифікація та інші.

2. Realtime Database: Хмарна NoSQL база даних Firebase Realtime Database забезпечує автоматичну синхронізацію даних у реальному часі між додатками та хмарою, що полегшує обмін даними між кількома клієнтами. Це може бути корисним для зберігання прогресу користувачів у вивченні англійської мови та синхронізації даних між різними пристроями.

3. Cloud Firestore: Масштабована та потужна хмарна NoSQL база даних, оптимізована для зберігання та синхронізації структурованих даних на різних платформах. Cloud Firestore може бути використана для зберігання більш складних даних, таких як словники, граматичні правила або навчальні матеріали.

4. Cloud Storage: Сервіс для зберігання та доставки контенту, такого як зображення, відео, аудіо та інші файли, який забезпечує безпечне зберігання та масштабованість. Для додатку з вивчення англійської мови це може бути корисним для зберігання аудіо- та відеоматеріалів.

5. Cloud Functions: Платформа для розгортання серверного коду, яка дозволяє запускати функції на основі подій або за розкладом, що корисно для виконання фонових завдань або реагування на події з бази даних.

Додатково буде інтегровано аналітику Firebase для відстеження подій, поведінки користувачів, відгуків та можливих проблем у додатку. Ця аналітика допоможе покращити якість продукту, виявити слабкі місця та зрозуміти, як

користувачі взаємодіють з додатком, що дозволить вносити необхідні зміни та покращення.

Для забезпечення зручного та ефективного процесу вивчення англійської мови додаток буде інтегруватися з одним або кількома зовнішніми API, такими як API словників або перекладачів. Це дозволить користувачам отримувати визначення слів, переклади фраз або контекстні підказки безпосередньо в додатку, не виходячи з нього.

Під час розробки буде приділено особливу увагу тестуванню та забезпеченню належної якості додатку. Будуть використані різні види тестування, такі як модульне тестування, інтеграційне тестування та тестування користувацького інтерфейсу, щоб виявляти та усувати потенційні помилки та недоліки.

Використання Firebase в додатку для вивчення англійської мови може включати використання аутентифікації для забезпечення безпечного доступу користувачів, Realtime Database або Cloud Firestore для зберігання даних користувачів та їхнього прогресу, Cloud Storage для зберігання мультимедійних файлів (аудіо, відео), Cloud Functions для виконання фонових завдань та аналітику для відстеження використання додатку та виявлення можливих проблем.

Слід зазначити, що Firebase пропонує детальну документацію, навчальні матеріали та приклади коду, що полегшує інтеграцію цих сервісів у додаток для Android [11].

Належний вибір бібліотек та фреймворків є ключовим чинником для успішної розробки мобільних додатків. Використання готових рішень, таких як ті, що пропонує Firebase, дозволяє заощадити час та зусилля, підвищити якість коду та прискорити виведення продукту на ринок. Проте важливо ретельно оцінювати потреби проекту та обирати лише ті інструменти, які дійсно необхідні та відповідають вимогам. Надмірне використання сторонніх бібліотек може призвести до зростання розміру додатку, збільшення залежностей та ускладнення процесу розробки та підтримки.

Крім того, під час вибору бібліотек та фреймворків слід враховувати їхню стабільність, продуктивність, популярність у спільноті розробників та регулярність оновлень. Використання застарілих або погано підтримуваних інструментів може призвести до проблем з сумісністю, безпекою та продуктивністю в майбутньому.

Для реалізації мобільного додатку для вивчення англійської мови необхідно також обрати відповідне інтегроване середовище розробки (IDE). Android Studio є офіційною та рекомендованою IDE від Google для розробки додатків на Android. Вона була спеціально розроблена для створення високоякісних додатків на Android, забезпечуючи зручне та інтуїтивне середовище для написання коду, налагодження, тестування та розгортання.

Android Studio пропонує широкий спектр потужних інструментів та функцій для полегшення процесу розробки додатків на Android, включаючи:

1. Інтелектуальний редактор коду з підсвічуванням синтаксису, автозавершенням коду та перевіркою помилок.
2. Потужний інструмент для налагодження (Debugger) з можливістю встановлення точок зупинки, перегляду значень змінних та покрокового виконання коду.
3. Вбудований емулятор Android для запуску та тестування додатків на віртуальному пристрої.
4. Інструменти профілювання продуктивності додатку, включаючи профілювання CPU, пам'яті, мережевого трафіку та енергоспоживання.
5. Підтримка різних мов програмування, таких як Java, Kotlin, C++ та інші.
6. Інтеграція з системами контролю версій, такими як Git.
7. Можливості побудови та розгортання додатків, включаючи підписування додатків та розгортання їх на пристрої або в магазини додатків [12].

Завдяки своїм потужним можливостям, зручному інтерфейсу та тісній інтеграції з екосистемою Android, Android Studio є ідеальним вибором

середовища розробки для створення мобільного додатку для вивчення англійської мови в рамках даної кваліфікаційної роботи.

Вибір належних інструментів та бібліотек для розробки мобільного додатку для вивчення англійської мови є важливим рішенням, яке може значно вплинути на якість, продуктивність та час розробки. Використання сучасних та перевірених бібліотек, таких як Material Design, ConstraintLayout, RecyclerView, Retrofit та Firebase SDK, забезпечить зручний та інтуїтивний інтерфейс користувача, ефективну взаємодію з мережевими ресурсами та хмарними сервісами, а також полегшить процес розробки та підтримки додатку.

Загалом, для реалізації мобільного додатку для вивчення англійської мови буде застосовано комплексний підхід, що поєднує сучасні технології, фреймворки та бібліотеки для розробки, зберігання даних, взаємодії з мережевими ресурсами та аналітики. Це забезпечить створення якісного, ефективного та зручного додатку, який буде адаптований до потреб користувачів

2.3. Висновки до другого розділу

У другому розділі були детально розглянуті та обґрунтовані вибір програмних засобів, бібліотек, фреймворків та інтегрованого середовища розробки для реалізації мобільного додатку для вивчення англійської мови. Після ретельного аналізу та оцінки різних альтернатив були зроблені наступні ключові вибори:

1. Платформа Android як цільова операційна система для розробки, з огляду на її високу поширеність, відкритий вихідний код, гнучкість і підтримку Google.

2. Мова програмування Kotlin, яка є сучасною, безпечною, лаконічною та повністю сумісною з Java, а також офіційно підтримується Google для Android-розробки.

3. Інтегроване середовище розробки Android Studio як офіційне та рекомендоване IDE для створення додатків на Android, що забезпечує зручне та потужне середовище для написання коду, налагодження та розгортання.

4. Для забезпечення швидкої та ефективної розробки, високої якості коду та сучасного зручного інтерфейсу були обрані такі провідні бібліотеки та фреймворки:

5. Material Design, ConstraintLayout та RecyclerView для створення привабливого та інтуїтивного інтерфейсу користувача, що відповідає сучасним стандартам та керівним принципам дизайну для мобільних платформ.

6. Retrofit для зручної та ефективної взаємодії з мережевими ресурсами та REST API, що дозволяє абстрагуватися від низькорівневих деталей та зосередитися на основній логіці додатку.

7. Firebase SDK - офіційний набір бібліотек від Google для інтеграції з різноманітними хмарними сервісами Firebase, такими як аутентифікація користувачів, хмарні бази даних, зберігання файлів, аналітика тощо.

Застосування обраних технологій, фреймворків та бібліотек дозволить створити додаток, який буде адаптований до потреб користувачів та відповідатиме новітнім тенденціям у галузі мобільної розробки.

Інтеграція з зовнішніми API, такими як API словників або перекладачів, забезпечить додатковий функціонал для користувачів, надаючи їм можливість отримувати визначення слів, переклади фраз чи контекстні підказки безпосередньо в додатку.

Використання різних видів тестування, включаючи модульне, інтеграційне та тестування користувацького інтерфейсу, гарантуватиме високу якість додатку, виявлення та усунення потенційних помилок та недоліків.

Загалом, обрані засоби для реалізації мобільного додатку для вивчення англійської мови є сучасними, потужними та перевіреними у галузі мобільної розробки. Їх комплексне застосування забезпечить створення якісного продукту, що відповідатиме вимогам та очікуванням користувачів.

РОЗДІЛ 3
РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ

3.1. Розробка загальної архітектури додатку в Android Studio

Мета створення мобільного додатку для вивчення англійської мови полягає в тому, щоб забезпечити користувачів зручним та ефективним інструментом для покращення своїх мовних навичок. Додаток повинен пропонувати різноманітні можливості для вивчення граматики, збагачення словникового запасу, вдосконалення навичок читання та прослуховування. Важливо також, щоб інтерфейс був інтуїтивно зрозумілим та привабливим для користувачів.

Для розробки мобільного додатку для вивчення англійської мови було обрано традиційну архітектуру для Android, що поєднує активності, фрагменти та спеціалізовані класи. Цей підхід добре підходить для даного проекту з кількох причин:

1. Простота та зрозумілість. Традиційна архітектура є добре відомою та широко прийнятою серед розробників Android, що полегшує розуміння та підтримку коду.
2. Відповідність принципам Android. Активності та фрагменти є невід'ємною частиною середовища розробки Android, і їх використання забезпечує сумісність із системними рекомендаціями та кращими практиками.
3. Модульність та масштабованість. Даний підхід дозволяє розділити функціональність додатку на окремі модулі (активності та фрагменти), що полегшує розширення та модифікацію коду в майбутньому.
4. Можливість повторного використання коду. Окремі компоненти, такі як адаптери, утиліти та моделі даних, можуть бути легко повторно використані в різних частинах додатку.

У рамках цієї архітектури будуть використані наступні ключові компоненти:

Активності (Activities): Головні контейнери для користувацького інтерфейсу та логіки додатку. Прикладами є MainActivity, LoginActivity, VocabularyActivity та GrammarActivity.

Фрагменти (Fragments): Повторно використовувані блоки інтерфейсу та логіки, які можна вбудовувати в активності. Наприклад, фрагменти для відображення списку слів або тестових запитань.

Адаптери (Adapters): Класи, що забезпечують зв'язок між даними та їх візуальним представленням у списках або іншими способами відображення. Наприклад, QuizListAdapter та MeaningAdapter.

Моделі даних (Data Models): Класи, що представляють структури даних, необхідні для функціонування додатку. Наприклад, QuizModel, QuestionModel та WordResult.

Допоміжні класи (Utility Classes): Класи, що надають загальні функції або сервіси, такі як робота з мережею (RetrofitInstance та DictionaryApi), обробка введення користувача (UiUtil) та інші.

Ця архітектура забезпечує чітке розділення обов'язків між компонентами, полегшуючи розуміння, тестування та підтримку коду. Такий підхід також дозволяє легко розширювати функціональність додатку в майбутньому за рахунок додавання нових активностей, фрагментів або модифікації існуючих компонентів.

MainActivity є головною активністю додатку (див. рис. 3.1) та виступає в ролі вхідної точки. Після запуску додатку, ця активність відображає головний екран з можливістю вибору різних режимів роботи, таких як граматика, словник та тестові питання. MainActivity також відповідає за навігацію до інших активностей, пов'язаних із вибраним режимом.

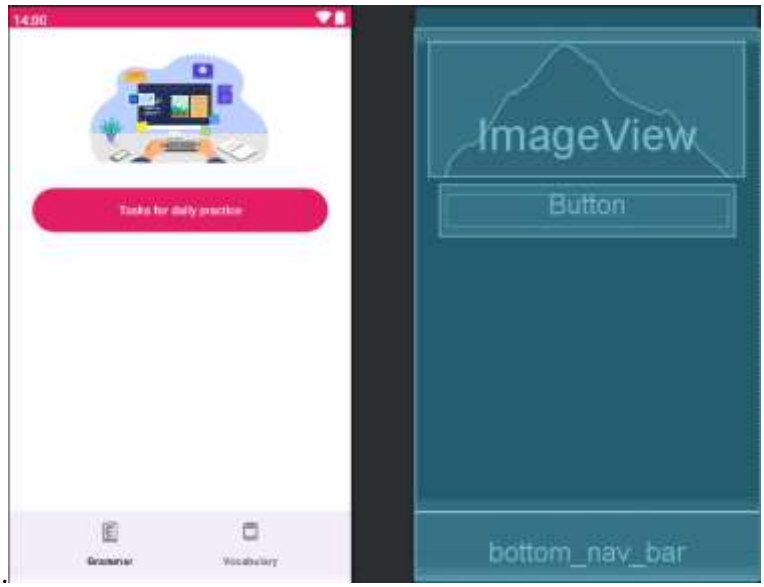


Рисунок 3.1 – Головна активність додатку MainActivity

GrammarActivity відповідає за відображення екрану з граматичними вправами та тестами (див. рис. 3.2). На цьому екрані користувачі можуть переглядати список доступних граматичних тем, вибирати відповідні вправи та проходити їх. Активність також може містити фрагменти для відображення деталей вправи або результатів тестування.

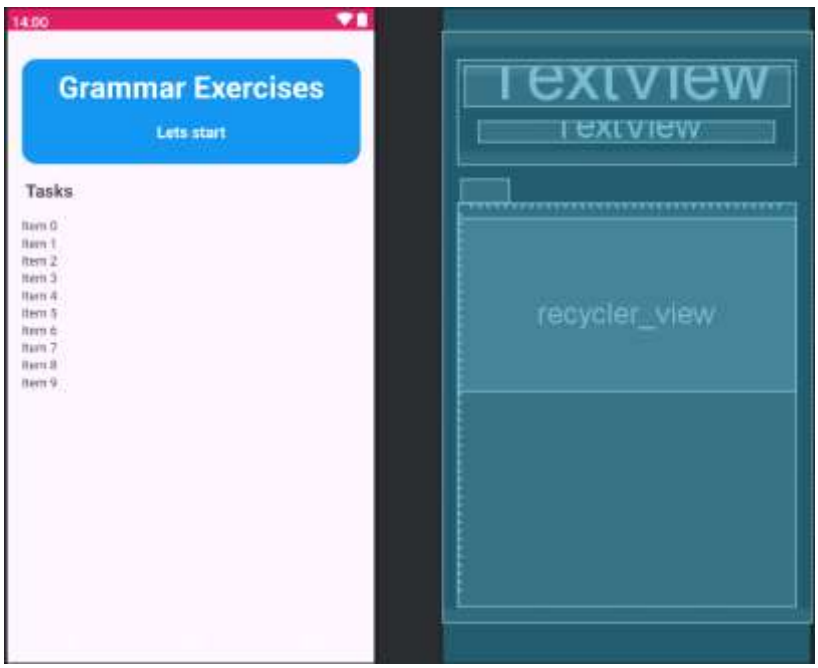


Рисунок 3.2 – Активність з граматичними вправами

VocabularyActivity надає користувачам доступ до словника (див. рис. 3.3). На цьому екрані користувачі можуть вводити слова для переглядати визначення, синоніми та антоніми. Активність може також містити фрагменти для відображення деталей перекладу або історії пошуку.

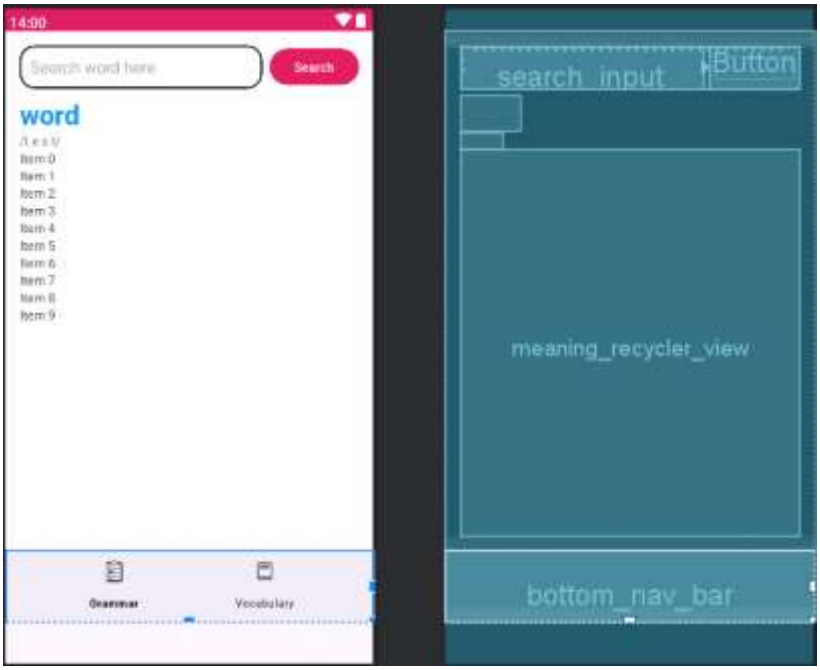


Рисунок 3.3 – Активність словника

QuizActivity відповідає за відображення екрану з тестовими питаннями та вправами (див. рис. 3.4). На цьому екрані користувачі можуть проходити різноманітні тести з граматики, словникового запасу або інших аспектів вивчення англійської мови. Активність може містити фрагменти для відображення питань, варіантів відповідей та результатів тестування.

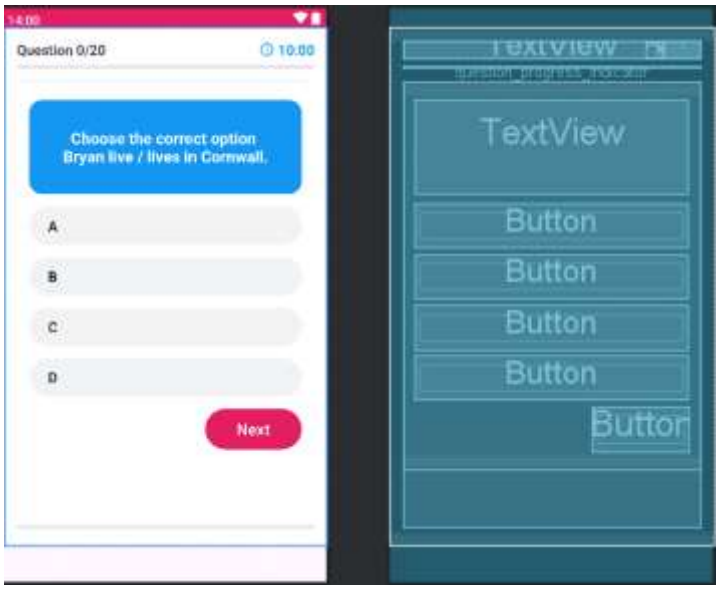


Рисунок 3.4 – Активність з тестовими питаннями

LoginActivity та SignupActivity відповідають за екран входу та реєстрації користувачів (див. рис. 3.5). Цей фрагмент може бути вбудований в різні активності, такі як MainActivity або окрема активність для авторизації. LoginFragment містить поля для введення даних користувача (електронна пошта та пароль), а також кнопки для входу або реєстрації.

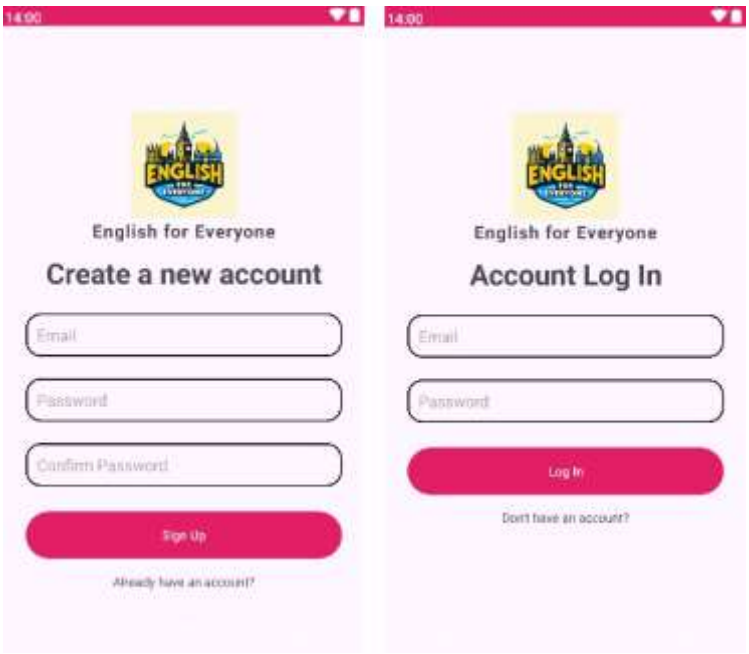


Рисунок 3.5 – Активності для реєстрації та входу у додаток

Ці компоненти забезпечують основну функціональність додатку та дозволяють користувачам вивчати англійську мову за допомогою різноманітних режимів та можливостей. Вони також можуть взаємодіяти з іншими допоміжними класами, такими як адаптери, моделі даних та утиліти, для забезпечення належного функціонування додатку.

Для забезпечення належної роботи додатку та зберігання даних про користувачів, вправи та словник, цей проект інтегрується з Firebase - потужною хмарною платформою від Google для розробки додатків.

Firebase Authentication є зручним та безпечним способом аутентифікації користувачів у додатку (див. рис. 3.6). LoginActivity та SignupActivity інтегровані з Firebase Authentication, що дозволяє користувачам реєструватися, входити в систему за допомогою електронної пошти та пароля, а також відновлювати облікові записи у разі втрати даних для входу.

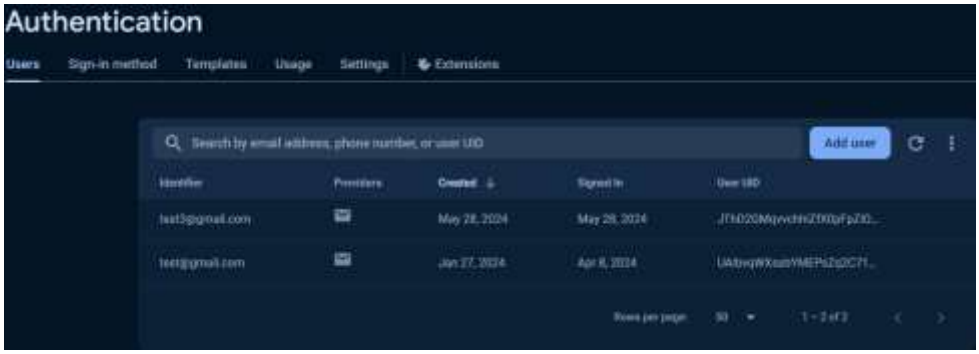


Рисунок 3.6 - Firebase Authentication для аутентифікації користувачів

Firebase Realtime Database - це хмарна NoSQL база даних, яка забезпечує надійне та масштабоване зберігання даних у реальному часі (див. рис. 3.7). У додатку ця база даних використовується для зберігання інформації про граматичні вправи, тестові запитання та словник. Активності GrammarActivity та VocabularyActivity взаємодіють з Realtime Database для отримання відповідних даних та їх відображення.

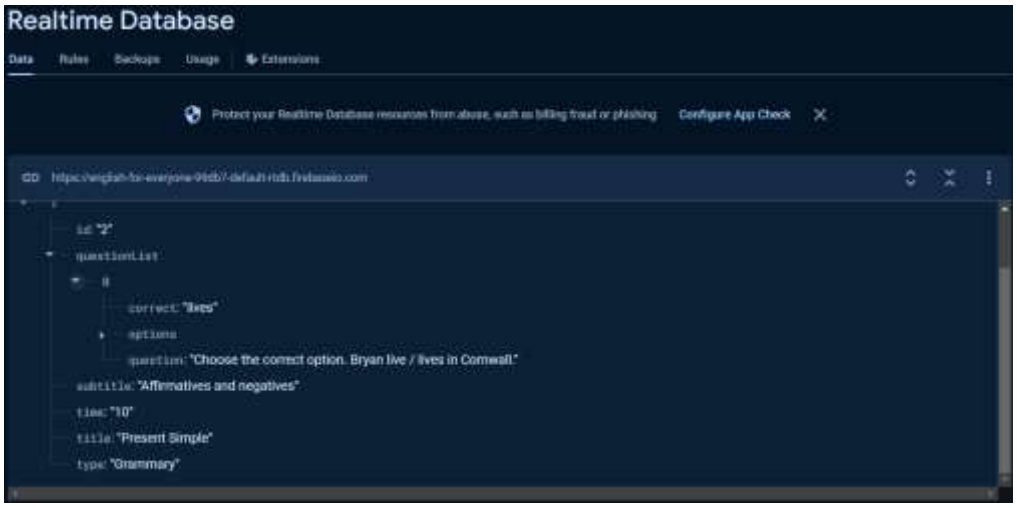


Рисунок 3.7 - Firebase Realtime Database для зберігання даних тестових запитань

Використання Firebase забезпечує низку переваг, таких як централізоване та безпечне зберігання даних, масштабованість, легку інтеграцію з Android-додатками, а також можливість швидкого розгортання та оновлення додатку без необхідності управління власною серверною інфраструктурою.

Інтеграція з Firebase дозволяє зосередитися на розробці основної функціональності додатку, водночас забезпечуючи надійність та безпеку даних користувачів і вмісту вправ та словника.

Retrofit є потужною бібліотекою від Square для взаємодії з REST API та виконання мережевих запитів. У даному додатку Retrofit використовується для отримання визначень слів із зовнішнього API словника.

Основними перевагами використання Retrofit є:

1. Простота та зрозумілість: Retrofit забезпечує чітку та інтуїтивно зрозумілу абстракцію для роботи з REST API, приховуючи складні деталі реалізації від розробника.

2. Типова безпека: Retrofit використовує типову безпеку завдяки впровадженню можливостей Java, гарантуючи відсутність нульових посилань у коді.

3. Зручність використання: Бібліотека пропонує зручні анотації для визначення HTTP-запитів, а також автоматично конвертує відповіді JSON у POJO-класи.

4. Асинхронність: Retrofit підтримує асинхронні запити, що запобігає блокуванню головного потоку та забезпечує плавний користувацький інтерфейс.

5. Інтеграція з RxJava: Retrofit легко інтегрується з RxJava, що дозволяє ефективно керувати асинхронними потоками даних та композицією запитів.

6. Вбудована підтримка різних типів даних: Retrofit підтримує широкий спектр типів даних для запитів та відповідей, включаючи об'єкти, масиви, списки та багато іншого.

7. Кешування та перезапити: Бібліотека дозволяє налаштовувати кешування відповідей та автоматично перезапускати невдалі запити.

У даному проекті класи RetrofitInstance та DictionaryApi інкапсулюють логіку взаємодії з API словника за допомогою Retrofit. RetrofitInstance налаштовує інстанс Retrofit з базовим URL та конвертером GSON для обробки JSON-відповідей. DictionaryApi визначає інтерфейс для взаємодії з API, використовуючи анотації Retrofit для опису HTTP-запитів та маршрутів.

Використання Retrofit у проекті значно спрощує процес отримання даних з віддаленого API словника, забезпечуючи ефективний, типово безпечний та зрозумілий спосіб роботи з мережевими запитами.

3.2. Тестування роботи системи

При запуску мобільного додатку «English for Everyone» (див. рис. 3.8) відкривається форма для реєстрації або входу в додаток, де потрібно ввести електронну пошту та пароль. Пароль повинен складатися щонайменше з 6 символів. Дані користувача записуються у Firebase Authentication, за допомогою якого можна змінити пароль або видалити обліковий запис.

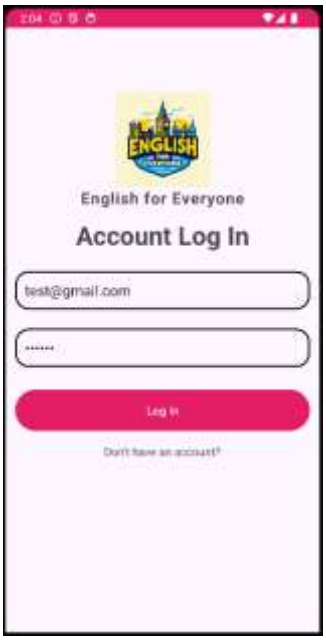


Рисунок 3.8 - Форма для реєстрації або входу

Якщо авторизація пройшла успішно, відкриється вікно з вибором вправ або словником (див. рис. 3.9). Для того, щоб перейти до вправ з граматики, потрібно натиснути на кнопку «Tasks for daily practice». Для переходу до словника потрібно натиснути на кнопку у панелі навігації під назвою «Vocabulary».

Рисунок 3.9 - Головне вікно додатку

При переході до Vocabulary відкриється екран Vocabulary Activity (див. рис. 3.10), на цьому екрані користувач побачить поле для введення слова, кнопку пошуку та порожній список для відображення результатів пошуку. Якщо користувач введе слово в поле та натисне кнопку пошуку, додаток надішле запит до зовнішнього API для отримання визначення та інформації про введене слово.

Після отримання відповіді від API, додаток відобразить на екрані введене слово, фонетичний запис слова, список значень слова. Для кожного значення буде відображено частину мови, визначення, список синонімів, список антонімів.

Якщо під час пошуку слова виникне помилка (наприклад, невідоме слово або проблеми з інтернет-з'єднанням), додаток відобразить відповідне повідомлення про помилку. Користувач зможе шукати інші слова, вводячи їх у текстове поле та натискаючи кнопку пошуку знову.

Рисунок 3.10 - Словник

При натисканні кнопки "Tasks for daily practice" додаток перейде на екран GrammarActivity (див. рис. 3.11). На цьому екрані відобразиться список доступних граматичних вправ, які завантажуються з бази даних Firebase. Кожна

граматична вправа буде представлена у вигляді окремого елементу в RecyclerView. Елемент вправи міститиме таку інформацію як назва вправи, короткий опис вправи, тривалість вправи (у хвилинах).

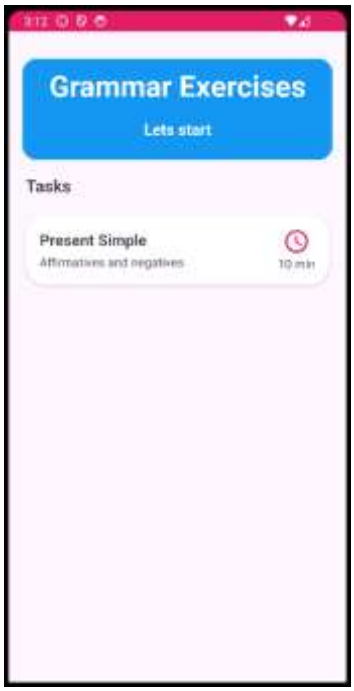


Рисунок 3.11 – Граматичні вправи

При натисканні на вправу, користувач буде перенаправлений на екран QuizActivity (див. рис. 3.12), де відбудеться завантаження питань цієї граматичної вправи. Користувач повинен вибрати одну з запропонованих відповідей, натиснувши на відповідну кнопку. Після вибору відповіді користувач може перейти до наступного питання, натиснувши кнопку «Next». Після завершення всіх питань вправи відобразиться діалогове вікно з підсумковим результатом користувача, включаючи загальну кількість правильних відповідей, відсоток успішності та рекомендацію щодо проходження/непроходження вправи. У діалоговому вікні буде кнопка «Finish», за допомогою якої можна повернутися до вибору вправ.

Рисунок 3.12 – Тести по граматиці

Загалом, додаток працює коректно і виконує основні функції, закладені в його дизайні. Проте, були виявлені деякі незначні проблеми, які необхідно усунути для забезпечення стабільної роботи системи. Основні функціональні можливості, такі як реєстрація та вхід користувачів, робота з граматичними вправами, взаємодія зі словником та проходження тестових питань, працюють належним чином.

3.3. Висновки до третього розділу

Розроблений мобільний додаток "English for Everyone" повністю виконує поставлену мету створення зручного та ефективного інструменту для вивчення англійської мови. Він успішно реалізує основні функціональні вимоги, такі як реєстрація та авторизація користувачів, доступ до граматичних вправ, функціональний словник та система тестових питань.

Успішна інтеграція з Firebase забезпечує надійну авторизацію через Firebase Authentication та зберігання даних про вправи та словник у Firebase

Realtime Database. Використання Retrofit для взаємодії з зовнішнім API словника спрощує отримання визначень слів та додаткової інформації, покращуючи цілісність додатку.

Розроблений додаток має низку переваг, включаючи зручний та інтуїтивно зрозумілий інтерфейс, різноманітність граматичних вправ для щоденної практики, функціональний словник з визначеннями, синонімами та антонімами, а також систему тестових питань для оцінки прогресу у вивченні мови.

Під час тестування роботи додатку було виявлено деякі незначні проблеми, такі як:

1. Іноді при переході між активностями з'являлися незначні затримки.
2. У рідкісних випадках при введенні слова у словнику відображалося застаріле значення.

Загалом, розроблений додаток є вдалою спробою створити корисний інструмент для вивчення англійської мови. Він поєднує в собі необхідну функціональність, зручний інтерфейс та сучасні технології розробки мобільних додатків.

У майбутньому планується продовжити вдосконалення додатку, покращуючи його продуктивність, додаючи нові види вправ та інтегруючи додаткові можливості, такі як голосовий переклад, генерація тестових питань на основі прогресу користувача та підтримка додаткових мов. Ці вдосконалення дозволять перетворити "English for Everyone" на ще потужніший та універсальний інструмент для вивчення мов.

ВИСНОВКИ

У результаті виконання дипломної роботи було розглянуто та розроблено мобільний додаток для вивчення англійської мови на платформі Android з використанням мови програмування Kotlin. Цей додаток спрямований на автоматизацію та покращення процесу вивчення іноземної мови, забезпечуючи інтерактивність та зручність для користувачів.

При створенні додатка враховувалися сучасні технології та вимоги до програмного забезпечення, такі як модульна структура, інтуїтивний та естетичний інтерфейс, підтримка різних рівнів складності, граматичні вправи, словник, відстеження прогресу, синхронізація даних у хмарі, офлайн режим, оптимізація для різних Android пристроїв, регулярні оновлення контенту та функціональності, а також дотримання найкращих практик, стандартів безпеки та конфіденційності даних користувачів.

Загальні результати роботи можна підсумувати наступним чином:

1. Проаналізовано методи та технічні засоби для реалізації мобільного додатка для вивчення англійської мови.
2. Розроблені вимоги до програмного забезпечення на основі аналізу потреб користувачів та переваг і недоліків існуючих систем.
3. Спроектовано та розроблено новий мобільний додаток з урахуванням сучасних технологій та користувацького досвіду.

Додаток складається з декількох основних компонентів, включаючи інтерфейс користувача, систему граматичних вправ та словника, механізми відстеження прогресу та синхронізації даних у хмарі. Інтерфейс реалізовано з використанням інструментів Android Studio та мови програмування Kotlin, що дозволило створити зручний та привабливий користувацький досвід.

Під час тестування додатка було підтверджено його відповідність очікуванням щодо зручності використання, ефективності навчання та стабільної роботи на різних пристроях Android. Отримані результати мають практичне

значення, оскільки сприяють автоматизації та покращенню процесу вивчення англійської мови для широкої аудиторії користувачів.

Отже, розроблений мобільний додаток для вивчення англійської мови є важливим кроком у покращенні якості та ефективності навчання іноземних мов. Цей додаток використовує передові технології та забезпечує надійну та зручну роботу, що сприяє підвищенню рівня знань користувачів та задоволенню їхніх потреб у вивченні англійської мови.

Потенційними напрямками для подальшого вдосконалення та розвитку додатку можуть бути:

1. Інтеграція з додатковими зовнішніми ресурсами та базами даних для збагачення навчального контенту.
2. Розширення функціоналу для підтримки вивчення інших іноземних мов.
3. Впровадження технологій штучного інтелекту та машинного навчання для персоналізації навчального процесу та адаптації до індивідуальних потреб користувачів.
4. Додавання можливостей для спільної роботи та взаємодії між користувачами, створюючи соціальну складову для вивчення мови.
5. Розробка версій додатку для інших платформ, таких як iOS або веб-версії, для забезпечення більшої доступності.

Загалом, розроблений мобільний додаток для вивчення англійської мови є прикладом успішного застосування сучасних технологій та методів для вирішення практичних завдань у галузі освіти та навчання. Його подальший розвиток та вдосконалення може зробити значний внесок у покращення якості та доступності ресурсів для вивчення іноземних мов, відкриваючи нові можливості для самовдосконалення та міжкультурної комунікації в сучасному глобалізованому світі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Рагульскіс М., Шуляков В.М., Шуляков І.М., Андросов Т.С. (2020). Розробка мобільного додатку для вивчення англійської мови. У збірнику "Комп'ютерні технології і мехатроніка". С. 236-237.

2. Блинова Н. М., Кирилова О. В., Долженко М. В. Дидактичний потенціал мобільних застосунків для вивчення англійської мови як іноземної. Вісник Університету імені Альфреда Нобеля. Серія «Педагогіка і психологія». Педагогічні науки. 2023. № 1 (25). С. 184-192.

3. Мороз Л., Ковалюк В., Масло І. Використання мобільних додатків у процесі вивчення англійської мови // Інноватика у вихованні. 2023. Вип. 17. С. 224-229.

4. Allen G. Android Studio 4.2 Development Essentials - Kotlin Edition. Payload Media, 2021.

5. David G., Josh S., Andrew B. Kotlin Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides) 2nd Edition, 2024.

6. Denis P., Loveth N. Tiny Android Projects Using Kotlin, 2024.

7. Neil S. Android Studio Iguana Essentials - Kotlin Edition: Developing Android Apps Using Android Studio 2023.2.1 and Kotlin, 2024.

8. Документація Android по RecyclerView [Електронний ресурс]. - Режим доступу: https://developer.android.com/guide/topics/ui/layout/recyclerview

9. Документація Retrofit [Електронний ресурс]. - Режим доступу: https://square.github.io/retrofit/

10. Офіційний веб-сайт Firebase [Електронний ресурс]. - Режим доступу: https://firebase.google.com

11. Firebase Documentation [Електронний ресурс]. - Режим доступу: https://firebase.google.com/docs

12. Офіційний веб-сайт Android Studio [Електронний ресурс]. - Режим доступу: https://developer.android.com/studio

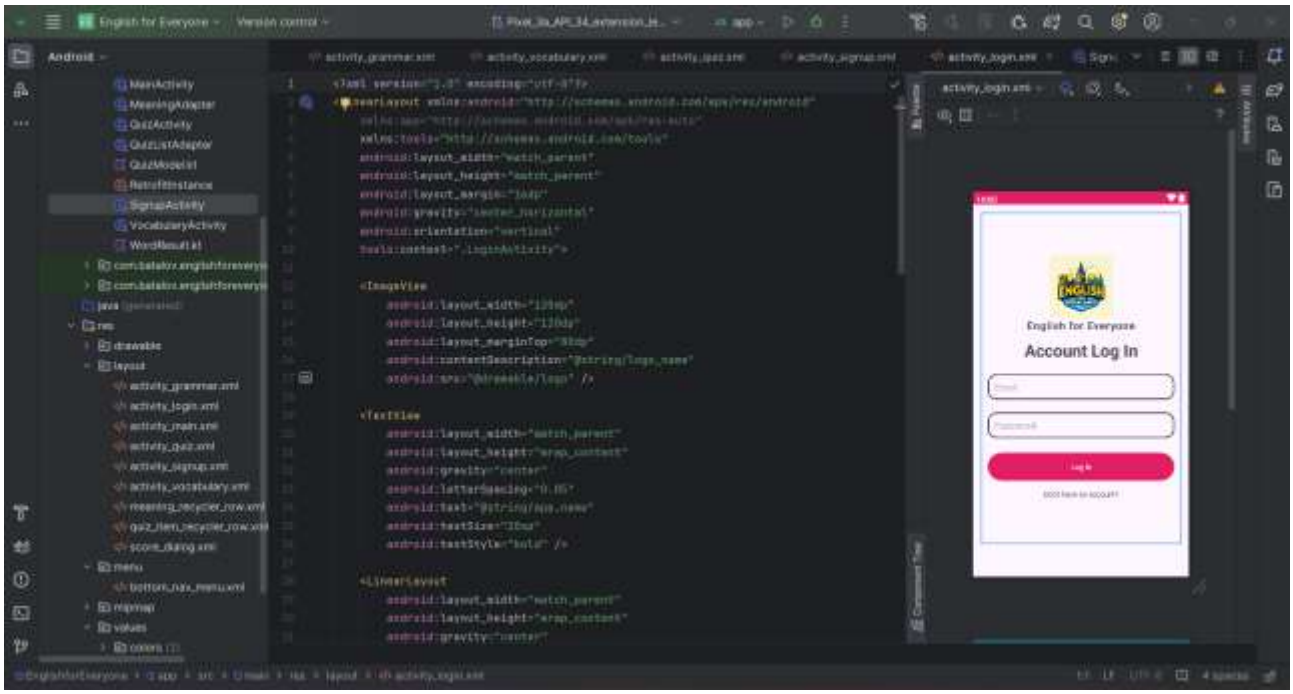
13. Офіційна документація Kotlin [Електронний ресурс]. - Режим доступу: https://kotlinlang.org/docs/home.html

14. Розробка мобільних додатків від А до Я: повний гайд [Електронний ресурс]. - Режим доступу: https://dan-it.com.ua/uk/blog/rozrobka-mobilnih-dodatkiv-vid-a-do-ja-povnij-gajd/

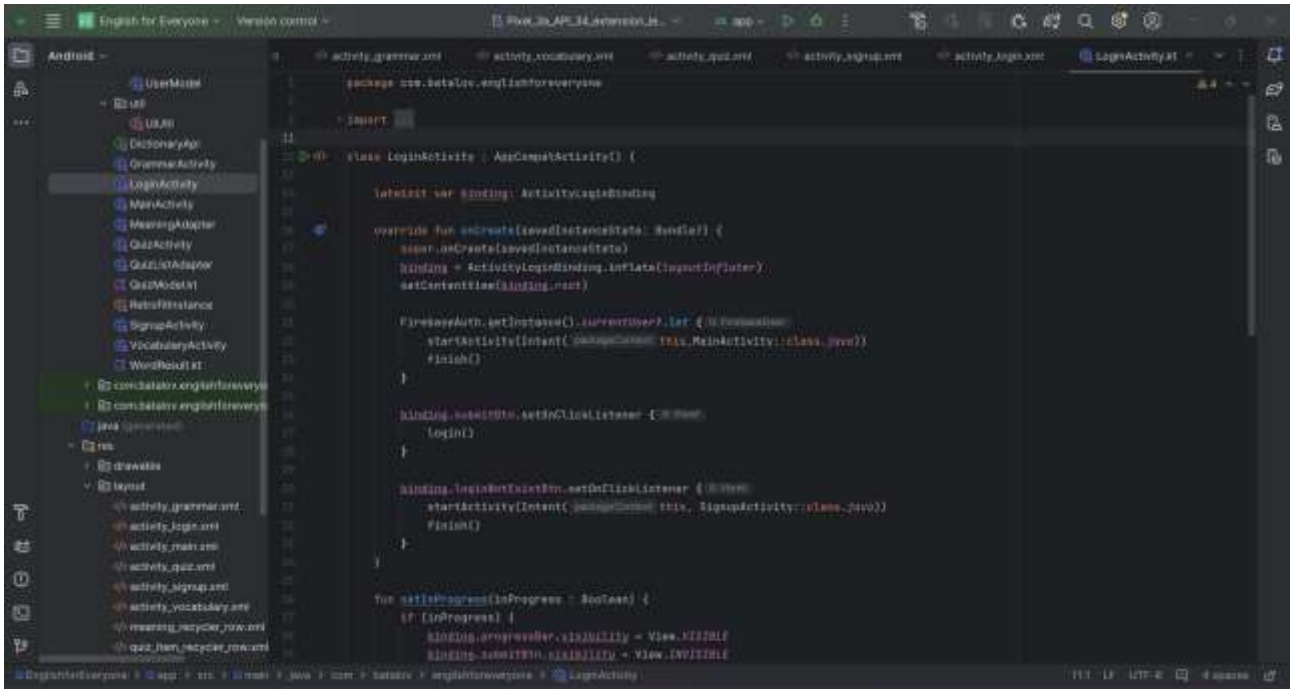
15. Найкращі додатки для вивчення англійської мови [Електронний ресурс]. - Режим доступу: https://englishprime.ua/uk/prilozheniya-dlya-izucheniya-anglijskogo/

ДОДАТОК А

Розробка коду дизайну в Android Studio.



Написання коду для роботи додатку, наприклад для входу у обліковий запис.



Файл strings.xml в Android Studio використовується для зберігання текстових рядків, які використовуються в додатку. Основні функції:

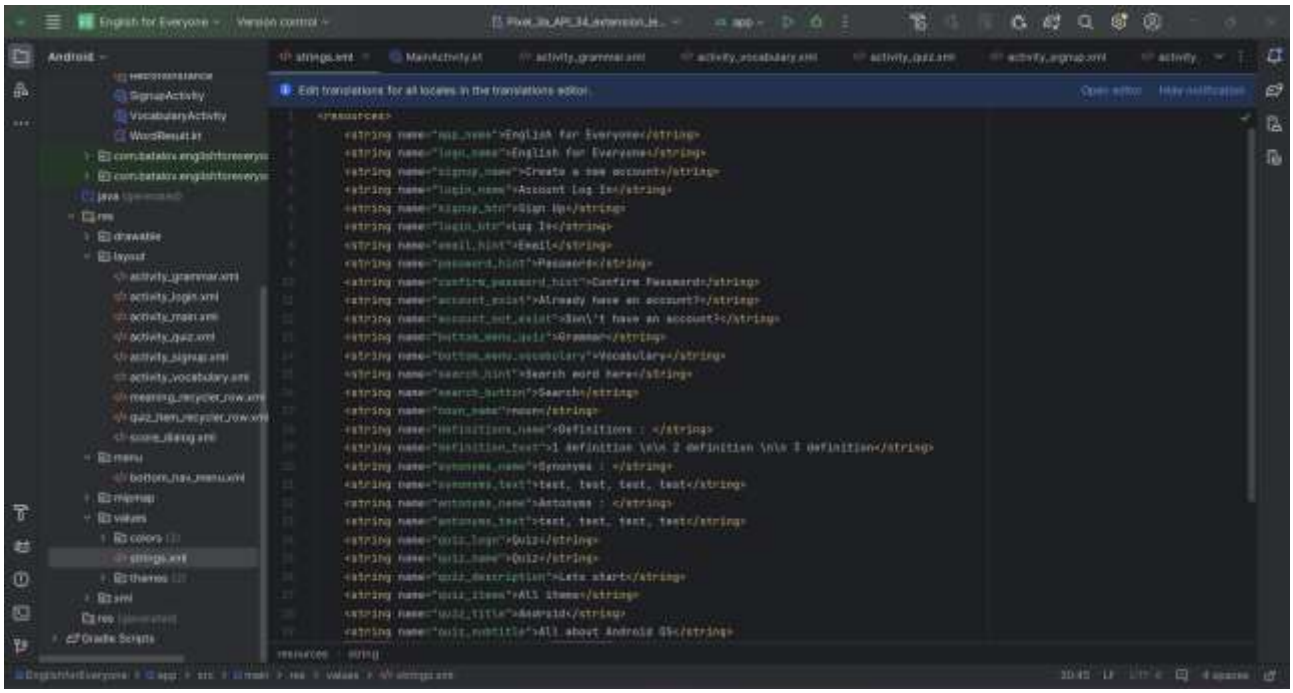
1. Централізоване зберігання рядків: Замість того, щоб розміщувати текстові рядки безпосередньо в коді, їх можна зберігати в цьому файлі.

2. Інтернаціоналізація (i18n): Файл strings.xml підтримує декілька версій для різних мов і локалей, що дозволяє легко локалізувати додаток.

3. Зміна рядків без зміни коду: Оскільки рядки відокремлені від коду, їх можна легко змінювати без необхідності змінювати та перекомпілювати код додатку.

4. Використання ресурсів у коді: Рядки з файлу strings.xml можна використовувати в коді додатку за допомогою методу getString().

Таким чином, файл strings.xml є важливим ресурсним файлом, який допомагає керувати текстовими рядками додатку, забезпечуючи централізоване зберігання, локалізацію та простоту оновлення.



Файл colors.xml використовується для визначення кольорів, які будуть використовуватись в додатку Android. Основні функції:

1. Централізоване визначення кольорів: Замість того, щоб використовувати значення кольорів безпосередньо в коді, їх можна визначити в цьому файлі для зручності та централізованого управління.

2. Легкість зміни кольорів: Завдяки централізованому визначенню кольорів, їх можна легко змінити в одному місці, без необхідності вносити зміни у весь код.

3. Підтримка різних тем: Кольори можна групувати для різних тем дизайну, що полегшує перемикання між темами.

Файл themes.xml використовується для визначення стилів та тем інтерфейсу додатку Android. Основні функції:

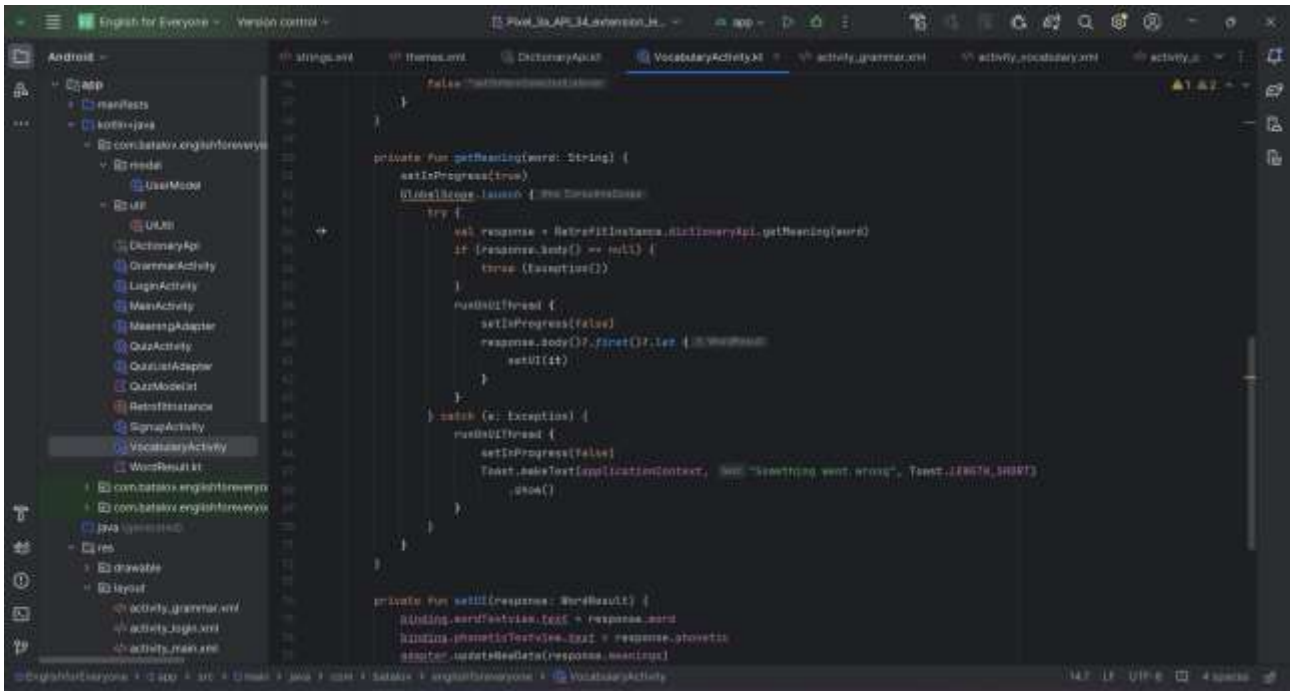
1. Визначення візуального стилю: Тут можна задати кольори, шрифти, розміри тексту, стилі елементів інтерфейсу тощо для уніфікованого дизайну.

2. Успадкування стилів: Теми можуть успадковуватись від базових тем Android, що дозволяє переписувати лише потрібні стилі.

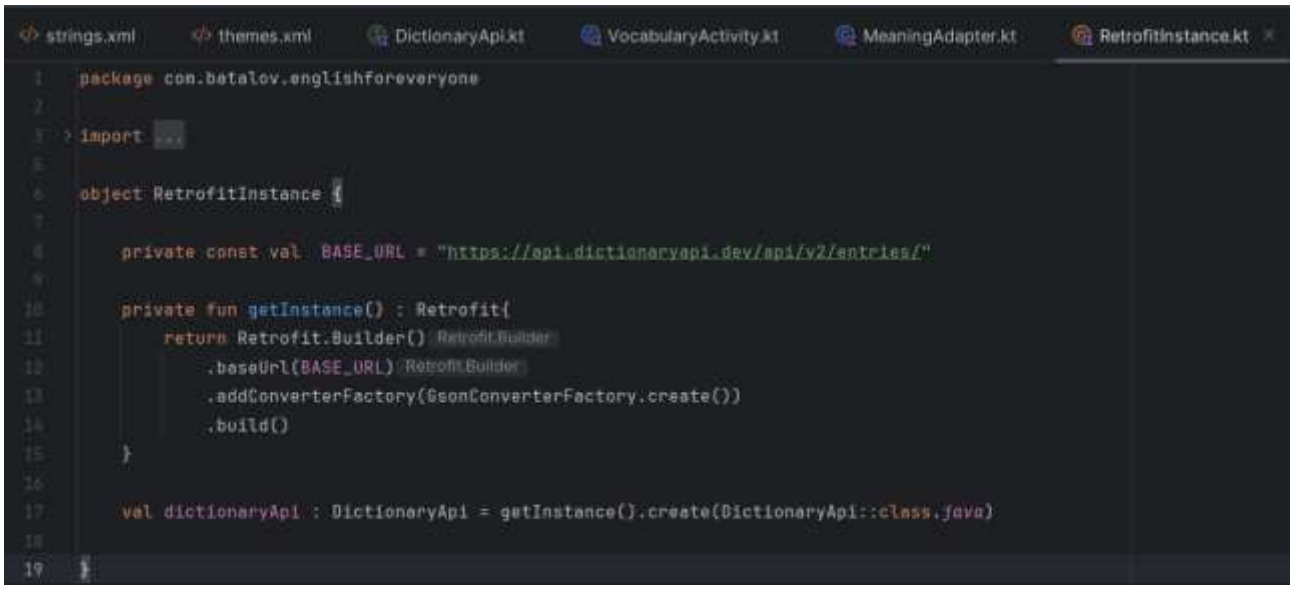
3. Легкість змін дизайну: Завдяки централізованому визначенню стилів, їх можна легко змінювати в одному місці, не змінюючи код.

4. Підтримка нічного режиму: Можна визначити теми для денного та нічного режимів.

Отримання значень слова у словнику за допомогою API.



Цей код налаштовує та створює екземпляр Retrofit для взаємодії з API словника. Він визначає базовий URL API, створює екземпляр Retrofit з конвертером Gson для обробки JSON, та створює екземпляр інтерфейсу DictionaryApi, який можна використовувати для виконання запитів до API в різних частинах додатку.



Цей код визначає структуру даних для результатів, що повертаються API словника, використовуючи data class в Kotlin. Він містить:

1. WordResult - клас для основних даних слова, включаючи слово, фонетичну транскрипцію та список значень.

2. Meaning - клас для кожного значення слова, містить частину мови, список визначень, синоніми та антоніми.

3. Definition - клас для окремих визначень значення слова.

Ці класи використовуються для автоматичного перетворення JSON-відповідей від API у об'єкти Kotlin за допомогою бібліотеки Gson в Retrofit, що полегшує роботу з даними.