

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Навчально-методичний комплекс
«Інститут післядипломної освіти»**

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри
_____ Віталій РОМАНКЕВИЧ

“ ____ ” червня 2021 р.

Дипломний проєкт

на здобуття ступеня бакалавра

зі спеціальності 123 «Комп'ютерна інженерія»

на тему: Комп'ютерні засоби для вивчення граматики англійської мови

Виконав\ла: студентка IV курсу, групи КВ-71

Государська Анастасія Олексіївна

Керівник к.т.н., доцент Орлова М.М.

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М.

Рецензент: професор кафедри ОТ, д.т.н., проф. Сімоненко В.П.

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань

Студент _____
(підпис)

Київ – 2021 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Спеціалізовані комп'ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ

«_20_» _жовтня_ 2020 р.

**ЗАВДАННЯ
на дипломний проєкт студента
Государська Анастасія Олексіївна
(прізвище, ім'я, по батькові)**

1. Тема проєкту «Комп'ютерні засоби для вивчення граматики англійської мови»,

керівник проєкту Орлова М.М., к.т.н, доцент,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «25» травня 2021 р. № 1331-С

2. Термін подання слухачем проєкту 08.06.2021

3. Вихідні дані до проєкту див. Технічне завдання

4. Зміст пояснювальної записки

- Основні концепції комп'ютерної графіки.
- Опис розроблених алгоритмів.
- Аналіз алгоритмів виводу зображення.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

- Структура програми. Схема алгоритму.
- Структура програми. Схема алгоритму.

- Архітектура додатку. Схема алгоритму.
- Схема життєвого циклу Android додатку. Схема алгоритму.

6. Консультанти розділів проєкту^{1*}

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М. доцент		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
	Вивчення літератури за тематикою проєкту	10.02.2021	
	Узгодження технічного завдання	20.03.2021	
	Розробка програмного забезпечення	26.03.2021	
	Відлагодження програмного продукту	10.04.2021	
	Підготовка матеріалів першого розділу дипломного проєкту	15.04.2021	
	Підготовка матеріалів другого розділу дипломного проєкту	23.04.2021	
	Підготовка матеріалів третього розділу дипломного проєкту	29.04.2021	
	Підготовка графічної частини дипломного проєкту	07.05.2021	
	Оформлення документації дипломного проєкту	09.05.2021	

Студент

(підпис)

Анастасія ГОСУДАРСЬКА

Керівник проєкту

(підпис)

Марія ОРЛОВА

^{1*} Консультантом не може бути зазначено керівника дипломного проєкту

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (50 с., 15 рис., 2 табл., 4 додатки).

Об'єктом розробки є створення ігрового додатку для навчання дітей молодшого шкільного віку граматичним явищам англійської мови.

Розроблений мобільний додаток дозволяє: розширювати процес здійснення рівного доступу до освіти з боку різних верств населення, отримувати миттєвий зворотній зв'язок і оцінку результатів навчання, здійснювати навчання персоналізовано у будь-якому місці і у будь-який час, підтримувати безперервність і ситуативних характер освітнього процесу, і найголовніше – поєднати навчальну, розважальну і релаксаційну функції, дозволяючи навчати молодших школярів, не перевантажуючи їх, стимулюючи їхній інтерес до пізнання нового, спрощуючи усвідомлення абстрактних реалій за рахунок візуалізації.

В процесі розробки були використані технології, вбудовані в середовище розробки INTELIJ IDEA, мовою програмування було обрано Java, мовою для маркування було обрано XML. В якості бази даних використовувалась SQLite.

В ході розробки:

- виявлено і проаналізовано мобільні додатки для вивчення Present Simple молодшими школярами;
- визначено переваги та недоліки існуючих мобільних додатків з точки зору ефективності для навчального процесу і зручності використання молодшими школярами;
- розроблено загальну модель додатку;
- описано схеми даних;
- визначено інструменти для розробки додатку;
- розроблено інтерфейс головної сторінки, сторінки вибору ігор та сторінок міні-ігор додатку.

Створення цього мобільного додатку дозволить усунути однотипність і нерізноманітність завдань, що характеризують більшість існуючих додатків, та призведе до втримання інтересу користувачів молодшого шкільного віку на більш тривалий час.

Ключові слова: мобільний додаток, ANDROID, JAVA, SQLITE, XML.

ABSTRACT

Qualification work, including an explanatory note (50 pages, 15 figures, 2 tables, 4 appendices).

The object of development is the development of a game supplement for teaching children of primary school grammar.

The mobile application allows you to: expand the process of equal access to education by different segments of the population, receive instant feedback and evaluation of learning outcomes, carry out personalized learning anywhere, anytime, maintain the continuity and situational nature of the educational process , and most importantly - to give impose, attentive and relaxing functions, allowing to teach young students, not to pass them, to stimulate revision of infection interest in learning a new round, burning sleep recognition. In the process of technology development, technology in the middle of the development of INTELIJ IDEA, a new software for image processing in Java, for data processing based on XML. SQLite was used as a database.

During development:

- mobile add-ons for studying Present Simple by young students were analyzed and analyzed;
- the advantages and disadvantages of existing mobile applications in terms of efficiency for the educational process and ease of use by young students are identified;
- the general model of the application is developed;
- descriptions of the data scheme;
- defined tools for application search;
- the interface of the main page, game selection page and application mini-game pages have been developed.

Sash of this mobile add-on will allow eliminating uniformity and opacity of tasks allowing to describe quantity of qualitative add-ons, and providing keeping of young generation's attention.

Keywords: mobile APP, ANDROID, JAVA, SQLITE, XML.

[illegible]

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№	Примітки
	A4	ІАЛЦ.045490.005 Д1	Структура проєкту	1		
			Креслення			
	A4	ІАЛЦ.045490.006 Д2	Взаємодія з користувачем	1		
			Структура проєкту			
			Креслення			
	A4	ІАЛЦ.045490.007 Д3	Архітектура	1		
			Креслення			
	A4	ІАЛЦ.045490.008 Д4	Життєвий цикл	1		
			Креслення			
		Диск CD-ROM	Текст пояснювальної			
			записки			
			Графічний матеріал	1		

					ІАЛЦ.045490.001 ОА	Арк.
						2
	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ	1
2. ПІДСТАВА ДЛЯ РОЗРОБКИ.....	1
3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ	1
4. ДЖЕРЕЛА РОЗРОБКИ.....	1
5. ТЕХНІЧНІ ВИМОГИ.....	1
5.1. Вимоги до програмного продукту, що розробляється	1
5.2. Вимоги до апаратного забезпечення	2
5.3. Вимоги до програмного та апаратного забезпечення користувача	2
6. ЕТАПИ РОЗРОБКИ	3

					ІАЛЦ. 045490.002 ТЗ								
Зм	Лист	№ докум.	Підп.	Дата	Комп'ютерні засоби для вивчення граматики англійської мови				Літ.	Лист	Листів		
Розроб.		Государська										1	4
Перев.		Орлова											
Н. контр.		Клятченко			Технічне завдання				НТУУ «КПІ ім. Ігоря Сікорського», ФПМ, КВ-71				
Затв.		Тарасенко											

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Комп'ютерні засоби для вивчення граматики англійської мови».

Галузь застосування: на уроках англійської мови у молодшій школі та у позакласній діяльності .

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проектування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення мобільного додатку для вивчення граматики англійської мови.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до програмного продукту, що розробляється

- сумісність операційною системою Android 4.4 і вище;
- адаптивна верстка;
- можливість обирати рівні ігор;

					ІАЛЦ.045490.002 ТЗ	Лист 1
Зм	Лист	№ докум.	Підп.	Дата		

- можливість змінювати налаштування;
- можливість змінювати темну тему на світлу;
- зберігання прогресу;
- наявність діалогових вікон з інструкціями до кожного рівня.

5.2 Вимоги до апаратного забезпечення

- Оперативна пам'ять: 1 Гб;

5.3 Вимоги до програмного та апаратного забезпечення користувача

- Операційна система Android;

					ІАЛЦ.045490.002 ТЗ	Лист
						2
Зм	Лист	№ докум.	Підп.	Дата		

ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Вивчення літератури за тематикою проекту	15.04.2021
2.	Розроблення та узгодження технічного завдання	30.04.2021
3.	Аналіз існуючих рішень	05.05.2021
4.	Підготовка матеріалів першого розділу дипломного проекту	10.05.2021
5.	Підготовка матеріалів другого розділу дипломного проекту	18.05.2021
6.	Підготовка графічної частини дипломного проекту	20.05.2021
7.	Оформлення документації дипломного проекту	25.05.2021
8.	Попередній огляд матеріалів диплому на кафедрі	30.05.2021

Зм	Лист	№ докум.	Підп.	Дата

ІАЛЦ.045490.002 ТЗ

Лист

3

[illegible]

[illegible]

Пояснювальна записка до дипломного проєкту

на тему: Комп'ютерні засоби для вивчення граматики англійської мови

Київ – 2021 рік

ЗМІСТ

ПЕРЕЛІК ТЕРМІНІВ, СКОРОЧЕНЬ І ПОЗНАЧЕНЬ.....	2
ВСТУП.....	3
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ.....	6
1.1. Загальні положення та аналіз предметної області.....	6
1.2. Аналіз аналогічних застосунків.....	7
2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ.....	15
2.1. Вибір платформи.....	15
2.2. Історія Android.....	16
2.3. Вибір середовища розробки.....	21
2.4. Вибір мови програмування.....	22
2.5. Вибір системи керування базами даних.....	23
2.6. Вибір способу технічної розробки.....	24
3. СТРУКТУРА І АЛГОРИТМ РЕАЛІЗАЦІЇ ПРОГРАМИ	28
2.1. Загальний опис системи.....	28
2.2. Архітектура системи.....	29
2.3. Опис розробленого додатку.....	32
4. МОЖЛИВІ ВАРІАНТИ РОЗВИТКУ	44
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	50

					ІАЛЦ.045490.004 ПЗ				
Зм.	Арк.	№	Підп.	Дата	1 <u>«Комп'ютерні засоби для вивчення граматики англійської мови».</u> Пояснювальна записка	Літ.		Арку	Аркушів
		Государська А.О						1	50
		Орлова М.М.							
Н. контр.		Клятченко Я.М.				НТУУ "КПІ" ФПМ			
Затв.		Романкевич В. О				КВ-71			

ДОДАТКИ

Додаток А. Копії графічного матеріалу.

ІАЛЦ.045490.005 Д1. Взаємодія з користувачем. Структура проєкта. Схема алгоритму.

ІАЛЦ.045490.006 Д2. Взаємодія з користувачем. Схема алгоритму.

ІАЛЦ.045490.007 Д3. Android додаток для мобільних пристроїв. Архітектура. Схема структурна.

ІАЛЦ.045490.008 Д4. Android додаток для мобільних пристроїв. Життєвий цикл. Схема структурна.

Додаток Б. Фрагменти програмного коду

Додаток В. Презентація

					ІАЛЦ.045490.004 ПЗ	Лист
						5
Зм	Лист	№ докум.	Підп.	Дата		

ПЕРЕЛІК ТЕРМІНІВ, СКОРОЧЕНЬ І ПОЗНАЧЕНЬ

ОС – операційна система

SDK (software development kit) – комплект для розробки програмного забезпечення, що допомагає розробникам створювати застосунки для конкретних платформ

URL (Uniform Resource Locator) – система уніфікованих адрес електронних ресурсів, або визначник місцезнаходження ресурсу (файлу)

XML (eXtensible Markup Language) – розширювана мова маркування

API (Application Programming Interface) -

APK (Android Package) – формат архівних виконуваних файлів – застосунків для Android та інших операційних систем, що базуються на Android

IDE (Integrated Development Enviroment) - комплекс програмних засобів, що використовуються програмістами для розробки програмного забезпечення

ПО – програмне забезпечення

ВСТУП

У сучасному світі володіння англійською мовою, що є засобом міжнародного спілкування, набуває все більшого значення. У зв'язку із цим вивченню вищезазначеної мови у сучасній українській школі приділяється особлива увага. У більшості шкіл України цей процес розпочинається з першого класу. Тому саме у початковій школі має здійснюватися формування основних фонетичних, лексичних, орфографічних і граматичних навичок, на яких базуватиметься подальше засвоєння мови у середніх та старших класах.

Засвоєння основних правил побудови речень є першочерговим завданням, що має бути вирішене на уроках англійської мови у початкових класах. У сучасній методиці викладання іноземних мов провідним є судження про те, що молодші школярі ще не готові до усвідомлення структури граматичних явищ, і, відповідно, здатні лише до заучування певного набору фраз без розуміння того, як вони побудовані. Проте, не всі методисти підтримують вищезазначену думку, зазначаючи, що гра у навчальному процесі дозволяє усвідомити і автоматизувати граматичні явища, що поза грою, можуть викликати значені складнощі.

Мобільні додатки наразі вже стали найзручнішими і найдоступнішими інструментами поєднання освітніх та ігрових мотивів із широким спектром можливостей. Вони дозволяють розширювати процес здійснення рівного доступу до освіти з боку різних верств населення, отримувати миттєвий зворотній зв'язок і оцінку результатів навчання, здійснювати навчання персоналізовано у будь-якому місці і у будь-який час, підтримувати безперервність і ситуативних характер освітнього процесу, і найголовніше – поєднати навчальну, розважальну і релаксаційну функції, дозволяючи навчати молодших школярів, не перевантажуючи їх, стимулюючи їхній інтерес до пізнання нового, спрощуючи усвідомлення абстрактних реалій за рахунок візуалізації.

На сучасному етапі розвитку інформаційних технологій мобільні додатки

					ІАЛЦ.045490.004 ПЗ	Лист
						7
Зм	Лист	№ докум.	Підп.	Дата		

для вивчення іноземних мов не здатні повністю замінити взаємодію “вчитель – учень”, але можуть суттєво доповнити і урізноманітнити навчальний процес. Особливо важливим інструментом вони можуть стати за необхідності здійснення учнями дистанційного навчання, під час якого учням доводиться більше працювати самостійно. Отже, актуальність обраної теми дослідження полягає у тому, щоб продемонструвати можливості використання мобільних додатків у свідомому вивченні молодшими школярами граматичних явищ та довести їхній потужний потенціал для здійснення самостійної роботи учнів із матеріалом, що підлягає опануванню у початковій школі.

Мета дипломного проєкту – розробка ігрового додатку для навчання дітей молодшого шкільного віку граматичним явищам англійської мови.

В даному дипломному проєкті вирішуються наступні завдання:

- 1) виявити та проаналізувати мобільні додатки для вивчення Present Simple молодшими школярами;
- 2) визначити переваги та недоліки існуючих мобільних додатків з точки зору ефективності для навчального процесу і зручності використання молодшими школярами;
- 3) розробити загальну модель додатку;
- 4) описати схеми даних;
- 5) визначити інструменти для розробки додатку;
- 6) розробити інтерфейс головної сторінки, сторінки вибору ігор та сторінок міні-ігор додатку.

Дипломна робота складається із вступу, трьох частин і загальних висновків. У першій частині здійснюється загальний огляд існуючих мобільних додатків під операційну систему Android, виявлення їхніх переваг і недоліків та формування основних вимог до функціональності мобільних додатків в освітньому просторі молодшої школи. У другій частині реалізовано розробку загальної моделі додатку, описано схеми даних, обґрунтовано вибір інструментів

для його розробки. У третій частині представлено основний функціонал додатку, а саме інтерфейс головної сторінки, інтерфейс сторінки вибору ігор та інтерфейс міні-ігор для вивчення граматичної теми “Present Simple”.

					ІАЛЦ.045490.004 ПЗ	Лист
						9
Зм	Лист	№ докум.	Підп.	Дата		

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1. Загальні положення та аналіз предметної області

XXI століття – період розквіту інформаційних технологій. У наш час людина може натискати педаль газу в автомобілі, навіть не здогадуючись, що за рухом стежать сотні датчиків і мікропроцесорів, покликаних полегшити та зробити безпечнішим процес керування автомобільним транспортом. З кожним роком виробники вигадують все більш хитромудрі функції, що призводить до необхідності зручної взаємодії користувача і всієї зовнішньої інфраструктури. З'явилася необхідність у використанні чогось на кшталт пульта керування електронікою. Інженери Apple, а потім і Google, знайшли рішення. Вони створили операційну систему для телефонів, що дозволяє створення сторонніх додатків із досить різним функціоналом (починаючи від можливості читати електронні книги з екрана мобільних телефонів і закінчуючи управлінням побутовою технікою в «розумному будинку»). Отже, Android – портативна (мережева) операційна система для комунікаторів, планшетних комп'ютерів, електронних книжок, цифрових програвачів, наручних годинників, нетбуків і смартбуків, заснована на ядрі Linux [1].

Спочатку ця операційна система розроблялась компанією Android Inc., яку потім купила Google. Більше 75% смартфонів, проданих в третьому кварталі 2012 року, були оснащені операційною системою Android. Тепер кожен розробник електронного пристрою має можливість переробити Android під свій пристрій, гарантуючи сумісність сторонніх додатків для цієї ОС зі своїм устаткуванням. Якщо до виходу Android кожен виробник електронного пристрою самостійно писав або купував у когось операційну систему, втрачаючи таким чином масу корисних програм, то після перед виробниками частіше постає інше питання – яку версію Android потрібно підтримувати [2].

Ще 10-20 років тому повноцінне дистанційне навчання було практично неможливим. Необхідність переїзду в інше місто або навіть країну була однією з найбільш популярних причин віддавання переваги місцевим вузам, а не

університетам своєї мрії. Наразі ситуація кардинально змінилася. Маючи сучасний смартфон або планшет, ви маєте змогу завантажити сотні додатків, оптимізованих для навчання. Наприклад, LinguaLeo – для вивчення англійської мови, Rex4Fun від Microsoft Research – для вивчення програмування, довідники і словники Lingvo, Britannica та ін [2]. Більш того, додатки можуть забезпечити не тільки відображення інформації, а й інтерактив по її засвоєнню, перевірці.

1.2. Аналіз аналогічних застосунків

Сучасний формат шкільної освіти передбачає обов'язкове залучення до навчального процесу ігрових елементів, що дозволяють оптимізувати процес засвоєння нових знань. Особливого значення ігрова форма навчання набуває в молодшій школі, адже на цьому етапі в учнів ще слабо розвинене теоретичне та абстрактне мислення. Розвиток інформаційних технологій та необхідність здійснювати навчальний процес дистанційно у зв'язку з карантинними обмеженнями, пов'язаними із поширенням Covid-19, призводять до перенесення навчальних ігор із формату безпосередньої взаємодії «вчитель-учень» у віртуальний простір. Найбільш поширеними, доступними і легкими у використанні молодшими школярами наразі стають мобільні додатки, що здатні сполучати навчальну й ігрову складові та можуть застосовуватися задля автоматизації отриманих знань учнів у позакласній роботі з ними [3].

У рамках цього дипломного проекту було здійснено аналіз найбільш популярних мобільних додатків для вивчення англійської мови з метою з'ясування їхніх переваг і недоліків при використанні молодшими школярами [3]. Як приклад, було розглянуто реалізацію процесу презентації та автоматизації учнями граматичної теми Present Simple.

					ІАЛЦ.045490.004 ПЗ	Лист
						11
Зм	Лист	№ докум.	Підп.	Дата		

1.2.1. Додаток «LinguaLeo»

LinguaLeo є освітньою платформою, побудованою на ігровій механіці, із загальною кількістю користувачів, що перевищує 17 мільйонів. Цей сервіс доступний через додатки для IOS, Android, Windows Phone, а також як веб-додаток і розширення для браузера Chrome.

Серед переваг сервісу варто виділити загальну концепцію, що беззаперечно викликатиме зацікавленість у дітей: гравець має годувати левеня, успішно виконуючи завдання та отримуючи для нього фрикадельки у винагороду. Гра надсилає гравцю через email нагадування про необхідність зайти і нагодувати свого вихованця, що позитивно впливає на вироблення системного підходу до навчання. На платформі зібрано потужну базу автентичних матеріалів із реальних книг, статей, аудіозаписів та відеофільмів (рис. 1.1). Для того щоб зорієнтуватися в цих «джунглях», розробниками пропонується карта, на якій прокладається шлях, яким має рухатися гравець, опановуючи темою за темою. В основі тренування граматики покладено методику запам'ятовування патернів певних правил, що дозволяє сформувати міцний базис для формування граматично правильного мовлення [1].

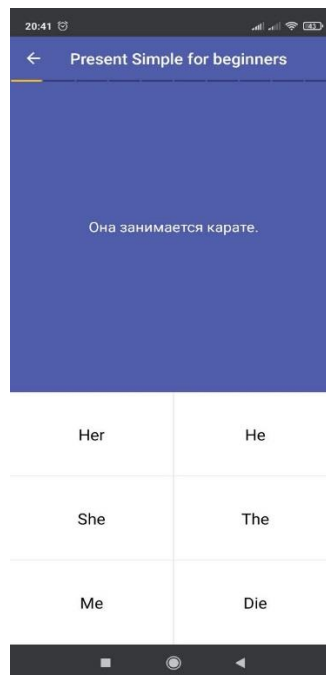


Рисунок 1.1 - Приклад вправи

Одним з основних недоліків цього сервісу, про які зараз говорять і самі розробники, є те, що не зважаючи на наявність гейміфікації, користувачі швидко втрачають зацікавленість у процесі. На нашу думку, проблема полягає, перш за все, в тому, що запропоновані завдання не відрізняються особливою різноманітністю. Для опанування Present Simple, так само як і інших граматичних тем, використовується лише одне завдання – формування речень із запропонованих слів, що швидко може стати нецікавим, особливо для дітей. Окрім того, саме граматичне правило презентоване у вигляді простого неструктурованого тексту, що ускладнює розуміння та засвоєння. Кращим було б його сприйняття, як би матеріал був викладений у вигляді схем, таблиць тощо [2].

Також залишається нереалізованим принцип переходу від простого до більш складного: зразу подаються стверджувальні, заперечні та запитальні речення, що більше підходить для тренування вже вивченого матеріалу, а не для його первинного засвоєння. Отже, не зважаючи на загальну концепцію сервісу, що могла б захопити дітей молодшого шкільного віку, сама подача матеріалу і запропоновані завдання більше підходять для старших школярів та дорослих [1].

1.2.2. Додаток «Duolingo»

Duolingo є безкоштовною платформою для вивчення іноземних мов в ігровій формі та здійснення краудосорсингових перекладів із загальною кількістю активних користувачів, що перевищує 25 мільйонів. Сервіс доступний через додатки для IOS, Android, Windows Phone і веб-версії [2].

Розробники вважають, що завдяки гейміфікації, платформа може застосовуватися для користувачів різного віку, починаючи із дошкільнят. Пропонується також формат для шкіл, що дозволяє викладачам створювати для своїх учнів аккаунти, призначати їм завдання і контролювати їхнє виконання [3].

Матеріал для вивчення подається поступово від слова до речення та групується тематично (рис. 1.2). У процесі його засвоєння користувач проходить певні рубежі, заробляє ігрову валюту, спостерігає за своїм прогресом,

отримує підбадьорювання від ігрових персонажів, втрачає життя, якщо припускається помилок, що додає азарту і підштовхує до досягнення подальших успіхів. Програма відстежує помилки, що допускаються під час проходження, і пропонує

цей матеріал ще, доки користувач не буде виконувати завдання правильно. Набір слів на перших етапах є досить обмеженим, із частими повторами. Це позитивний момент, оскільки він дає можливість концентруватися на граматиці, що є досить важливим для формування бази [4].

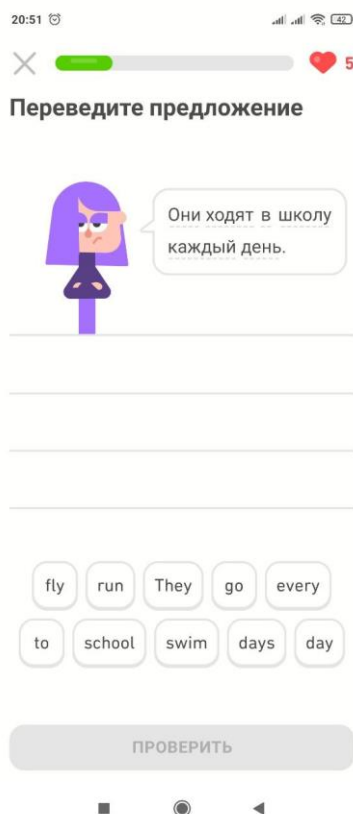


Рисунок 1.2 - Приклад вправи

Проте методика викладення і відпрацювання матеріалу не позбавлена недоліків. Починаючи рух від засвоювання простих слів, відпрацювання яких здійснюється у завданнях на добирання правильного підпису до малюнка чи двостороннього перекладу до словосполучення, програма несподівано починає пропонувати речення із використанням Present Simple ще до досягнення учнем цього етапу і ознайомлення його із правилами формування та використання

цього часу. Так само, як і у LinguaLeo, правила подаються сухо і неструктуровано. До того ж Present Simple і Present Continuous, розрізнення яких є складним для користувачів, у рідній мові яких є лише один теперішній час, подаються разом, що суттєво ускладнюватиме процес засвоєння. Коли подається to be у Present Simple, паралельно пропонуються завдання із використанням цього дієслова в інфінітивній формі, у наказовому способі і у майбутньому часі, що також не сприятиме систематизації знань [5].

Курс є довгим, завдання – однотипні, загалом базуються на методі прямого двостороннього перекладу (вибір правильної форми із кількох запропонованих, переклад фраз шляхом встановлення кубиків із словами у правильному порядку) із додаванням завдань на підстановку і відпрацювання правопису. З таким підходом буде важко утримувати зацікавленість у процесі не тільки дітей, а й дорослих [5].

1.2.3. Додаток «Simpler»

Simpler популярний мобільний додаток для IOS, Android, Windows Phone (веб-версія відсутня) із чудовими авторськими ілюстраціями, зручним інтерфейсом та продуманим методичним апаратом, що базується на логічній (від простого до складного) і дозованій подачі матеріалу та систематичному його повторенні [6].

Програма у рівному ступені приділяє увагу вивченню лексичних одиниць та основам побудови речень, що подається блоками. Для засвоєння кожного з них пропонується спочатку ознайомлення з новими лексичними одиницями, потім презентація правила із подальшою їх автоматизацією шляхом виконання вправ. Після засвоєння певної кількості уроків пропонується підсумковий контроль. По мірі переходу від блоку до блоку відбивається прогрес, після проходження рівня пропонується сертифікат. Пропускати блоки неможна, також не вийде обійти презентацію слів та граматичних правил і зразу перейти до їхнього відпрацювання у вправах [6].

Хоча самі правила подаються несхематизовано і маленькими буквами, що досить складно для сприймання, особливо дітьми, позитивним є ілюстрація цих правил у реченнях, де правильна послідовність слів підсвічується. Для презентації слів дібрано красиві ілюстрації, що привертають увагу і запам'ятовуються, що у свою чергу сприятиме формуванню міцних асоціацій між малюнком, англійським словом і його значенням (рис. 1.3). Серед плюсів введення лексичного матеріалу слід відмітити не простий перехід від слова до слова, що може відбуватися несвідомо, а необхідність обрати серед двох варіантів перекладу англійського слова того, що йому відповідає, згідно малюнка.



Рисунок 1.3 - Інтерфейс програми

Автоматизація лексичного і граматичного матеріалу відбувається так само одноманітно, як і в інших додатках: 1) потрібно розставити кубики із написаними на них англійськими словами так, щоб утворилося граматично правильне речення; 2) підставити у речення потрібний за смислом чи формою елемент; 3) перекласти речення з англійської мови на російську і навпаки.

Після того, як учнями засвоєно основи формування речень і певний набір слів, за допомогою Simplr можна розвивати навички аудіювання та читання на матеріалі адаптованих детективних історій, але доступ до них можливий на платній основі. Мобільний додаток Simplr регулярно отримує оновлення, коректно та стабільно працює, рівень його безпеки відповідає високим показникам [6].

Порівняльна характеристика розглянутих і проаналізованих ресурсів представлена в таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз наведених вище застосунків

Критерії порівняння	LinguaLeo	Duolingo	Simplr
Платформи	IOS, Android, Windows Phone, вебдодаток	IOS, Android, Windows Phone, веб-додаток	IOS, Android, Windows Phone
Інтерфейс	Строге оформлення, орієнтація на дорослу та підліткову аудиторію	Строге оформлення, орієнтований на дорослу та підліткову аудиторію	Авторські ілюстрації, орієнтований на використання у молодшій та середній школах
Ігрова концепція, вимірювання прогресу	Є	Є	Є
Порядок викладення матеріалу	блочна подача, низька реалізація переходу від простого до складного	блочна подача, низька реалізація переходу від простого до складного	блочна подача, реалізовано перехід від простого до складного

Подача граматичних правил	Відсутність схематизації	Відсутність схематизації	Відсутність схематизації, активне засвоєння правил у прикладх
Різноманітність завдань	Низька	Низька	Низька

Висновок до першого розділу

Отже, порівнявши три популярних додатки, можемо зробити такі висновки:

- в усіх додатках реалізовано цікаву ігрову концепцію та вимірювання прогресу;
- не зважаючи на те, що розробники пропонують свої додатки для широкої аудиторії, LinguaLeo та Duolingo є більш цікавими для дорослої та підліткової аудиторії, тоді як Simpler може використовуватися у молодшій та середній школах;
- у всіх трьох додатках реалізовано блочну подачу матеріалу, проте поступового переходу від простого до складного найбільше дотримуються розробники Simpler;
- в усіх трьох додатках подача граматичних правил відбувається несхематизовано, що ускладнює сприймання;
- однотипність і нерізноманітність завдань є основним недоліком усіх трьох проаналізованих додатків, що призводить швидкої втрати інтересу користувачів.

2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1 Обґрунтування вибору платформи

Переваги розробки на Android:

- Низька вартість розробки та пристроїв. Наприклад, для iOS розробки потрібен макбук, що коштує досить багато. Написання коду під Android можливе на будь-якій операційній системі: macOS, Linux чи Windows.
- Однаковий стек технологій для всіх проектів. Людині, що хоче займатися розробкою під Android буде досить опанувати одну з двох мов, офіційну середу розробки та мінімум інструментів.
- Більша частина Android коду відкрита, тобто існує така можливість завантажити код на свій пристрій та досліджувати, як все працює зсередини. Сам код розбито на модулі, що робить його нескладним в розмінні. Якщо щось у додатку не працює, то існує можливість знайти початковий фрагмент коду та розібратися із проблемою, що виникла. Але задля безпеки існують частини коду, яких немає у вільному доступі. Наприклад, робота Play Market зсередини.
- Документація написана на досить високому рівні. Існує безліч інструкцій, що роблять комунікацію між спеціалістами по всьому світу легшою.
- Велика активність ентузіастів в інтернеті. Розробник на будь-якому етапі свого навчання може отримати кваліфіковану підтримку на таких платформах як StackOverflow та GitHub.
- Після натиснення кнопки Опублікувати додаток стає доступним для завантаження. Але потрібні ще декілька годин доки він пошириться по дата центрам.
- Вартість опублікування додатку. Порівняно з AppStore, в Google Play Market публікація відбувається за ціну меншу в 4 рази, також можна заважити, що платіж потребується лише один раз, а не щорічно.
- Java. Вивчення цієї мови — досить непоганий старт для людини, що починає займатися розробкою, оскільки є така можливість освоїти основні концепції

Зм	Лист	№ докум.	Підп.	Дата

ІАЛЦ.045490.004 ПЗ

Лист

19

ООП, із якими швидкість вивчення наступної мови збільшується. На Java, окрім мобільних додатків, пишуться веб-застосунки, а також ігри та сервери. Тому можлива зміна сфери діяльності без зміни основного інструменту розробки. А Swift та Objective-C — специфічні мови, що використовуються лише для iOS.

- Сфери застосування. Розробка можлива не лише для смартфонів, а і для розумних годинників, Smart TV, VR/AR, тому можливість знайти цікавий для себе проект є більш ймовірною [7].

Недоліки мобільної розробки під Android:

- Багато пристроїв. Необхідність підтримувати додатки для кожної нової моделі смартфонів різних виробників. На деяких неякісних пристроях додаток може працювати некоректно через апаратні проблеми, а не через провину розробника.
- Велика кількість екранів. Для вирішення проблеми різного співвідношення сторін недостатньо просто розтягнути інтерфейс застосунку. Вирішенням цієї проблеми стає підтримка декількох типів екранів і створення копій [7].
- Декілька версій операційної системи. Актуальним є не тільки остання і передостання версії. Google не може оновлювати систему на всіх пристроях. У цьому випадку актуальними опиняються одразу декілька версій Android. Не всі виробники швидко і якісно оновлюють свої пристрої [7].

2.2 Розвиток ОС Android

5 листопада 2007 року прес-реліз Альянсу Open Handset Alliance створив основу для майбутнього платформи Android. Альянс заявив про такі цілі Android як: «сприяння інноваціям на мобільних пристроях та надання споживачам кращого досвіду для користувачів, ніж більшість з того, що доступне на сучасних мобільних платформах». На той час у всьому світі було використано понад двох мільярдів мобільних телефонів, порівняно з 4,6 мільярдами, що використовувались у 2010 році. Однак між різними компаніями, що постачали мобільні пристрої, не було координації платформ. Завдяки введенню Android,

одна операційна система позбавила потреби перевтілювати телефонні програми та проміжне програмне забезпечення. Компанії, що створюють нові пристрої, тепер можуть набагато пильніше зосередитись на апаратному забезпеченні та базових компонентах.

Але ці компанії не єдині, хто виграв від запуску Android; розробники програмного забезпечення тепер можуть випускати додатки на кілька пристроїв із дуже невеликими змінами в базовій кодовій основі. Це дозволило розробникам витратити більше часу на роботу над програмами, на яких працювали ці телефони, і створювати багаті та вражаючі програми, до яких ми всі звикли. Частково це було пов'язано з філософією відкритого коду, що стоїть за Android, та ліцензією Apache, яка є ліцензією, що використовується для більшості вихідних кодів Android [10].

Android OS - це стек технологій з відкритим кодом, який працює на понад 400 мільйонах пристроїв по всьому світі (рис. 2.1). Цей стек технологій складається з різних компонентів, які дозволяють розробникам та виробникам пристроїв працювати самостійно [10]. Це можна розділити на п'ять основних частин - програми, фреймворки програм, власні бібліотеки, середовище виконання Android та ядро Linux.



Рисунок 2.1 – Архітектура Android системи

Додатки існують на найвищому рівні. Це інструменти, котрі всім, хто користується Android, найбільш знайомі. Android постачається з різними надійними програмами, що підтримують повсякденні потреби телефону, такими як обмін повідомленнями, електронна пошта, перегляд Інтернету та різні сторонні програми. Ці програми в основному написані мовою програмування Java. У нещодавній судовій справі з Oracle керівник Google Android, Енді Рубін, пояснив, чому обрав Java як мову використання для розробників. Основними положеннями Рубіна було те, що Java мала відоме торговельне ім'я і що Java викладається майже у всіх університетах світу [10].

Ці програми поширюються різними способами, найчастіше з Google Play Store (раніше Android Marketplace); однак ОС Android також підтримує встановлення програм через USB-з'єднання та з SD-карти [10].

Android надає розробникам можливість та інструменти для створення обширних, інтерактивних, насичених графічних додатків для користувачів і націлений на розгортання цих додатків у Google Play Store. Розробники мають доступ до тих самих API, що використовуються всередині основних додатків, а також доступ майже до всіх існуючих бібліотек Java. Процес розробки програм для Android див. У Розділі 6: Встановлення Android SDK для розробки додатків Intel.

Наступний рівень - там, де дорога розходиться. Власні бібліотеки та середовище виконання Android існують приблизно в одному просторі. Власні бібліотеки складаються та попередньо встановлюють двійкові файли C / C ++, від яких залежить система Android. Сюди входять усі бібліотеки у зеленому розділі (рис. 2.1). Наступні розділи містять описи деяких найбільш відомих власних бібліотек та їх функцій в Android [10].

Це часто називають менеджером вікон Android. Surface Manager використовується для складання того, як буде виглядати будь-який окремий екран. Він також робить деякі більш тонкі речі, які допомагають Android працювати безперебійно, наприклад, позаекранна буферизація та переходи.

Зм	Лист	№ докум.	Підп.	Дата

SQLite. Це база даних, яка використовується для збереження інформації протягом сеансів пристрою Android. На Android база даних SQLite зберігається всередині внутрішньої пам'яті пристрою, тому SD-карти можна міняти місцями, не втрачаючи інформацію про пристрій.

WebKit дозволяє відображати та відображати HTML на Android дуже швидко та ефективно. Це движок браузера за замовчуванням в системі Android, який доступний для системних та сторонніх додатків.

Усередині середовища виконання Android є два основні компоненти: основна бібліотека Java, яку забезпечує Android, та віртуальна машина Dalvik. Віртуальна машина Dalvik - це реалізація Google Java, яка оптимізована для використання на мобільних пристроях. Більш конкретні відмінності в Dalvik є дуже технічними та не висвітлюються в цій книзі.

Останнім із шарів є ядро Linux. Спочатку Android базувався на ядрі Linux 2.6, з деякими оптимізаціями для мобільного використання. Поточні версії Android базуються на ядрі Linux 3.1. Ядро Linux забезпечує доступ якомога ближче до апаратного забезпечення. Як результат, драйвери записуються в простір ядра, щоб працювати якомога швидше та ефективніше. Сюди входять такі речі, як управління внутрішніми радіоприймачами, увімкнення стереосистеми та камери, робота з живленням та зарядкою акумулятора, а також управління фізичною клавіатурою або кнопками на пристрої. Ядро Linux, як і Android, є проектом з відкритим кодом і широко використовується, особливо на серверах в корпоративних середовищах [10].

Після придбання Android, Inc. було створено проект з відкритим кодом Android (AOSP), який з тих пір очолював Google. AOSP відповідає за розробку та підтримку стеку програмного забезпечення Android. Як заявляє Google, мета проекту полягає в наступному:

Метою проекту з відкритим кодом Android є створення успішного реального продукту, який покращує мобільний досвід для кінцевих користувачів [10].

Android розробляється та підтримується з урахуванням зворотних можливостей (рис. 2.2). Це означає, що нові пристрої можуть запускати додатки, розроблені аж до кексу Android (1.5). Офіційна підтримка Android SDK сходить лише до Cupcake (1.5), тому додатки, написані для пристроїв до Cupcake, не гарантовано працюють на найновіших пристроях Android.

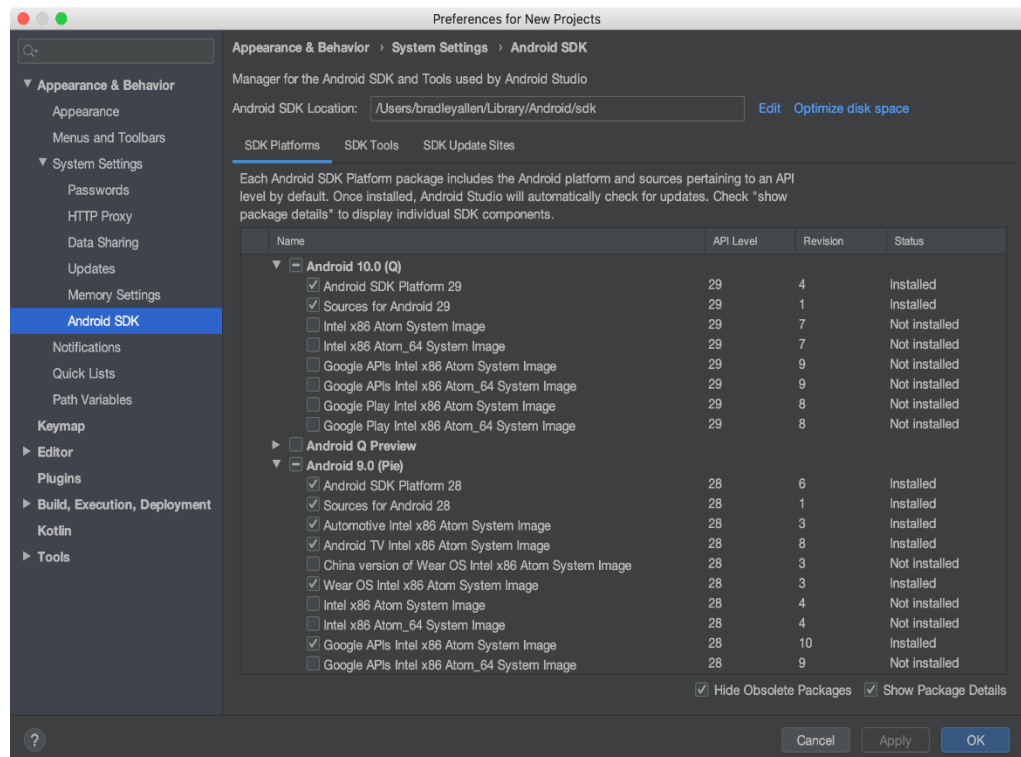


Рисунок 2.2 – Android SDK

Протягом AOSP було випущено багато різних версій Android для мобільних пристроїв. Коли виходять нові версії Android, власникам мобільних пристроїв дозволяється вибирати, чи оновити свою ОС. З кожною ітерацією Android розробникам стає доступним новий SDK, а на підтримувані пристрої додаються різні нові функції. Розробники програмного забезпечення повинні пам'ятати про застарілі функції попередніх версій під час розробки нових додатків [10].

З усіма цими версіями Android функції та зміни ОС призвели до багатого та орієнтованого на користувача досвіду. Пересічний користувач, який майже

нічого не знає про технічні аспекти пристрою, може дуже просто керувати ним [10].

2.3 Обґрунтування вибору середовища розробки

Середовище розробки, що використовується - IntelliJ IDEA. Для розробки мобільних додатків можна обирати між трьома середовищами розробки, а саме – Eclipse, IntelliJ IDEA та Android studio. Переваги серед Eclipse у тому, що вона є openSource, тобто кожен бажаючий може подивитися, з чого складається дана система, чи не має в ній потенційної загрози. Це є важливим для інформаційної безпеки. Але головною перевагою все ж є плагіни та спеціалізовані версії. Якщо ви Java розробник, що добре розуміє, якими інструментами потрібно користуватися для реалізації свого продукту, вам не потрібні стабільність та певні обмеження системи, написаної кимось іншим. Тоді кращим рішенням стає каркас, на якому досить легко створити середовище розробки під конкретну задачу. Те ж саме стосується мов програмування, що підтримуються. Крім тих мов, що підтримуються більшістю універсальних серед розробки (Java, C-подібні мови, Python та інші), Eclipse має засоби підтримки мов програмування, що рідко використовуються (GNAT bench, Hibiachi або Fortran). Отже, Eclipse - не середовище розробки у звичайному сенсі цього слова, а невеликий стандартний набір розширень, що можна доповнити самостійно [6].

IntelliJ IDEA називають “розумною системою” через такі риси:

- глибокий семантичний аналіз написаного коду та передбачення можливих варіантів продовження;
- автозаповнення;
- глибокий та безпечний рефакторинг;
- автоматично вигадує зручні ідентифікатори для змінних.

Система має свої недоліки та переваги.

Недоліки:

					ІАЛЦ.045490.004 ПЗ	Лист
						25
Зм	Лист	№ докум.	Підп.	Дата		

- не всі мови однаково добре підтримуються IDEA (для таких мов як Rust, Go, Haskell реалізовані тільки підсвічування синтаксису, без автозаповнення, переходу до визначення, рефакторингу);
- в деяких випадках IDEA працює дуже повільно. Наприклад, якщо в проекті є файл на 5000 рядків коду. В такому випадку потрібно розбити його на менші файли;
- IDEA споживає досить багато ресурсів.

Переваги:

- інтеграція с Git та іншими системами контролю версій, побудова діаграм класів, перехід по ієрархіям, декомпілятор байткоду [Java](#);
- потужний відладчик;
- автоформатування коду;
- редактор не потрібно доналаштовувати під конкретну мову.

Отже сенс IntelliJ IDEA - не інтерфейс та маса розширень, а інтелект в роботі із Java.

2.4 Обґрунтування вибору мови програмування

Java – універсальна об'єктно орієнтована мова програмування, що була розроблена компанією Sun Microsystems, наразі власником технології якої є компанія Oracle Corporation. Ця платформа здебільшого призначена для розробки та запуску Desktop-застосунків. Сучасний набір інструментів Java робить можливими проектування та розробку багатовіконних складних кросплатформених додатків, що не поступаються класичним застосункам по користувацьким характеристикам [8].

Основні переваги платформи Java:

- можливість запуску застосунків на більшості існуючих операційних систем;
- висока надійність та безпека;

- переносимість ;
- висока продуктивність;
- автоматичне керування виділенням пам'яті;
- проста та зручна розробка web-застосунків;
- можливості фільтрації вводу/вивода;
- засоби створення багатопоточних додатків;
- доступ до реляційних баз даних за допомогою JDBC — Java DataBase Connectivity.

Java розробникам програмного забезпечення пропонується такі переваги, що допомагають розробляти додатки для рішення бізнес задач майже будь-якої складності:

- кросплатформеність;
- зручність при розробці;
- надійність функціонування;
- ефективна система безпеки.

Також слід зауважити, що при створенні компонентів UI використовується мова маркування XML [9]. Це має такі переваги:

- дизайнери можуть створювати ці макети без навичок програмування на Java;
- конструкція XML-макета проста для розуміння;
- можливість збереження користувацького інтерфейсу окремо від коду застосунку;
- досить при розробці додатку, що використовує багато мов;
- посилання на рядки, розміри теми, креслення.

2.5 Обґрунтування вибору системи керування базами даних

За замовчуванням в Android використовується реляційна база даних SQLite, що підтримує такі типи: INTEGER (аналог long в Java), TEXT (аналог String в Java), REAL (аналог double в Java). Інші типи перед збереженням в базі

					ІАЛЦ.045490.004 ПЗ	Лист
						27
Зм	Лист	№ докум.	Підп.	Дата		

потрібно конвертувати. Бібліотека Android містить в собі абстрактний клас SQLiteOpenHelper. За допомогою нього можна створювати, відкривати та оновлювати бази даних.

Серед переваг SQLite можна виділити:

- файлова структура - база даних складається з одного файлу, тому її досить легко переносити на інші машини
- немає необхідності налаштування сервера СУБД;
- вільна ліцензія;
- кросплатформеність;
- висока швидкість простих операцій;
- підтримка транзакцій, тригерів, представлень (views), вкладених запитів;
- БД зберігається в одному файлі, права доступу можна контролювати стандартними засобами ОС;
- економічна за ресурсами.

Недоліки:

- відсутність системи управління правами доступу користувачів;
- відсутність можливості підвищення продуктивності.

Отже, SQLite в своїх розробках використовувати доцільно лише у випадку врахування особливостей цієї БД. Добре підходить для проектів, що містять мало операцій записів, мають обмежені ресурси сервера, не потрібна система прав доступу до БД [7].

2.6 Обґрунтування вибору способу технічної розробки

З кожним роком збільшується кількість користувачів мобільних пристроїв. Отже, стає актуальним питання вибору способу технічної розробки і реалізації проектів на мобільних пристроях. Існують два варіанти: розробка мобільного застосунку та розробка мобільного web сайту. Звісно, оптимальний варіант –

реалізація обидвох варіантів, але якщо існують обмеження у бюджеті, то потрібно провести аналіз необхідних функцій, цільової аудиторії та вартості розробки. Після цього, зваживши всі плюси та мінуси, робиться вибір на користь одного із способів.

Мобільна версія сайту відрізняється від звичайної тим, що вона адаптується під всі екрани мобільних пристроїв. На перший план відводиться зручність користувачів у взаємодії із контентом на сайті або інтернет магазині. Також потрібно враховувати, що мобільна версія сайту дуже важлива при постійному збільшенні числа користувачів мобільних телефонів. Переваги та недоліки даного способу наведено нижче.

Переваги:

- вартість мобільного сайту включається у вартість створення дизайну сайту із самого початку;
- для мобільного сайту не потрібно створювати під різні платформи новий сайт, оскільки він однаково працює на всіх мобільних пристроях;
- мобільний сайт не потрібно заздалегідь викачувати в магазини.

Недоліки:

- сайт із поганим адаптивним дизайном може негативно відобразитися на швидкості роботи сайту та користуванні;
- відсутність доступу в оффлайн режимі;
- Відсутність push повідомлень.

При всьому цьому існує безліч сайтів, що мають дуже погане адаптування під мобільні пристрої. Також слід зазначити, що реклама, не потрібний контент та спливаючі вікна - не кращі помічники під час створення зручного інтернет магазину. Також слід уникати складних структур та дратуючих анімацій заради швидкості сайту.

Мобільні застосунки мають багато корисних для продажу функцій, які відсутні у мобільному сайті. Тут також існують свої переваги і недоліки:

- геолокація, щоб мати зв'язок із найближчим офісом компанії;

- пуш-повідомлення для інформування людей щодо поточного статусу їх замовлення;
- камера для сканування коду, щоб отримувати інформацію про продукт;
- доповнена реальність, щоб розширити можливості;
- ідентифікація за допомогою відбитку пальця, щоб позбавити клієнтів від постійної необхідності вводу логіна та пароля.

Недоліки:

- додаткові витрати на розробку;
- додаткові налаштування. Перед тим, як користувач зможе завантажити мобільний додаток із магазину, його потрібно там розташувати.

На відміну від мобільного сайту, мобільний застосунок більше привертає увагу клієнтів та допомагає скоріше слідкувати за змінами в онлайн магазині. Але мобільний застосунок, на жаль, потребує більших витрат часу та грошей.

Отже, клієнти, що хочуть використовувати вбудовані функції сучасних смартфонів, щоб зробити різноманітнішим користувацьких досвід, обирають мобільні додатки. Якщо мета - залучити до онлайн магазину нових клієнтів, тоді краще рішення - мобільна версія сайту.

Висновок до другого розділу

У даному розділі проаналізовано різні операційні системи та обрано операційну систему для розробки мобільного додатку. Серед технологій розглянуто переваги та недоліки Android порівняно з IOS. Обрано Android через кросплатформенність, відкритість коду, високу якість написання документації та велику онлайн спільку для розробників. Під час вибору середовища програмування розглядалися Android Studio, IntelliJ IDEA, Eclipse. Обрана IntelliJ IDEA через наявність зручних інструментів для розробки Android додатків. Серед мов програмування обрана Java, серед баз даних SQLite через кросплатформенність та високу швидкість виконання простих операцій.

3. СТРУКТУРА І АЛГОРИТМ РЕАЛІЗАЦІЇ ПРОГРАМИ

3.1 Загальний опис системи

Система, що розроблялась, є Android застосунком для вивчення англійської мови.

Основна мета даного проєкту - вирішення проблем, що виявлені в існуючих на сьогоднішній день додатках, представлених у Play Market. Виходячи з першого розділу, зроблені висновки, що система повинна бути наділена такими характеристиками:

- наявність ігрової концепції вимірювання прогресу;
- порядок викладення матеріалу повинен відбуватися від простого до складного;
- граматичні правила повинно бути подано схематично, а не суцільним текстом;
- завдання повинні бути різноманітні;
- користувацький інтерфейс повинен бути адаптивним, тобто працездатним на екранах із різним розширенням;
- повинні бути наявними світла та темна теми, між ними можливе перемикання.

Додаток повинен складатися із:

- головного екрану;
- екрану налаштування;
- екрану вибору рівня;
- екрану для кожного окремого рівня.

Головний екран

Після натиснення на іконку додатку на робочому столі смартфона, Головний екран відкривається за замовчуванням. На цьому екрані розміщується

кнопка переходу до Екрану вибору рівня. В правому верхньому куті розміщена кнопка, натиснувши яку користувач переадресовується на Екран налаштування, в лівому - кнопка вибору між світлою і темною темами.

Екран вибору рівня

На цьому екрані представлено можливість переходу між рівнями.

Екран для кожного окремого рівня

Структура екрану кожного рівня є унікальною. Наприклад, для першого рівня було створено п'ять блоків. На кожному з них розміщено слово. Користувач має скласти із наявних слів граматично правильне речення. Перед кожним рівнем з'являється діалогове вікно, на якому схематично зображене граматичне правило, успішне засвоєння якого гарантує проходження рівня. Зверху знаходиться шкала прогресу, після кожного правильно зібраного речення, вона заповнюється на одну позицію. Якщо користувач робить помилку, гра починається з початку.

Під час розробки програмного забезпечення було використано сучасні технології, що стало основною причиною швидкої розробки системи.

3.2 Архітектура системи

В більшості випадків додатки мають єдину точку входу з робочого стола або програми запуску, а потім запускаються як єдиний монолітний процес. З Android застосунками справи склалися інакше. Вони мають більш складну структуру. Типовий Android додаток включає в себе декілька компонентів, в які входять Activities, Fragments, Services, ContentProviders та BroadcastProviders.

Потрібно оголосити деякі з цих компонентів або всі у маніфесті додатку. Після цього ОС Android використовує цей файл, щоб вирішити, як інтегрувати застосунок у загальний користувацький інтерфейс пристрою. Враховуючи, що правильно написаний додаток включає в себе декілька компонентів і користувачі часто взаємодіють з декількома додатками за короткий проміжок часу, додатки

повинні адаптуватися до різних типів робочих процесів і задач, якими керує користувач.

Найважливішим принципом для слідування є розподілення відповідальності. Класи користувацького інтерфейсу повинні включати в себе тільки логіку, що буде обробляти взаємодію користувацького інтерфейсу та операційної системи. Слідуючи цьому правилу, розробник уникає проблем, що пов'язані з життєвим циклом додатку.

Інший важливий принцип полягає в тому, що потрібно керувати своїм користувацьким інтерфейсом із моделі, а краще – із постійної моделі. Моделі являють із себе компоненти, що відповідають за обробку даних для застосунку. Вони не залежать від об'єктів View і компонентів додатку, тому на них не впливає життєвий цикл застосунку та пов'язані із ним проблеми.

Постійна модель є ідеальною за такими причинами:

- користувачі не втратять дані, якщо ОС знищить додаток, щоб звільнити ресурси;
- застосунок продовжить працювати в тих випадках, коли мережеве з'єднання нестабільно або недоступно.

Займаючись організацією основи застосунку на модельних класах із чітко визначеною відповідальністю по управлінню даними, додаток стає легшим в тестуванні і підтримці.

Неможливо мати один спосіб написання додатків, що краще за все підходить для кожного сценарію. При цьому дана архітектура являє із себе гарну відправну точку для більшості ситуацій та робочий процесів (рис. 3.1).

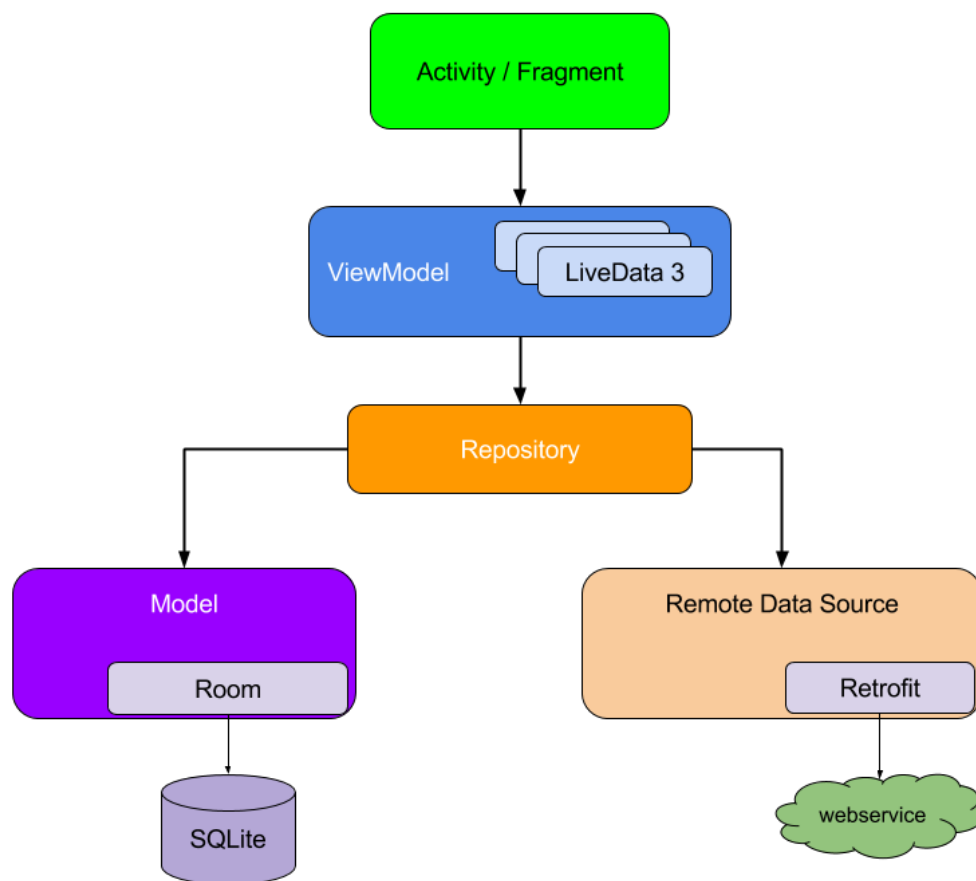


Рисунок 3.1 – Схема взаємодії модулів готової архітектури

Кожний компонент залежить тільки від компонента на один рівень нижче його. Наприклад, Activity та Fragments залежать лише від моделі представлення. Єдиним класом, що залежить від багатьох інших, є Repository. В даному прикладі сховище залежить від постійної моделі даних та віддаленого внутрішнього джерела даних.

Цей дизайн паттерн створює користувацький послідовний і приємний досвід. Незалежно від того, чи повернеться користувач до застосунку через декілька хвилин після його закриття чи через декілька днів, він миттєво побачить інформацію про те, що застосунок зберігається локально. Якщо дані застарілі, то модуль сховища додатку починає оновлювати дані в фоновому режимі.

Дане програмне забезпечення включає в себе такі компоненти:

- база даних;
- клієнт.

Перед тим, як починати розробляти даний програмний продукт, існувала мета створити мобільний додаток, операційна система якого буде здатна охопити велику частину ринку, та той, в якому буде можливим легке модифікування в майбутньому. Тому в даному випадку операційною системою обрано Android, мовою програмування - Java.

Проект має таку структуру: коренева папка містить в собі папки src, в якій знаходиться вихідний код додатку, що включає в себе файл AndroidManifest, що вказує точку входу в додаток, задає загальну інформацію про додаток, визначає атрибути додатку, здійснює запит прав доступу до певних критичних операцій; res, де знаходяться директорії - ресурси додатку(іменування кожної папки визначає тип ресурсів, які в ній повинні міститися. Наприклад, res/values - для рядкових ресурсів, res/layout - для файлів компонування екрана, res/layout - для файлів зображень); gen - містить java-файли, що створюються автоматично під час розробки додатку(ці файли не можна редагувати самостійно); Main.java - файли вихідного коду, де описаний клас головної Activity.

3.3 Опис розробленого додатку

Робота Android застосунку, розробленого під час дипломного проектування, реалізована через декілька активностей.

3.3.1 activity_main

Розмітка activity_main активності знаходиться у файлі activity_main.xml (рис. 3.2). Дана активність являє собою точку входу у додаток. Нижче наведено її зовнішній вигляд:



Рисунок 3.2 – Головний екран

При натисненні на кнопку Start відбувається перехід на сторінку вибору рівнів. Подія натиснення на дану кнопку оброблюється у java файлі наступним чином:

```
Button buttonStart = (Button)findViewById(R.id.button_start);
buttonStart.setOnClickListener(new View.OnClickListener(){
    @Override

    public void onClick(View v){
        try{
            Intent intent = new Intent(MainActivity.this, GameLevels.class);
            startActivity(intent); finish();

        } catch(Exception e){

        }
    }

});
```

Це відбувається у методі onCreate, що викликається при створення або перезапуску активності. У методі можливо існує можливість зробити перевірку, чи запускається додаток вперше. Якщо значення змінної savedInstanceState дорівнює null, то додаток запускається вперше.

```
if ( savedInstanceState == null )  
{  
    ...  
}  
else  
{  
    ...  
}
```

3.3.2 game_levels

Дана активність відповідає за вибір рівня і має такий вигляд (рис. 3.3):



Рисунок 3.3 – Екран вибору рівня

При натисненні кнопки Back користувач повертається на активність, описану вище. Кнопки 1,2,3... ведуть на активності, що відповідають певним рівням. Також при натисненні на кнопку back на смартфоні (рис. 3.4)



Рисунок 3.4 – Зображення кнопки Back

також відбувається повернення до активності main_activity. За це відповідає метод onBackPressed:

```
@Override
public void onBackPressed(){

    try {

        Intent intent = new Intent(GameLevels.this, MainActivity.class);
        startActivity(intent); finish();
    } catch(Exception e){

    }

}
```

3.3.3 first level

Основною ціллю в навчанні граматики є формування у учнів граматичних навичок, що є одним з найважливіших компонентів у розмовному мовленні, аудіюванні, читанні та письмі.

Слово *граматика* багатозначне. З одного боку воно вживається позначаючи «граматичний устрій», тобто побудову слова та речення, що належать певній мові, яким носій володіє інтуїтивно и який він усвідомлює в

процесі навчання в школі. З іншого боку, граматика є розділом мовознавства, де вивчаються закономірності зміни та поєднання слів, що утворюють речення та вирази, які мають зміст.

Без володіння граматикою в першому значенні неможливе усне спілкування.

Вміння грамотно поєднувати слова, змінювати вирази в залежності від того, що ви хотіли би сказати в певний момент часу, є однією з найважливіших умов використання мови як засобу спілкування.

Оволодіння граматикою мови, що вивчається, важливе не тільки для формування продуктивних вмінь в усному та писемному мовленні, але й для розуміння мовлення інших людей при аудіювання і читанні.

Можна грамотно побудувати власне висловлювання, використовуючи достатньо обмежений набір граматичних конструкцій. Однак це зовсім не гарантує того, що люди не будуть використовувати більш складні структури у власному мовленні. Це може стати досить вагомою перешкодою для розуміння суті висловлювань, не кажучи вже о соціолінгвістичних рисах, що виражаються у можливому підтексті того, що було сказано. У разі відсутності повноцінного розуміння, відсутнє і повноцінне спілкування. У такому разі недостатній рівень граматичних навичок переростає у бар'єр на шляху формування мовної компетенції.

При вивченні іноземної мови на етапі формування навички слід звертати увагу на підготовчі вправи. Одноманітність в організації вправ явно вказує на нерозуміння основних типів та видів вправ, направлених на автоматизацію матеріалу. Слід звертати увагу також на класифікацію вправ по діям, що виконуються учнями. Можна виділити вправи на відновлення, підстановку, трансформацію, підкреслення, угруповання, альтернативний/множинний вибір тощо. Такі дії є некомунікативними, але аспектними, тренувальними, одномовними. Можуть бути одно- або багатоцільовими. Такі вправи добре слугують для якісного засвоєння форм мовних явлень. Вищеназвані вправи

допомагають додати різноманітності учбовому процесу, а також уникнути монотонності, підвищити мотивацію, градувати труднощі.

Перший рівень слугує прикладом вправ із використанням підстановки, що скеровані на освоєння форми мовного взірця та тренування лексики за рахунок аналогій. Вправа повинна виконуватися у швидкому темпі та без перекладу, містить в собі всього одну операцію, тому є досить легкою та може бути використана на підготовчому етапі формування навичок одразу після імітативних. До переваг відносять простота виконання та організації, тренування мовних взірців, виконаних у різних режимах (і в ігровій формі також). До недоліків можна віднести механічність роботи.

Зовнішній вигляд активності, що відповідає першому рівню, наведено нижче (рис. 3.5):

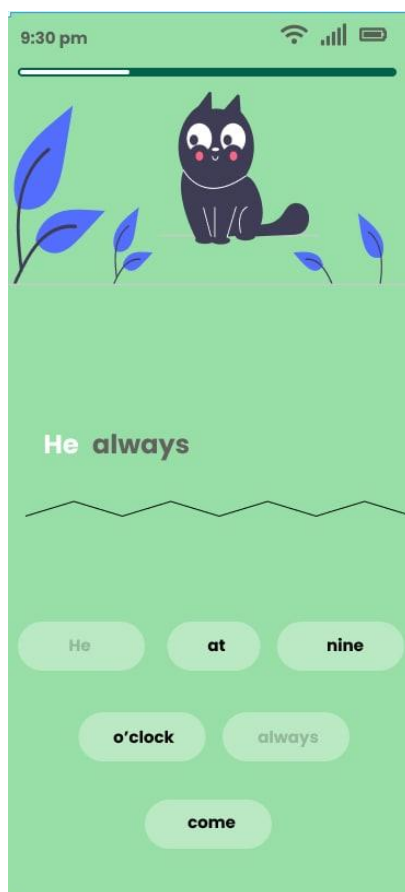


Рисунок 3.5 – Екран першого рівня

Алгоритм гри виглядає наступним чином: користувачу потрібно із наведених блоків скласти граматично правильне речення. Таку послідовність дій потрібно виконати декілька разів. У цей час буде відбуватися заповнення шкали прогресу. У разі допущення користувачем хоча б однієї помилки, рівень починається спочатку і супроводжується повідомленням вигляду (рис. 3.6):



Рисунок 3.6 – Приклад вікна, що виникає у разі поразки

При натисненні на кнопку Continue користувач переміщується на початок першого рівня.

Перед кожним рівнем з'являється вікно зі структурним зображенням граматичного правила, на якому буде побудований ігровий процес даного блока. Воно має такий зовнішній вигляд (рис. 3.7):

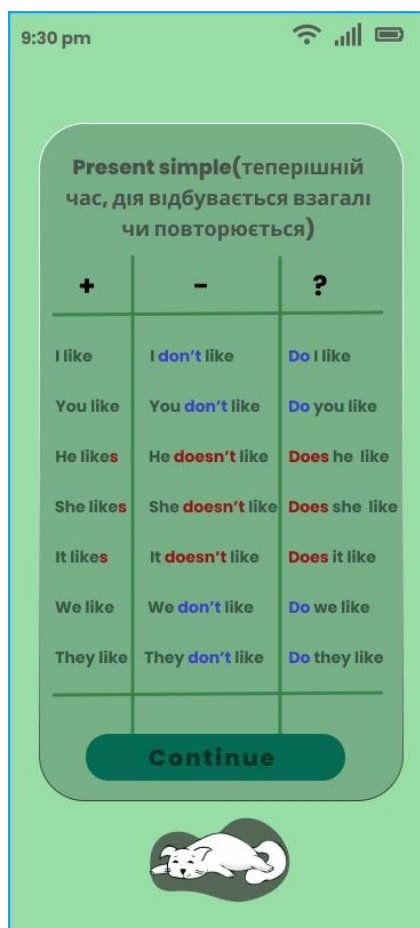


Рисунок 3.7 – Вікно із модальним вікном з правилом

3.3.3 second_level

Другий рівень є прикладом трансформаційної вправи. Такий вид вправ опирається на механізм трансформації, (синонімічні заміни як уточнення, корекція граматичного оформлення, викладу чужого мовлення) і тому є одномовними. В учбовому процесі успішно слугують для освоєння форми явищ. Також до даного виду можна віднести вправи на розкриття дужок. Слід пам'ятати, що воно повинно виконуватися без перекладу. При розробленні основної структури вправи викладач повинен враховувати рівень лексики, якою володіє учень. Із трансформацією також пов'язані вправи на відмінювання дієслів. До переваг можна віднести те, що дана вправа є відмінним способом до формування мовної навички утворення дієслівної форми, так як стереотипуються зв'язки відмінювання у певному порядку.

Нижче наведено скріншот екрану другого рівня (рис. 3.8):

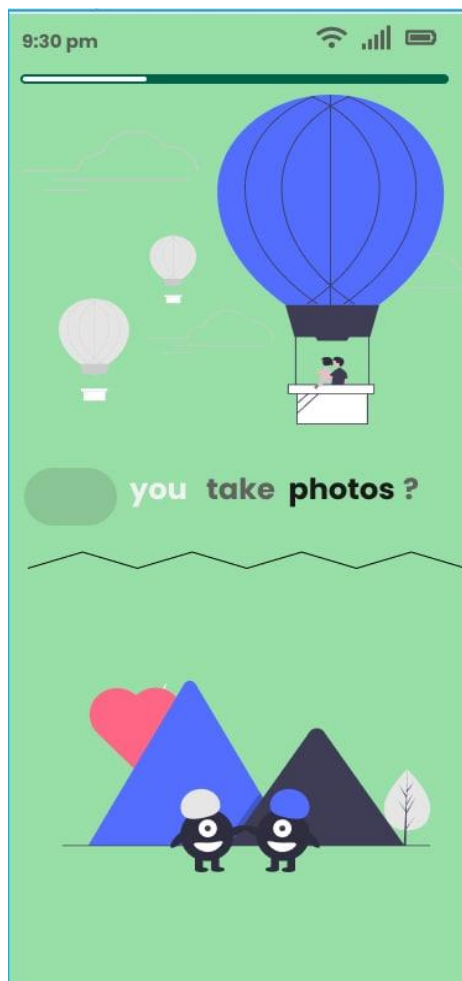


Рисунок 3.8 – Другий рівень

Алгоритм гри полягає в наступному: користувачу на екрані представлене речення, в якому відсутнє одне слово. На місці, виділеному фігурою, може бути розміщене або одне із допоміжних дієслів часу Present Simple do/does, або їх комбінація із частинкою not, або порожню місце. Користувачу потрібно підібрати слово, яке потрібно вставити у речення в залежності від контексту. Подібні дії потрібно повторити декілька разів для доведення до автоматизації. У разі допущення помилки користувач починає рівень спочатку, отримуючи повідомлення про невдачу та можливість ще раз ознайомитися із правилами. У разі успішного проходження користувач отримує привітання із успішним проходженням рівня, яке має такий вигляд (рис. 3.9):

Зм	Лист	№ докум.	Підп.	Дата

ІАЛЦ.045490.004 ПЗ

Лист

43



Рисунок 3.9 – Повідомлення про успішне проходження рівня

3.3.4 third level

Третій рівень є прикладом вправи на поєднання/пошук співвідношень.

Корпус вправи виглядає як два стовпця або дві групи елементів різного рівня, між якими потрібно встановити співвідношення. Завдання багатофункціональне і добре підходить для використання із досить різними цілями. Може бути ускладнене, якщо корпус вміщує в себе зайві елементи.

Алгоритм рівня відповідає такій схемі: перед користувачем представлені три зображення, під кожним з яких знаходиться речення. Речення подаються у різних формах часу Present Simple (питальній, заперечній і стверджувальній). Користувачу потрібно співвіднести відповідне речення із відповідною картинкою.

Інтерфейс даного рівня має такий вигляд (рис. 3.10):

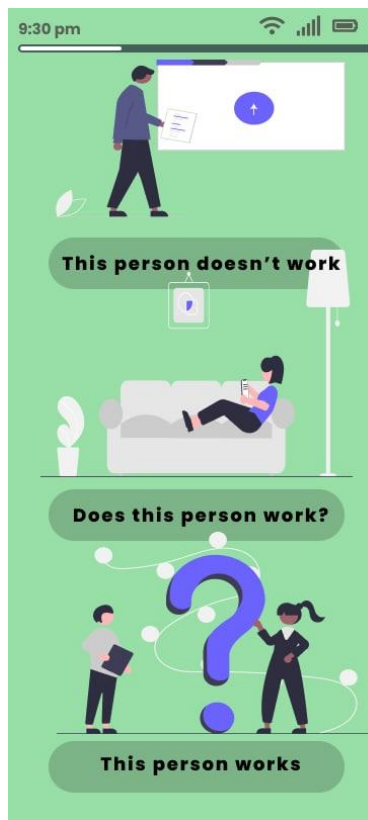


Рисунок 3.10 – Екран третього рівня

Як і в попередніх рівнях, користувачу потрібно повторити успішне виконання даного алгоритму декілька разів для кращого засвоєння. У іншому випадку йому потрібно буде почати рівень з початку.

Висновок до третього розділу

В даному розділі було проілюстровано створений у результаті дипломного проєктування мобільний додаток. У результаті виправлені недоліки, знайдені у першому розділі у існуючих аналогах даного додатку. А саме:

- реалізовано вимірювання прогресу;
- реалізовано поступовий перехід від простого до складного;
- аодача граматичних правил відбувається схематизовано, що робить легшим і комфортнішим сприймання;
- однотипність і нерізноманітність завдань, що призводить швидкої втрати інтересу користувачів, усунено.

4. МОЖЛИВІ ВАРІАНТИ РОЗВИТКУ

У ході розробки Android додатку для даного дипломного проєкту реалізовано не всі види вправ для того, щоб вирішити такі проблеми як недостатня кількість вправ, нераціональний час для виконання вправ, одноманітні режими і форми виконання, низька мозкова активність учнів у ході роботи, нереалізований перехід від простого до складного тощо. Для того, щоб остаточно вирішити дані проблеми, непоганим рішенням стало б розширення кількості рівнів та додання ще більшої різноманітності.

Наприклад, можна додати вправу на складання/відновлення, що передбачає складання цілого з елементів, відновлення правильного, логічного порядку із запропонованих елементів. Одиниці корпусу вправи можуть бути різного рівня: складання слова з букв використовуються для формування лексичних та орфографічних навичок; складання речення із заданих слів використовуються для формування синтаксичних навичок; складання тексту/діалогу із заданих фрагментів використовується при навчанні експресивним видам мовної діяльності, для контролю розуміння прочитаного.

Також можливе додавання вправи на розширення. Існує два основних варіанта завдань на розширення:

- при навчанні техніці читання воно має вигляд трапеції та слугує для розширення поля зору та зняття труднощів при переході від читання слів до читання речень;
- при навчанні експресивним видам мовної діяльності вони використовуються в якості підготовчих вправ різного рівня: на розширення речень (за рахунок прикметників, додатків, однорідних членів речення тощо) або тексту (як опора для усного або письмового монологу).

Можливе додання вправи на доповнення, що використовується для формування та контролю лексичних навичок. Облегшений варіант такої вправи включає в себе перелік лексики для підстановки. Завдання не потребує

перекладу. Правильний вибір лексичної одиниці свідчить про розуміння її значення, тому основними вимогами перед виконання даної вправи є актуалізація лексики та зняття фонетичних труднощів.

Більшість вправ можуть мати модифікації різного рівня складності. Варіанти ускладнень наведено в таблиці 4.1.

Таблиця 4.1 – Варіанти підвищення складності для проаналізованих вправ

Вид вправи	низький	середній	високий
Відкриття дужок	Підставити в дані особи/часу	Поставити в необхідну форму, обравши перед цим із двох часів при наявності маркерів	Поставити в необхідну форму, обравши перед цим із трьох варіантів (наприклад, при наявності декількох тлумачень в залежності від ситуації)
Підстановка в речовий зразок	Із зоровою опорою на мовний зразок	Без опори на зразок	Підстановка, що потребує розширення/комбінування мовного зразка
Підстановка в таблицях	Повна відповідність	З вибірковою відповідністю	З елементами трансформації
Вставити в пропуски	Одиниці дано в потрібній формі	Дано однокореневі одиниці, потрібна трансформація	Одиниці не дано
Вправи на відмінювання	Відмінювання хором за викладачем, читання по	Відмінювання з опорою на схематичну	В будь-якому порядку, усно, в швидкому темпі, із зоровою опорою на інфінітив або без опори

	таблиці відмінювання	таблицю закінченнями	з	
--	-------------------------	-------------------------	---	--

Отже, також можливе запровадження різних рівнів складності в перспективі.

Висновки до четвертого розділу

Отже, програма може бути розширена шляхом додавання нових рівнів із вправами, в яких будуть задіяні інші методики. Це допоможе уникнути таких проблем:

- одноманітні режими та форми виконання вправ;
- тривалість виконання однієї вправи;
- недостатня кількість вправ в системі;
- врахування рівня складності;
- неграмотний підбір вправ;
- низька мозкова діяльність учнів під час роботи;
- недостатня кількість умовно-комунікативних та комунікативних вправ у системі роботи.

Ці проблеми вирішить впровадження вправ за такими принципами:

- складання/відновлення;
- розширення;
- доповнення.

ВИСНОВКИ

В дипломному проєкті розроблено мобільний додаток для вивчення граматики англійської мови.

У першому розділі проведено порівняння трьох популярних додатки та з'ясовано, що в усіх додатках реалізовано цікаву ігрову концепцію та вимірювання прогресу; не зважаючи на те, що розробники пропонують свої додатки для широкої аудиторії, LinguaLeo та Duolingo є більш цікавими для дорослої та підліткової аудиторії, тоді як Simpler може використовуватися у молодшій та середній школах; у всіх трьох додатках реалізовано блочну подачу матеріалу, проте поступового переходу від простого до складного найбільше дотримуються розробники Simpler; в усіх трьох додатках подача граматичних правил відбувається несхематизовано, що ускладнює сприймання; однотипність і нерізноманітність завдань є основним недоліком усіх трьох проаналізованих додатків, що призводить швидкої втрати інтересу користувачів.

В другому розділі обрано операційну систему для розробки мобільного додатку. Серед технологій розглянуто переваги та недоліки Android порівняно з IOS. Обрано Android через кросплатформенність, відкритість коду, високу якість написання документації та велику онлайн спільку для розробників. Під час вибору середовища програмування розглядалися Android Studio, IntelliJ IDEA, Eclipse. Була обрана IntelliJ IDEA через наявність зручних інструментів для розробки Android додатків. Серед мов програмування обрана Java, серед баз даних SQLite через кросплатформенність та високу швидкість виконання простих операцій.

В третьому розділі проілюстровано створений у результаті дипломного проєктування мобільний додаток. У результаті виправлені недоліки, знайдені у першому розділі у існуючих аналогах даного додатку. А саме:

- реалізовано вимірювання прогресу;
- реалізовано поступовий перехід від простого до складного;
- подача граматичних правил відбувається схематизовано, що робить

легшим і комфортнішим сприймання;

					ІАЛЦ.045490.004 ПЗ	Лист
						50
Зм	Лист	№ докум.	Підп.	Дата		

- однотипність і нерізноманітність завдань, що призводить швидкої втрати інтересу користувачів, було усунено.

В четвертому розділі зпоказано, що програма може бути розширена шляхом додавання нових рівнів із вправами, в яких будуть задіяні інші методики, окрім тих, що були реалізовані під час розробки даного мобільного застосунку.

Слід зазначити, що даний проєкт охоплює лише базовий функціонал для вивчення граматики англійської мови, проте він вже готовий для стартового використання.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРИНИХ ДЖЕРЕЛ

1. Lingualeo.ru – английский язык – отзывы [Электронный ресурс]. Режим доступа: <https://irecommend.ru/content/lingualeo>
2. Lingualeo.ru - изучение английского языка – отзывы [Электронный ресурс]. Режим доступа: https://otzovik.com/reviews/lingualeo_ru-izuchenie_angliyskogo_yazika/
3. Отзывы о сервисе Duolingo [Электронный ресурс]. Режим доступа: <https://startpack.ru/application/duolingo-learn-language/reviews>
4. Duolingo – учим языки бесплатно – отзывы [Электронный ресурс]. Режим доступа: <https://irecommend.ru/content/duolingo-uchim-yazyki-besplatno>
5. Сравнение Duolingo и Lingualeo [Электронный ресурс]. Режим доступа: <https://startpack.ru/compare/duolingo-learn-language/lingualeo-learn-language>
6. Simpler – программа для Android – отзывы [Электронный ресурс]. Режим доступа: https://otzovik.com/reviews/simpler-programma_dlya_android/
7. Android [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/Android>
8. Java [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/Java>
9. XML [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/XML>
10. History and evolution of the Android OS [Электронный ресурс]. Режим доступа: https://link.springer.com/chapter/10.1007/978-1-4302-6131-5_1