

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

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

**Індивідуальний дослідницький проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інтегровані інформаційні системи»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Мобільний додаток для швидкого читання «SpeedRead»»**

Виконав:

студент IV курсу, групи ІА-81

Шіхерт Антон Анатолійович _____

Керівник:

Асистент кафедри ІСТ,

Шинкевич Микола Костянтинович _____

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

Студент _____

Київ – 2022 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАВДАННЯ

на індивідуальний дослідницький проєкт студенту

Шіхерту Антону Анатолійовичу

1. Тема проєкту «Мобільний додаток для швидкого читання «SpeedRead»», керівник проєкту Шинкевич Микола Костянтинович.
2. Термін подання студентом проєкту: 15 червня 2022 року
3. Вихідні дані до проєкту: мобільний додаток для швидкого читання, методиками швидкого читання, з реалізацією хмарних сервісів.
4. Зміст пояснювальної записки: аналіз предметної області, аналіз існуючих рішень, вибір технологій та реалізації, реалізація користувацького інтерфейсу та програмної частини мобільного додатку з подальшим тестуванням.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): діаграма діяльності, діаграма прецедентів, діаграма розгортання, діаграма послідовностей.
6. Дата видачі завдання 1 грудня 2021 року

Календарний план

№ з/п	Назва етапів виконання проєкту	Термін виконання етапів проєкту	Примітка
1	Аналіз предметної області	09.05.2022	
2	Аналіз існуючих рішень	16.05.0222	
3	Попередня підготовка текстових матеріалів	23.05.2022	
4	Реалізація користувацької частини	25.05.2022	
5	Реалізація програмної частини	30.05.2022	
6	Оформлення документації дослідницького проєкту	06.06.2022	

Студент

Антон ШІХЕРТ

Керівник

Микола ШИНКЕВИЧ

АНОТАЦІЯ

Шіхерт А.А. Мобільний додаток для швидкого читання «SpeedRead». КПІ ім. Ігоря Сікорського, Київ, 2022.

Проект містить 63 сторінки тексту, 35 рисунків, 2 таблиці, 22 посилання на 20 джерел.

Ключові слова: мобільний додаток, методи читання, швидке читання, операційна система «Android», методика, техніка.

Об'єктом розробки є мобільний додаток для швидкого читання.

Мета розробки – збільшити коефіцієнт корисної вартості читання текстового матеріалу.

В результаті виконання дипломного проєкту поставлена мета була досягнута, мобільний додаток дає змогу читати текстовий матеріал з швидкістю в декілька разів вищою ніж середня швидкість читання тексту людиною, за дослідженою статистикою. Під час користування додатком можна написати текст, зберегти його у пам'ять пристрою, завантажити текстовий документ з пристрою, та прочитати його однією з запропонованих технологій швидкого читання.

SUMMARY

Schichert A.A. Mobile application for fast reading «SpeedRead» Igor Sikorsky KPI, Kyiv, 2022.

The project contains 63 pages of text, 35 figures, 2 tables, 22 references to 20 sources.

Keywords: mobile application, reading methods, speed reading, Android operating system, methods, techniques.

The object of development is a mobile application for fast reading.

The object of the development is to increase the coefficient of useful value of reading text material.

As a result of the diploma project, the goal was achieved, the mobile application allows you to read text material at a speed several times higher than the average speed of reading a person, according to statistics. When using the application, you can write text, save it in the device memory, download a text document from the device, and read it using one of the suggested speed reading technologies.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IA81.300БАК.003 ПЗ	Пояснювальна записка	63		
6	A3	IA81.300БАК.003 Д1	Мобільний додаток для	1		
7			швидкого читання			
8			«SpeedRead».Діаграма			
9			розгортання.			
10	A3	IA81.300БАК.003 Д2	Мобільний додаток для	1		
11			швидкого читання			
12			«SpeedRead».Діаграма			
13			прецедентів.			
14	A3	IA81.300БАК.003 ДЗ	Мобільний додаток для	1		
15			швидкого читання			
16			«SpeedRead».Діаграма			
17			діяльності.			
18	A3	IA81.300БАК.003 Д4	Мобільний додаток для	1		
19			швидкого читання			
20			«SpeedRead».Діаграма			
21			послідовностей.			
22						
23						
24						
25						
26						
27						
28						

					IA81.300БАК.003 ТП		
Зм.	Аркуш	№ докум.	Підпис	Дата			
Розроб.		Шіхерт А.А.			Літ.	Аркуш	Аркушів
Керівн.		Шинкевич М.К.			Т	1	63
Затв.					КПІ ім. Ігоря Сікорського Група ІА-81		
					Мобільний додаток для швидкого читання «SpeedRead». Відомість проєкту		

**Пояснювальна записка
до індивідуального дослідницького проєкту
на тему: «Мобільний додаток для швидкого
читання «SpeedRead»»**

Київ – 2022 року

ЗМІСТ

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ.....	4
ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Типи читання	7
1.1.1 Загальне читання.....	7
1.1.2 Ретельне читання.....	7
1.1.3 Швидкий перегляд	8
1.1.4 Діагональне читання	8
1.2 Теорія швидкого читання.....	9
1.3 Методики швидкого читання.....	9
1.3.1 Методика швидкого читання «по діагоналі»	9
1.3.2 Безперервне читання.....	10
1.3.3 Придушення субвокалізації	10
1.3.4 RSVT.....	11
1.3.5 Техніка читання Spritz	12
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	15
2.1 Мобільний додаток: «Reedy».....	15
2.2 Мобільний додаток «Speed reading»	16
2.3 Мобільний додаток «Speed Reader».....	18
2.4 Мобільний додаток «Balto Speed Reading»	20
2.5 Порівняльна характеристика існуючих рішень	21
3 ВИБІР ТЕХНОЛОГІЙ ТА РЕАЛІЗАЦІЇ	23
3.1 Засоби проектування.....	23
3.1.1 Сервіс для розробки користувацького інтерфейсу «Figma»	23
3.1.2 Сервіс для побудови діаграм «diagrams.net».....	24

					IA81.300BAK.003 ПЗ		
Зм.	Лист	№ док.ум.	Підпис		Мобільний додаток для швидкого читання «SpeedRead». Пояснювальна записка		
Розробив	Шіхерт А.А.						
Перевірив	Шинкевич М.К.				Т	2	63
Затв.					КПІ ім. Ігоря Сікорського Група ІА-81		

3.2	Техніки швидкого читання у додатку	24
3.2.1	Техніка RSVP у додатку	24
3.2.2	Техніка Spritz у додатку	26
3.2.3	Власна техніка швидкого читання	28
3.2.4	Техніка читання «Вказівник»	30
3.3	Засоби розробки	31
3.3.1	Мова програмування Kotlin	31
3.3.2	Мова розмітки XML.....	32
3.3.3	Середовище розробки Android Studio	34
3.3.4	Постачальник хмарних послуг Firebase.....	35
3.3.5	Система автоматичного збирання Gradle	36
4	РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ	39
4.1	Користувацький інтерфейс додатку	39
4.1.1	Створення логотипу додатку	39
4.1.2	Проектування користувацького інтерфейсу	40
4.1.3	Реалізація користувацького інтерфейсу	41
4.2	Програмна частина додатку	51
4.2.1	Основна діяльність.....	51
4.2.2	Фрагмент вибору файлу	52
4.2.3	Фрагмент створення власного файлу.....	53
4.2.4	Діяльність додавання файлу з пам'яті пристрою	54
4.2.5	Фрагмент швидкого читання	55
5	ТЕСТУВАННЯ ТА АНАЛІЗ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	57
5.1	Адаптивність інтерфейсу	57
5.2	Тестування програмного забезпечення.....	57
5.2.1	Тестування засобами тестування Android Studio.....	57
5.2.2	Тестування ручним методом.....	58
	ВИСНОВКИ	60
	ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ

ОС – операційна система;

RSVT – швидке серійне візуальне представлення;

ORP – оптимальна позиція розпізнавання;

JVM – Віртуальна машина Java яка виконує байт-код Java;

Android – операційна система для смартфонів;

APK – Формат файлу Android додатку;

Google Play - магазин додатків від компанії Google;

XML (англ. Extensible Markup Language) - розширювана мова розмітки;

Юнікод - стандарт кодування символів, що включає знаки майже всіх письмових мов світу;

IDE (англ. Integrated development environment) - інтегроване середовище розробки;

Material Design – система інструкцій для створення користувацького інтерфейсу;

Діяльність (англ. Activity) - основний компонент у додатках операційної системи «Android»;

Фрагмент (англ. Fragment) – компонент візуального та логічного інтерфейсу додатку який може використовуватись багаторазово.

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		4

ВСТУП

Створюючи цей мобільний застосунок було взято до уваги розвиток сучасного технологічного світу та людини в ньому, адже потік інформації збільшується і люди адаптуються до цього, не є секретом що сучасне покоління, через великий обсяг інформації та її різноманітність має кращу техніку читання ніж старше покоління, але меншу концентрацію уваги до інформації що споживає.

В контексті цього питання, однією з цілей мобільного додатку є розвиток техніки швидкого читання, яка в свою чергу допомагає покращити якість сприйняття інформації. В наш час людину оточує безліч повідомлень і дуже важливо їх сприймати, та вчасно реагувати на них, тож потреба в скорочитанні наразі є невід'ємною для кожної людини, яка хоче комфортно існувати в сучасному світі.

Актуальність зумовлена тим, що зараз не існує такого мобільного додатку який поєднував би в собі якість та сучасність, який давав би змогу користувачеві користуватися перевагами швидкого методу читання у буденному житті та відповідав би усім затребуваним критеріям. А додатки які не відповідають таким критеріям або реалізують методики читання не у повному обсязі, негативно впливають на інтерес користувачів до теми швидкого читання.

Серед недоліків у існуючих додатках, можна виділити:

- недоліки додатку, що спричиняють його передчасне закриття;
- присутність відволікаючих факторів, що заважає читанню;
- застарілий інтерфейс;
- відсутність необхідного функціоналу для комфортного використання;
- непідйомну ціну для пересічного користувача.

Мета даного проєкту – створити мобільний додаток який даватиме змогу користуватися перевагами швидкого читання без спеціальних навичок,

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		5

розвивати техніку читання та спростити процес навчання цьому, зацікавити користувачів темою швидкого читання та загалом розвинути їх увагу до змісту прочитаного.

Завдання цього проєкту:

- огляд переваг та недоліків існуючих рішень. Дослідження предметної області, для визначення критеріїв за яким буде оцінюватися наш проєкт та його аналоги;
- на основі проведення досліджень існуючих рішень, визначити вимоги до проєкту;
- огляд технологій;
- реалізація додатку та аналіз рішень що були прийняті при розробці додатку, опис реалізації додатку та його компонентів;
- тестування додатку.

Дипломний проєкт складається з наступних розділів: вступ, аналіз предметної області, аналіз існуючих рішень, вибір технологій та реалізації, реалізація мобільного додатку, тестування та аналіз якості мобільного додатку, висновки, список використаних джерел із 20 найменувань, 1 додатку. Графічна частина включає 4 кресленики формату А3. Загальний обсяг 63 сторінки.

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		6

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Типи читання

Існують різні типи читання. Вони існують і поділені за логікою, яку ціль переслідує читач. Типи читання залежать від мети та характеру сприйняття тексту.

Виділяють чотири типи читання:

- загальне читання;
- ретельне читання;
- швидкий перегляд;
- діагональне читання.

1.1.1 Загальне читання

Загальне читання — найбільш звичний для нас вид читання, ціль якого ознайомитися з суттю або сюжетом прочитаного у довільній формі та у будь-який, комфортний читачеві термін. Такий метод читання підходить для читання художньої літератури, загального ознайомлення з науковими статтями, та іншою літературою, що не вимагає ретельного вивчення, для ретельного вивчення існує ретельний метод читання.

Інформація отримана від такого методу читання здебільшого є важливою для читача.

1.1.2 Ретельне читання

Ретельне читання – тип читання, якому притаманний глибинний аналіз прочитаного матеріалу, увага приділяється кожному слову, кожне слово зрозуміле читачеві. За умови використання такого методу читання читач розуміє весь текст і запам'ятовує його. Після прочитання тексту таким методом читач може згадати його в деталях або поставити під сумнів

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		7

прочитане, через повне розуміння слів та змісту прочитаного матеріалу. Характерною рисою ретельного читання є більша потреба в часі витраченому на прочитання тексту, на відміну від загального читання, швидкого перегляду та «діагонального читання».

Цим методом зазвичай користуються люди яким конче необхідне повне розуміння тексту і його аналіз, наприклад філологи, або люди які читають технічну документацію до складного пристрою.

1.1.3 Швидкий перегляд

Швидкому перегляду тексту притаманна менша увага до деталей тексту, що читається, і концентрація на суті матеріалу або основній сюжетній лінії. Під час такого читання, читач нехтує побічними реченнями і словами, і виділяє лише суть, можуть пропускатись цілі абзаци тексту, за умови, що вони не є критичними для розуміння. Але більшу частину тексту читач все ж читає, через те, що не шукає щось конкретне, а ознайомлюється з матеріалом.

Такий прийом в читанні використовується для загального огляду, наприклад, коли читач вибирає книгу в книгарні і хоче зрозуміти чи зацікавить вона його. Він ознайомиться зі стилем написання автора та сюжетом, витративши на це небагато часу.

1.1.4 Діагональне читання

Діагональне читання – цей метод читання також називають скануванням, під час використання такого методу читач швидко продивляється речення у тексті з метою пошуку потрібної йому інформації, решту інформації, яку читач не потребує ігнорується.

До прикладів такого читання можна віднести читання новин, пошук потрібної дати в текстових документах, або словникового терміну, який цікавить читача.

					ІА81.300БАК.003 ПЗ	Лист
						8
Зм.	Лист	№ докум.	Підпис	Дата		

Ціль нашого мобільного додатку саме в збереженні засвоюваності матеріалу і збільшенні швидкості звичайного типу читання текстів. Для цього розглянемо наступні теорії швидкого читання [1].

1.2 Теорія швидкого читання

Швидкість читання звичайної людини статистично складає 800 знаків за хвилину при цьому кількість необхідної для роботи та навчання інформації постійно зростає. Через це з кожним роком кількість людей що цікавляться темою швидкого читання зростає.

Швидке читання – це процес швидкого розпізнавання та засвоювання фраз чи речень на сторінці одночасно, а не визначення окремих слів [2].

Нова інформація будь то електронні листи, звіти та веб-сайти на роботі, чи соціальні мережі, книги та журнали вдома. Ми, ймовірно, відчуваємо тиск, щоб отримати цю інформацію швидше, щоб ми могли «залишатися в курсі» і приймати обґрунтовані рішення.

Існують спеціальні вправи для розвитку швидкості сприймання тексту та покращення уваги до прочитаного, але в контексті дослідницького проекту вони розглянуті не будуть, вони виходять за рамки теми цієї роботи.

1.3 Методики швидкого читання

Поширення отримали чотири види (або техніки) скорочення: читання «по діагоналі», читання без зупинки, придушення субвокалізації, RSVP (швидка послідовна візуальна демонстрація).

1.3.1 Методика швидкого читання «по діагоналі»

Читання «по діагоналі» – це вид швидкого читання, що поєднує кілька технік: читання «зігзагами», яке практикував Теодор Рузвельт, читання

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		9

сторінки цілком, «спринтерська» техніка читання сторінки тексту одним поглядом [3]. Всі ці техніки ґрунтуються на практиці читання за допомогою периферійного зору.

На думку офтальмологів та біологів, читання периферійним зором не дозволяє правильно розпізнавати текст і має низькі показники засвоюваності. Крім того, читання «по діагоналі» не підходить людям із хворобами зору, наприклад людям із короткозорістю. Читаючи «по діагоналі», ви можете розрізняти окремі фрази чи слова, але об'єднання слів в осмислений текст – завдання складне, якщо не сказати більше, практично неможливе для більшості людей.

1.3.2 Безперервне читання

Практика безперервного читання полягає у фіксації читача фрагменті тексту та швидкому переході до наступного абзацу або речення. Переходи змінюються невеликою паузою для осмислення інформації, після якої процедура фіксації та переходу повторюється.

Проблема цієї техніки така сама, як і у «діагональної»: прагнучи економити час, читець не встигає освоїти текст загалом – страждає на якість сприйняття тексту. Тести показують, що не всі читачі, які користуються цією технікою, можуть переказати зміст прочитаного.

1.3.3 Придушення субвокалізації

Читаючи книгу або статтю в Інтернеті, людина промовляє "про себе" текст. Це вчені називають субвокалізацією, чи промовлянням, необхідним засвоєння інформації. Техніка придушення субвокалізації покликана заощадити час, що йде на «промовляння», і за рахунок цього прискорити швидкість читання.

Техніка дає хороше прискорення швидкості читання, але як і попередні практики, погано впливає на засвоюваність текстів. Іншими словами, вона не підходить для вивчення складних досліджень та наукових текстів, наприклад, перевантажених посиланнями, виносками, примітками, формулами. Читач просто «втратить» незнайомі слова та символи, запам'ятавши лише знайомі йому елементи тексту.

Дослідження Рональда Карвера довели, що повністю позбутися субвокалізації не можна. За допомогою досвіду з електромагнітними сенсорами вчений з'ясував, що мозок людини не може обійтися без субвокалізації, а слова промовляють навіть ті, хто читає зі швидкістю 700-800 слів за хвилину [4].

1.3.4 RSVP

Ця техніка читання ґрунтується на методі демонстрації інформації на дисплеї комп'ютера або смартфона. Метод набув широкого поширення в середині 2010-х завдяки технології Spritz. Її суть у послідовній демонстрації слів на екрані. Читач може відрегулювати швидкість відтворення тексту і, якщо відчуває сили, іноді «прискорюватися».

Головна проблема RSVP-техніки швидкого читання полягає в тому, що програми не знають, які слова знайомі читачеві і не змінюють ритмічний малюнок тексту, а "перемотування" до важкого для сприйняття фрагменту займе додатковий час. Як і у випадку з придушенням «субвокалізації», основним фактором прискорення залишається не розробка особливої навички, а багатий словниковий запас читача: чим більше людина знає слів, тим вища ймовірність, що процес швидкого читання пройде без запинок, пауз та перемотування.

Отже, техніка може збільшити швидкість читання, але не підходить для всіх текстів. За допомогою програми можна читати інформаційні

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		11

повідомлення та розважальні сайти, але важко буде зрозуміти зміст художнього тексту [5].

Суть цієї техніки буде використана у нашому мобільному додатку – послідовна демонстрація слова з акцентом на середній літері, графічне зображення рисунок 1.1.



Рисунок 1.1 – Вирівнювання слова способом RSVP[7]

Техніка читання RSVT базується на здатності зору людини до периферійного читання, коли людина концентрує свою увагу на центральній літері, або слові, мозок «автоматично» сприймає решту слова.

1.3.5 Техніка читання Spritz

Технологія Spritz базується на методі RSVT але використовує вирівнювання слів за акцентовою літерою, для збільшення швидкості читання і більшій зручності читача[7]. На рисунку 1.2 зображене графічне представлення цієї техніки.



Рисунок 1.2 – Вирівнювання слова методом Spritz[7]

Показник ORP – показник оптимального позиціонування для розпізнавання, технологія Spritz, реалізує його краще ніж RSVP, тому що, слово позиціонується до умовної лінії центру, що дає змогу читачеві менше рухати очима в пошуку правильної позиції для читання. На рисунку 1.2, ця лінія позначена пунктиром – лінія вирівнювання слова.

Як видно з рисунку 1.2 метод вирівнювання Spritz мінімізує рух очей порівняно з методом RSVP, що в свою чергу зможе показати нам позитивну динаміку в сприйнятті тесту під час його швидкого читання та підвищити показник ORP слів в цілому

Показник ORP є надважливим у питанні створення такого додатку, якщо акцент буде зроблено не в тому місці довгого слова, або буде зміщений в сторону, то мозок читача почне підсвідомо шукати оптимальну позицію для прочитання слова і користувач додатку буде швидше втомлюватися, або навіть пропускати декілька наступних слів, що є неприйнятним для мети дослідницького проєкту.

Висновки до розділу 1

У цьому розділі була дослідження предметна область читання та швидкого читання. Серед методів читання було виділено їх типи та наведено їх основні характеристики. Окрема увага приділялась методу швидкого читання, основні з них були виділені та проаналізовані додатково з вказанням на недоліки кожного з них.

Окрема увага була приділена методу RSVT та Spritz, як двом методам на яких буде базуватись метод читання мобільного додатку.

Основна мета виходячи з проведеного аналізу – зберегти здатність користувача сприймати матеріал і робити це швидко та комфортно.

З аналіз технік читання було зроблено висновки щодо підходу, який буде використано у нашому мобільному додатку. А саме, це буде модифікація техніки швидкого читання Spritz, техніка послідовної демонстрації з акцентом на центрі слова, також було прийнято рішення додати можливість використання цієї технології на сукупності слів, з акцентом на центральному слові або центрі центрального слова. Тож аналіз технік швидкого читання дає змогу визначити оптимальний варіант для реалізація власного методу читання матеріалу. Також у мобільному додатку буде збережена можливість використання технік читання RSVT та Spritz.

Через розбіжність у здатності людей до читання і різниці словникового запасу буде реалізована гнучка система вибору швидкості читання текстового матеріалу.

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		14

2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Більша частина існуючих рішень має низку недоліків, або взагалі не виконують свою функцію. Частина додатків зосереджується на монетизації, що заважає читачеві сприймати інформацію під час читання, або не повністю реалізує технологію швидкого читання, що також робить додатки не корисними та погано впливає на ідею швидкого читання в цілому, користувач завантажує додаток, ознайомлюється з ним, розуміє що йому не комфортно ним користуватись і вирішує що взагалі такий метод читання йому не підходить, через такі помилки в створенні і реалізації додатків для швидкого читання не може здобути популярності серед людей. Нижче наведені приклади реалізації таких додатків.

2.1 Мобільний додаток: «Reedy»

Цей додаток було створено загалом для читання текстових електронних книг, статей та інтернет сторінок на мобільних пристроях з можливістю переводу тексту в голос, але акцент було зроблено на функцію швидкого читання матеріалу за технологією RSTV [8], але реалізована вона не повністю, на екран виводиться слово без акценту на його центральній літері, як того вимагає технологія швидкого читання. Також до недоліків можна віднести позиціонування слова на екрані, воно зміщене до лівої частини та вгору від центру екрану, якщо зміщення вгору можна пояснити, як частину адаптивності розмітки під розмір екрану смартфона, то зміщення слова ліворуч лише заважає читанню, що в сукупності з відсутністю акцентової літери слова негативно впливає на показник ОРР, рисунок 2.1.

Отже, додаток включає в себе функцію швидкого читання, але реалізована вона не повністю, і не відповідає тій технології яку зазначено в описі мобільного додатку.

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		15

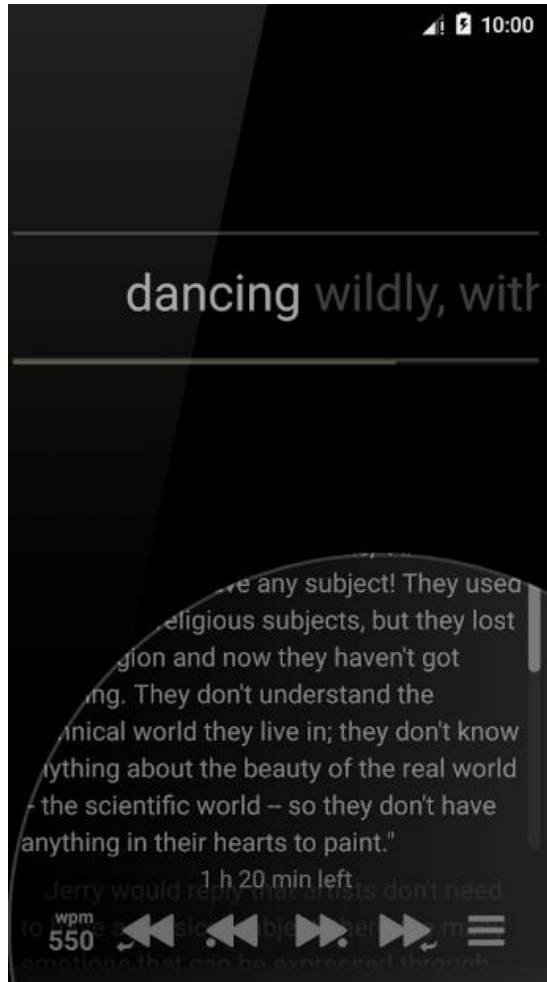


Рисунок 2.1 — Позиціонування слова в мобільному додаткові «Reedy»

2.2 Мобільний додаток «Speed reading»

Додаток «Speed reading» було створено для навчання швидкого читання і розвитку цієї здібності у користувачів [9]. Додаток включає в себе велику кількість вправ для розвитку уваги та покращення навичок необхідних для швидкого читання та запам'ятовування інформації, рисунок 2.2.

Окрім вправ, додаток має функцію читання завантажених з пристрою файлів, проте не вийшло протестувати і вивчити реалізацію цієї функції додатку, через те, що файл не відкрився, а додаток закінчив свою роботу, спробувавши ще декілька разів на інших файлах та пристроях, і отримавши той самий результат, стало зрозуміло що функція, не працює.

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		16

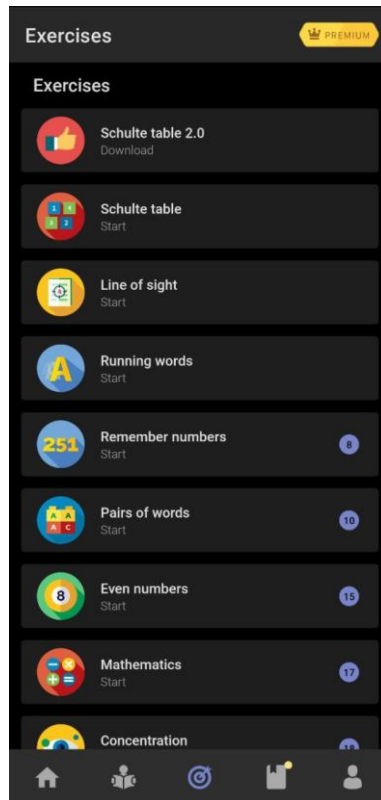


Рисунок 2.2 – Інтерфейс додатку «Speed reading» з вправами

Помилка читання файлу в додатку «Speed reading», рисунок 2.3.

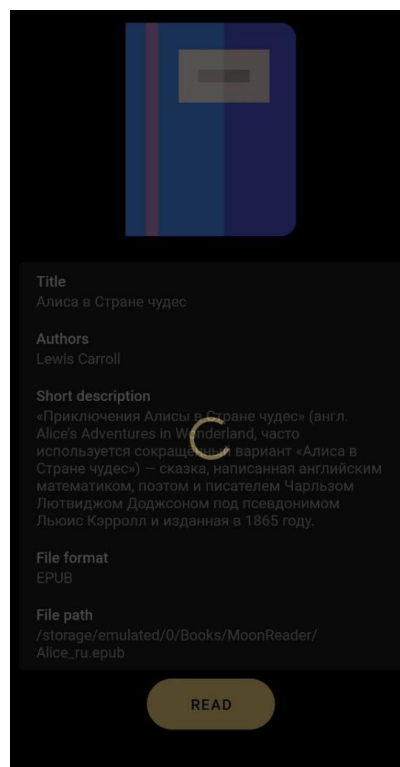


Рисунок 2.3 – Інтерфейс додатку «Speed reading» помилка при завантаженні

					Лист
					17
Зм.	Лист	№ докум.	Підпис	Дата	ІА81.300БАК.003 ПЗ

Отже, додаток має обширний набір інструментів для розвитку техніки читання та розвитку уваги, також він має в собі функцію ведення статистики занять, але реалізація швидкого читання тексту з файлів реалізована не повним чином, що унеможливорює читання файлів при використанні цього додатку.

2.3 Мобільний додаток «Speed Reader»

Додаток «Speed Reader», як видно з його домашньої сторінки орієнтований саме на скорочитанні, рисунок 2.4 [10].

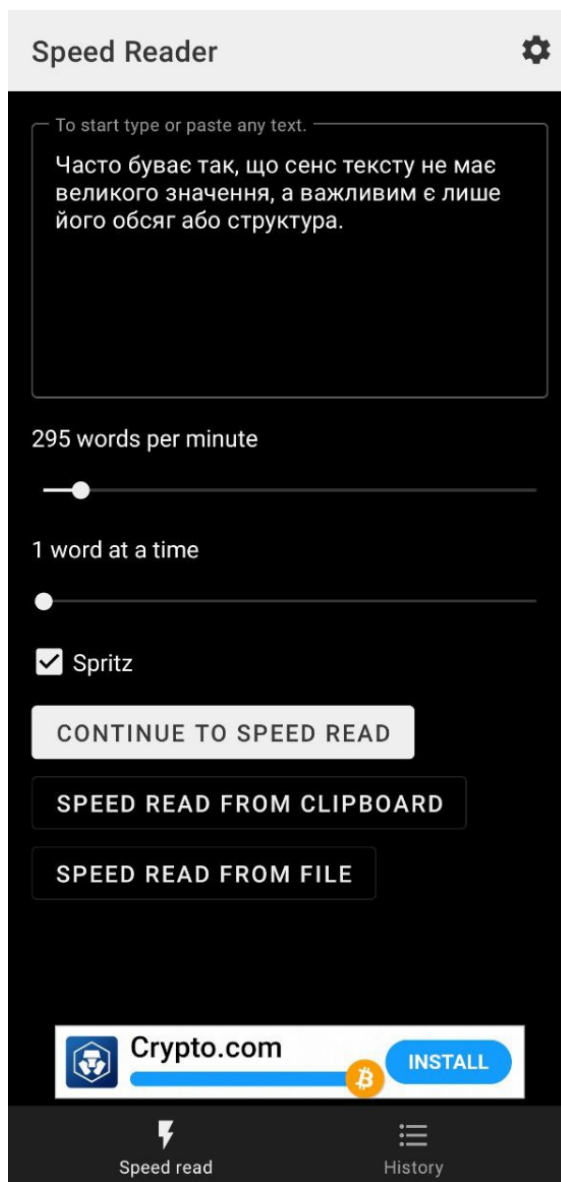


Рисунок 2.4 – Домашня сторінка «Speed Reader»

На домашній сторінці знаходиться вибір швидкості читання та кількості слів для читання, також можливість використання технології Spritz, та можливість читання файлів з девайсу.

Метод реалізації технології Spritz в додатку реалізовано повною мірою та відповідає вимогам технології описаної в документації, рисунок 2.5.

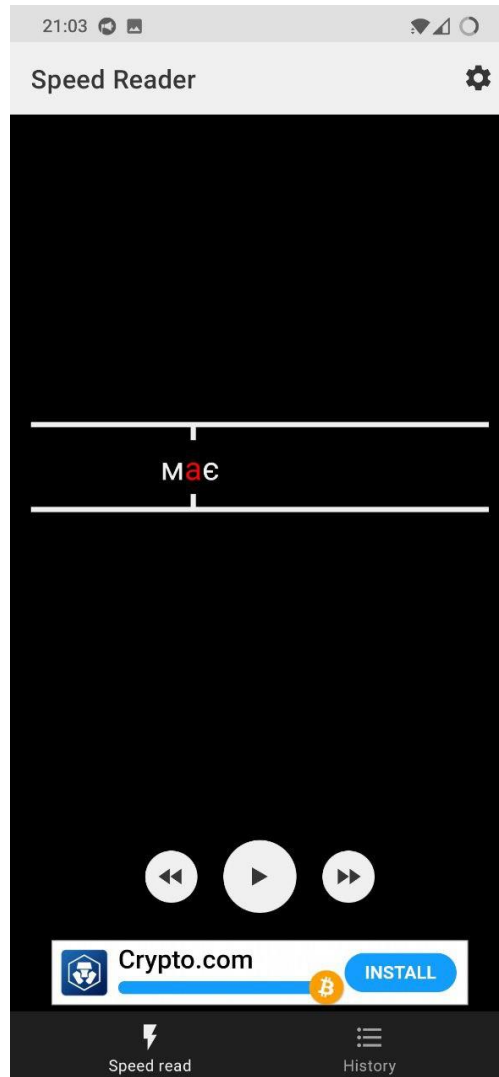


Рисунок 2.5 – Реалізація технології Spritz у додатку «Speed Reader»

Загалом додаток виконує свої функції, та дає змогу користуватися перевагами швидкого читання з будь яким типом файлів і текстами з буферу обміну, і файлами завантаженими з пам'яті пристрою.

Також у додатку реалізована історія прочитаного матеріалу, і до файлів прочитаних раніше можна повертатися і продовжувати читання.

					ІА81.300БАК.003 ПЗ	Лист
						19
Зм.	Лист	№ докум.	Підпис	Дата		

До недоліків можна віднести застарілий інтерфейс та відсутність альбомної орієнтації екрану, не можливість слідкувати за відсотком прочитаного тексту та за текстом в цілому.

2.4 Мобільний додаток «Balto Speed Reading»

Цей додаток є лідером серед додатків для швидкого читання на мобільних пристроях, він сконцентрований на методі швидкого читання RSVP та очікувано відповідає всім вимогам технології, включає в себе низку налаштувань для комфортного читання, зміну кольорів фону та тексту [11]. Приклад модифікованого інтерфейсу, рисунок 2.6.

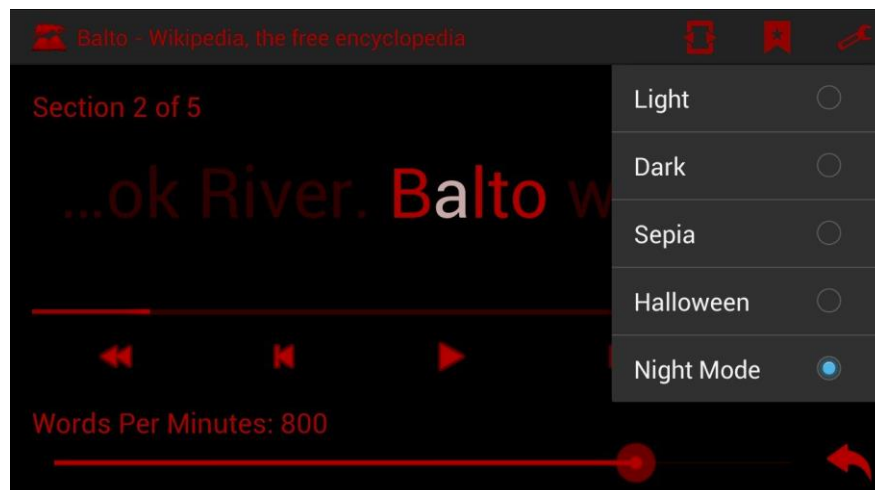


Рисунок 2.6 – Модифікований інтерфейс мобільного додатку «Balto Speed reading» [12]

Додаток реалізовано багатьма мовами та на відміну від інших додатків реалізовано читання за параграфами (після прочитання параграфу відбувається затримка для логічного закінчення).

Також «Balto Speed Reading» включає в себе технологію читання веб-сток, та файлів з хмарового сховища, що безумовно є перевагою додатку.

Загалом мобільний додаток відповідає більшості вимог, для додатку метою якого є швидке читання текстів, але не є сучасним та не має в собі функції

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		20

читання в портретному режимі, що є не зручним, але важливим в реалізації для зручності використання.

2.5 Порівняльна характеристика існуючих рішень

Після дослідження існуючих рішень, визначено переваги і недоліки кожного з них, та складено таблицю приведену нижче, таблиця 2.1.

Таблиця 2.1 — Порівняльна таблиця існуючих додатків.

Функції	Reedy	Speed reading	Speed Reader	Balto Speed Reading
Завантаження файлів	+	-	+	+
Зміна методик швидкого читання	-	-	+	-
Зручність інтерфейсу	+	+	-	+
Історія файлів	-	+	+	+
Сучасність	+	+	+	-

Висновки до розділу 2

Було виявлено переваги та недоліки кожного з додатку та створено порівняльну таблицю.

Грунтуючись на недоліках існуючих додатків для швидкого читання, можливо уникнути цих недоліків у нашому додаткові. Використовуючи переваги існуючих рішень можливо мімікрувати до них в створюваному рішенні.

Недолік додатку «Speed reading» доводить важливість тестування додатків, для уникнення його помилок, що призводять до його майже повної неієздатності.

Після аналізу існуючих рішень було з'ясовано, що додатки мають необхідний функціонал, але немає рішення яке б поєднувало всі необхідні технології в одному додаткові. Це доводить що створення такого додатку є затребуваним і актуальним.

					ІА81.300БАК.003 ПЗ	Лист
						22
Зм.	Лист	№ докум.	Підпис	Дата		

3 ВИБІР ТЕХНОЛОГІЙ ТА РЕАЛІЗАЦІЇ

3.1 Засоби проектування

До засобів проектування відносяться сервіси або додатки за допомогою яких було створено макети та схеми взаємодії користувача з мобільним додатком.

3.1.1 Сервіс для розробки користувацького інтерфейсу «Figma»

Figma – онлайн-сервіс для розробки прототипів та дизайну користувацького інтерфейсу [12].

Сервіс працює у онлайн форматі, і у форматі додатку, що встановлюється на пристрій користувача, також має можливість сумісної, одночасної роботи над одним проектом.

Figma використовує векторну графіку для примітивів та тексту, що дозволяє працювати з макетами будь-якого розміру, а сучасний та лаконічний інтерфейс дозволяє користувачу легко в ньому орієнтуватися. Наявність безкоштовних бібліотек з готовими базовими елементами інтерфейсу, значно спрощують роботу над макетами.

До технічних вимог сервісу відноситься необхідність у використанні операційної системи «Windows» або «macOS» та наявність стабільного підключення пристрою до мережі «Інтернет».

Послуги сервісу надаються безкоштовно для індивідуальних користувачів, але є платними для фахових команд.

З огляду на переваги зазначені вище, сервіс «Figma» є одним з найкращих сучасних рішень, для створення прототипів користувацького інтерфейсу, тож користувацький інтерфейс додатку було створено саме за використання цього сервісу.

3.1.2 Сервіс для побудови діаграм «diagrams.net»

Для побудови діаграм та схем дослідницького проекту, було використано сервіс «diagrams.net», зважаючи на його переваги у порівнянні з аналогам.

diagrams.net (раніше draw.io) - це безкоштовне кросплатформне програмне забезпечення для малювання графіків із відкритим вихідним кодом, розроблене на HTML5 та JavaScript. Його інтерфейс можна використовувати для створення діаграм, таких як блок-схеми, каркаси, діаграми UML, організаційні діаграми та діаграми мережі.

diagrams.net доступний у режимі онлайн, як крос-браузерна веб-програма та як офлайн-додаток для настільних комп'ютерів для Linux, macOS та Windows. Веб-додаток не вимагає онлайн-входу чи реєстрації, його можна відкривати з локального жорсткого диска та зберігати на ньому.

Сучасний сервіс для побудови діаграм, що має в собі численні переваги, такі як:

- інтегрування з хмарними сервісами;
- доступність як плагіну, для вбудовування веб-програм.

З огляду на переваги, програмного забезпечення було встановлено його доцільність для його використання при виконанні дослідницького проекту.

3.2 Техніки швидкого читання у додатку

У мобільному додатку буде реалізовано більшість з розглянутих технологій швидкого читання, які можливо адаптувати для використання на мобільному пристрої.

3.2.1 Техніка RSVP у додатку

Техніку читання RSVP, яку було описано у розділах вище, у її первинному вигляді можливо використовувати, лише при відображенні слів

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		24

які виводяться по черзі на екран пристрою, представлення такого варіанту читання зображено на рисунку 3.1.



Рисунок 3.1 – Прототип реалізації технології RSVP з відображенням актуального для читання слова

Цей метод реалізації є найбільш відповідним до умов технології RSVP. Як видно за рисунку 3.1, слово центрується на екрані за своєю довжиною і акцент ставиться на центральну літеру (на рисунку позначена червоним). Також в мобільному додатку буде реалізовано модифікацію цього методу, яка даватиме змогу читати тестовий файл з використанням технології RSVP, але при відображенні повного текстового файлу на екрані пристрою, приклад такої реалізації наведено на рисунку 3.2.

Часто буває так, що сенс тексту не має великого значення, а важливим є лише його обсяг або структура.

Часто буває так, що сенс тексту не має великого значення, а важливим є лише його обсяг або структура.

Часто буває так, що сенс тексту не має великого значення, а **ВАЖЛИВИМ** є лише його обсяг або структура.

Часто буває так, що сенс тексту не має великого значення, а важливим є лише його обсяг або структура.

Часто буває так, що сенс тексту не має великого значення, а важливим є лише його обсяг або структура.

Рисунок 3.2 – Прототип реалізації технології RSVP з повним відображенням тексту

Така реалізація методу допоможе користувачам у яких менш розвинута техніка читання, комфортно користуватися її перевагами, так як вона більш схожа на звичну техніка читання. Також до переваг такого методу можна віднести можливість користувача периферійним зором сприймати наступне слово.

3.2.2 Техніка Spritz у додатку

Технологія Spritz має відмінності в реалізації від технології RSVT. Як було описано раніше за реалізації техніки Spritz актуальне слово для читання центрується за центральною літерою слова, як відображено на рисунку 3.3.



Рисунок 3.3 – Прототип реалізації технології Spritz з актуальним словом

В цій реалізації буде змога прибрати колір акцентовій літері, або змінити його на бажаний. Така реалізація, за технологією Spritz, мінімізує потребу в русі очей, під час читання, що збільшує показник ОРР.

Для цієї технології також буде створено модифікацію, для читання текстів, з повним відображенням тексту, як показано на рисунку 3.4. При такому методі реалізації важче сприймати наступне слово, але легше розуміти, де в тексті знаходяться логічні абзаци, що позитивно впливає на сприйняття тексту в цілому.

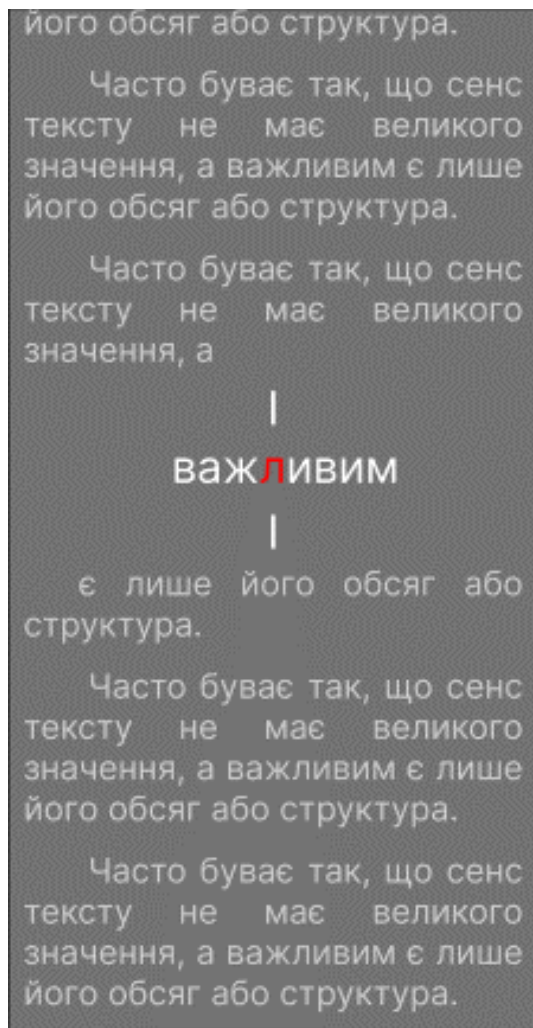


Рисунок 3.4 - Прототип реалізації технології Spritz з повним текстом

3.2.3 Власна техніка швидкого читання

Грунтуючись на аналізі актуальних технік швидкого читання, було розроблено власну техніку, яку можна охарактеризувати як модифікацію техніки Spritz.

В цій техніці не використовується акцентована літера слова, акцент ставиться на центрі слова в цілому.

Під час тестування методів читання було виявлено, що акцентована літера (неважливо помічена вона кольором чи ні), відволікає на себе зайву увагу, що негативно впливає на показник ОРР. Тож було вирішено відмовитися від неї. Інші переваги методу Spritz, буде збережено.

Приклад реалізації власної техніки швидкого читання зображено на рисунку 3.5 представлено і метод читання «по слову», і метод читання з повним текстом.

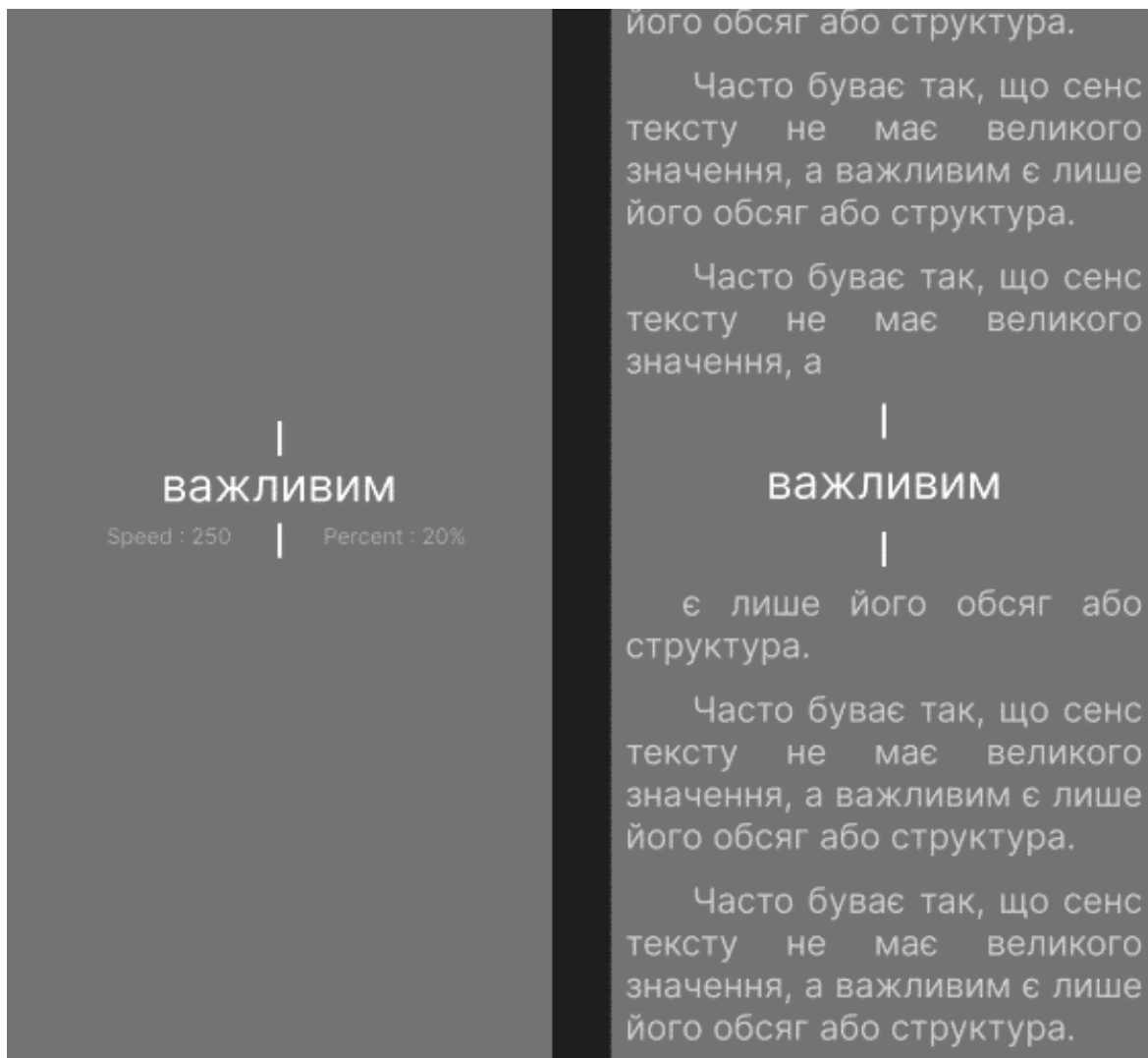


Рисунок 3.5 – Прототип реалізації власного методу швидкого читання

Такий метод читання мінімізує візуальний шум на екрані читання, що підвищує показник ОРР.

Для визначення продуктивності власної технології швидкого читання з додатків користувачів, за їхньої згоди, буде зібрано інформацію про частоту використання даного методу у порівнянні з іншими методами швидкого читання, також цей метод буде встановлено за замовчуванням у додатку.

3.2.4 Техніка читання «Вказівник»

Техніка читання «Вказівник» це звична техніка читання для користувачів з невеликим вдосконаленням, а саме додаванням вказівника до читаного тексту (пальцем у реальному житті).

Додавання вказівника слугує для концентрації читача на певному слові та не дає йому повертатись до читаного матеріалу, що позитивно впливає на розвиток техніки читання та уважності читача.

Вказівник, що рухається з поступовою швидкістю, є однією з умов для поступового та рівномірного сприйняття читаного матеріалу. Приклад реалізації техніки читання «Вказівник» наведено на рисунку 3.6.

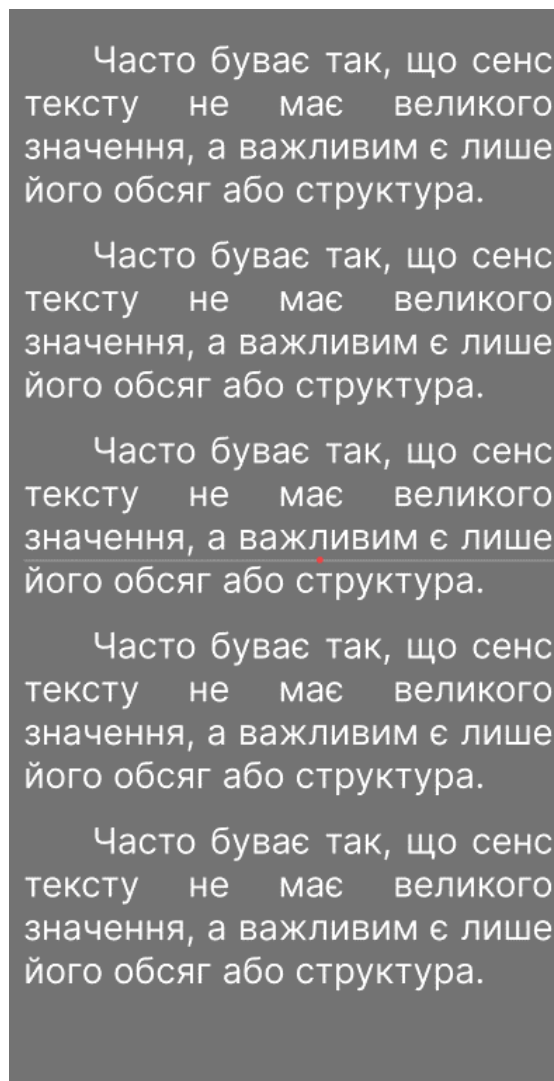


Рисунок 3.6 – Прототип реалізації техніки читання «Вказівник»

Вказівник на рисунку 3.6 заражений у вигляді червоної крапки, але його колір та форма може бути змінений у вкладці налаштування додатку. Вказівник рухається від слова до слова з поступовою швидкістю, попередньо вказаною перед читанням текстового матеріалу в налаштуваннях техніки читання.

3.3 Засоби розробки

До засобів розробки відносяться мови програмування та середовища розробки які будуть використані у створенні наукового дослідницького проєкту.

3.3.1 Мова програмування Kotlin

Kotlin – статично типизована, об’єктно-орієнтована мова програмування працююча поверх JVM і розробляється компанією JetBrains[13]. Компілюється також в JavaScript. Мова є рекомендованою для створення Android додатків.

Також мова Kotlin використовується в таких додатках як:

- Gradle;
- Corda;
- Evernote;
- Coursera;
- Spring;
- Atlassian (Trello).

Мова Kotlin використовується для написання серверної сторони додатків. Приклади відомих компаній які використовують Kotlin на стороні серверу у своїх продуктах:

- Spring;
- Adobe;
- Atlassian;

- Shazam;
- Amazon Web Services;
- Expedia group
- Intuit
- ING.

Мова Kotlin є сумісною з екосистемою Java, всі бібліотеки JVM сумісні з мовою Kotlin.

Однією з особливостей мови Kotlin є можливість написання мультиплатформових додатків. Один і той же код написаний мовою Kotlin з використанням сервісу Multiplatform може використовуватись і в Android додатку, і у додатку на операційній системі IOS, і у веб версії додатку.

Мова Kotlin була обрана як основна мова написання додатку зважаючи на всі її переваги.

3.3.2 Мова розмітки XML

XML - стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет [14].

Мова XML це спрощена підмножина мови розмітки SGML. Сама мова складається з текстових знаків, мова є придатною для читання людиною
 рисунок 3.9.

Окрім написання цією мовою самого користувацького інтерфейсу додатку, також на цій мові буде реалізовано файли зберігання текстових полів, та файл Manifest – який являє собою основну інформацію про програму та систему, векторні зображення, що використовуватимуться у додатку, також буде реалізовано мовою XML.

Повністю на мові розмітки XML у додатку буде написано інтерфейс користувача.

```

<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:padding="8dp"
    app:cardCornerRadius="8dp"
    app:cardElevation="8dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/card1"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/cv_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:padding="8dp"
            android:text="Title"
            android:textSize="16sp"
            android:textStyle="bold"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

```

Рисунок 3.9 – Приклад синтаксису мови розмітки XML з розроблюваного додатку

Кожен документ XML має бути синтетично коректним для його коректної роботи. Коректний документ – документ що відповідає всім синтаксичним правилам написання документу XML.

Коректний документ XML включає в себе:

- лише один елемент у корені;
- елементи розмітки позначаються (<пункт>Пункт 1</пункт>);
- один елемент не може мати декілька атрибутів з однаковою назвою;
- елементи можуть бути вкладеними, але мусять повністю перебувати в елементі в якому знаходяться;
- документ має складатися лише з дозволених символів Юнікоду.

3.3.3 Середовище розробки Android Studio

При розробці програмного забезпечення для мобільного телефону доцільно використовувати середовище розробки адаптоване для цього, таким середовищем розробки є Android Studio, це офіційне середовище від компанії Google, створене у колаборації з JetBrains і базується на платформі IntelliJ IDEA [15].

До переваг можна віднести те, що в Android Studio є інтегрований емулятор Android пристроїв, що спрощує тестування та розробку програмного забезпечення. Також є інтегрований редактор XML з обширною бібліотекою компонентів для створення користувацького інтерфейсу – візуальний редактор макетів.

За допомогою візуального редактору макетів створюються, або додаються з існуючої бібліотеки елементи користувацького інтерфейсу, в редакторі дизайну автор може попередньо переглядати, як виглядає його користувацький інтерфейс на пристроях з різним розміром екрану та версією Android, та адаптувати його відповідно до розміру екрану на різних пристроях, щоб інтерфейс був однаково зручний на екранах різного розміру.

Також у функціоналі середовища Android Studio є «Аналізатор APK», який створений для того, щоб автор додатку мав змогу попередньо оцінити розмір додатку та за можливості або потреби, зменшити його і перевірити вміст APK на наявність зайвих файлів. Розмір APK – важливий показник оптимізації додатку, чим більший розмір файлу завантаження, тим довше додаток буде відкриватись на пристроях. Занадто великий розмір додатку може призвести до його непрацездатності на пристроях що мають невеликі обчислювані потужності.

Це середовище програмування також надає розробнику можливість тестування додатку на етапах розробки та релізу.

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		34

3.3.4 Постачальник хмарних послуг Firebase

Firebase – американська компанія, постачальник хмарних послуг, заснована в 2011 році Ендрю Лі та Джеймсом Тэмпліном, і поглинена в 2014 році корпорацією Google [16].

Користуючись сервісом, розробник буде мати змогу випускати більш якісний та керований продукт, адже хмарні технології пропонують більш гнучку взаємодію з додатком після його випуску на масову аудиторію.

Firebase надає низку хмарних послуг таких як:

- послуга керованої серверної інфраструктури;
- послуга для релізу та моніторингу додатків;
- послуга з супроводження додатків.

Детальніше послуги які будуть використані зображені на кресленику ІА81.300БАК.003 Д1.

Керована серверна інфраструктура буде використана у проєкті для реєстрації нових користувачів та доступу до раніше зареєстрованих аккаунтів, у додатку буде реалізовано систему аутентифікації. Також ця ж інфраструктура дає змогу реалізувати хмарову базу даних, яку і буде реалізовано в проєкті, для зберігання текстових файлів, невеликого розміру (до 5 мегабайт).

Послуга з релізу та моніторингу додатку, буде використовуватися для моніторингу після релізу мобільного додатку до Google Play, що дасть змогу відслідковувати активність користувачів на сторінках додатку, та мати представлення того, які сторінки застосунку слід покращити в майбутній оновленнях. Також, ця послуга даватиме нам змогу аналізувати додаток на предмет помилок, які можуть виникнути після релізу та з'ясувати в якій частині додатку вони з'являються. Як було зазначено в науковій роботі раніше цей сервіс даватиме змогу відслідковувати який саме режим сокорочитання користується популярністю у користувачів і відповідно до цього виконувати майбутні оновлення застосунку.

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		35

Послуга з супроводження додатків системи Firebase буде використана під час майбутніх модифікацій додатку, для тестування нового функціоналу та зміни поведінки додатку після релізу. Для цього є система «Remote Config» - для зміни конфігурації в проєкті після релізу, та послуга «A/B testing» - для локалізованого тестування майбутнього нового функціоналу. Як саме мобільний даток буде використовувати сервіс, зображено на кресленику ІА81.300БАК.003 Д2.

3.3.5 Система автоматичного збирання Gradle

Gradle - система автоматичного збирання, яка розвиває принципи, закладені в Apache Ant та Apache Maven і використовує предметно-орієнтовану мову (DSL) на основі мови Groovy замість традиційної XML-подібної форми представлення конфігурації проєкту [17]. Система Gradle є сумісною з більшістю актуальних IDE, такими як:

- Android Studio;
- IntelliJ IDEA;
- Eclipse;
- NetBeans.

Gradle є популярною системою збірки проєктів, цій системі довіряють свій продукт найбільші компанії представлені наразі на ринкові послуг. Системі «Gradle» довіряють найбільші компанії ринку послуг, такі як:

- LinkedIn;
- Android;
- Netflix;
- Adobe;
- Elastic.

Використання цієї системи є рекомендованим Google для збірки мобільних додатків на базі операційної системи «Android», та дозволяє розробнику зручно користуватися сторонніми бібліотеками, для додавання

					ІА81.300БАК.003 ПЗ	Лист
						36
Зм.	Лист	№ докум.	Підпис	Дата		

бібліотеки до додатку використовується одна лінія коду в Gradle файлі. Також за використання Gradle буде створено унікальний, зашифрований ключ-підпис, для ідентифікації унікальності додатку, він буде необхідний для додавання проєкту до Play Market та майбутніх оновлень додатку.

Під час створення додатку, саме система Gradle використовувалась для збірки проєкту, представлення файлів зображено на рисунку 3.9.

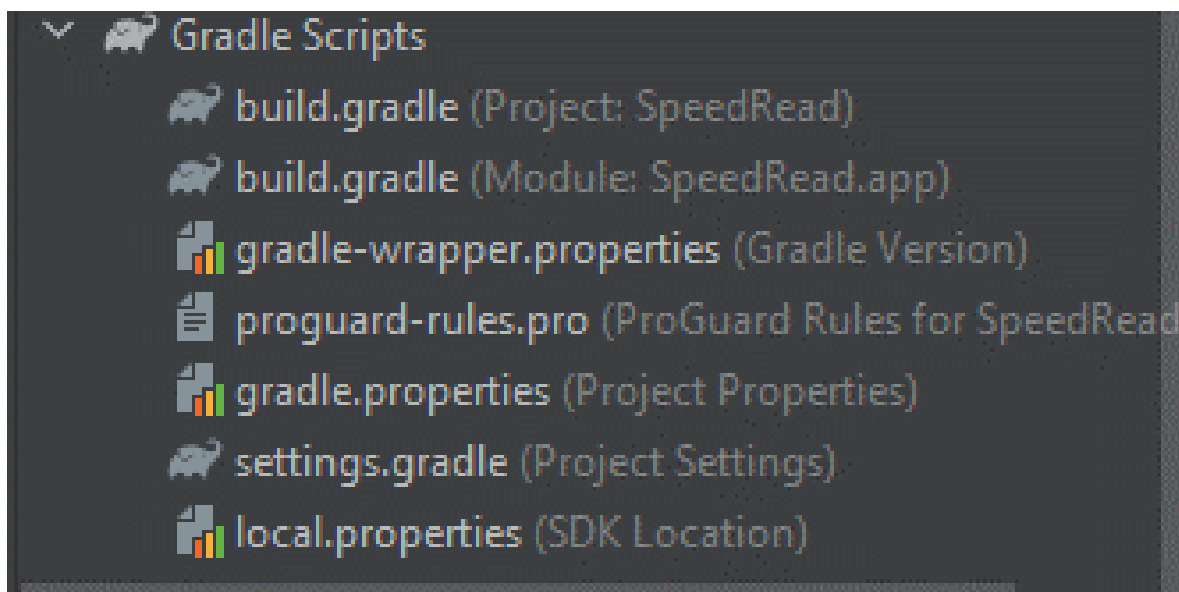


Рисунок 3.9 – Файли «Gradle», що використані у мобільному додаткові

Висновки до розділу 3

У розділі були розглянуті обрані сучасні засоби проектування, та наведено аргументацію щодо їх вибору.

Детальний розгляд технік швидкого читання дав змогу прийняти рішення щодо технік, які будуть реалізовані у мобільному додатку.

Також, у розділі були розглянуті засоби розробки Android додатку та обрано технології, які будуть використані при створенні мобільного додатку.

Як основний спосіб читання в додатку буде встановлено власний метод читання та додано послугу Firebase з релізу та моніторингу, для відстежування працездатності методу, та додано Firebase послугу з супроводження додатків, для можливості зміни основного методу читання, якщо метод, запропонований

як основний, виявиться непридатним для використання, або не сподобається користувачам додатку. Як додаткові методи читання, буде використано метод читання Spritz, RSTV та метод читання «Вказівник», статистика про використання цих методів також буде зібрана на опрацювання, та на основі цих даних будуть здійснюватися майбутні оновлення додатку.

Мовою програмування логічної частини додатку було обрано мову програмування Kotlin, зважаючи на рекомендації Google, та її переваги порівняно з іншими мовами, які застосовують для написання Android додатків. Мовою написання клієнтської частини застосунку було обрано мову XML.

Системою автоматичного збирання проєкту обрано - Gradle, окрім того що система має значну кількість переваг, ця система є рекомендованою для використання під час створення Android застосунків.

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		38

4 РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ

У цьому розділі буде описано реалізацію користувацького інтерфейсу та основних частин логічної реалізації додатку.

4.1 Користувацький інтерфейс додатку

У розділі буде спроектовано користувацький інтерфейс мобільного додатку та реалізовано його у додатку.

4.1.1 Створення логотипу додатку

Логотип додатку – важлива частина створення додатку, адже він має бути впізнаваним поміж інших додатків та лаконічним.

За основу логотипу було взято перші літери назви додатку «SpeedRead». Створений логотип, зображено на рисунку 4.1.

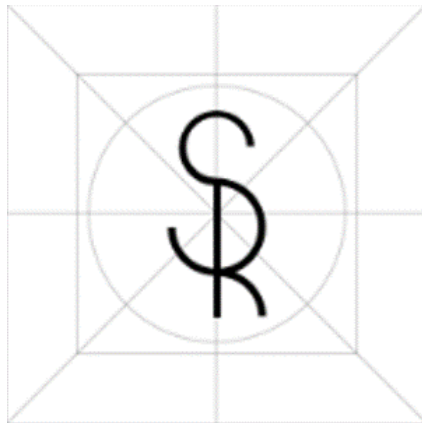


Рисунок 4.1 – Логотип застосунку

Як видно з рисунку 4.1, логотип створений за правилами створення, які описано в документації «Material Design» [18], це зроблено для адаптивності «іконки» під різні розміри екрану девайсів.

4.1.2 Проектування користувацького інтерфейсу

Для проектування користувацького інтерфейсу було використано сервіс «Figma», створений інтерфейс базується на кресленику ІА81.300БАК.003 ДЗ. Результат проектування зображено на рисунку 4.2.

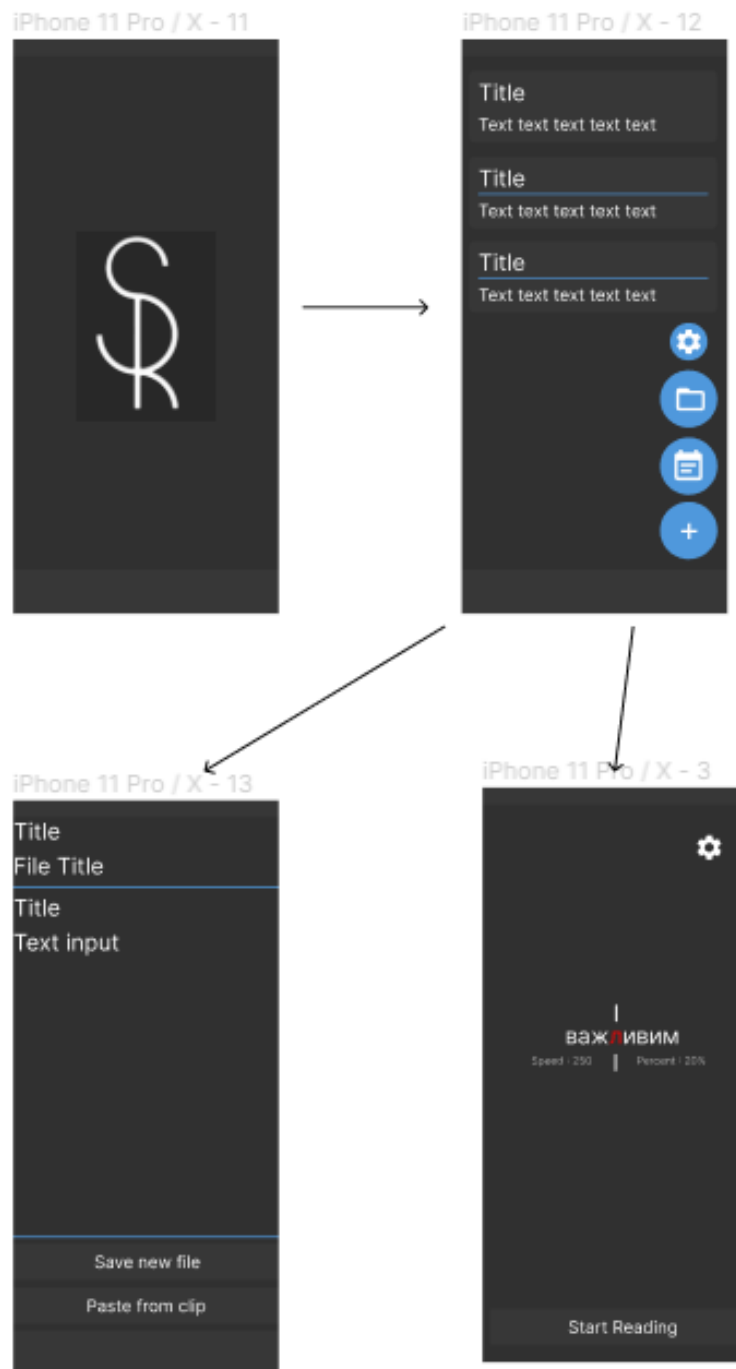


Рисунок 4.2 – Прототип користувацького інтерфейсу

За макетом зображеним на рисунку 4.2, відбувалося створення користувацького інтерфейсу мобільного додатку.

4.1.3 Реалізація користувацького інтерфейсу

Для реалізації багатосторінкового додатку, необхідно розуміти послідовність користувача під час використання додатку, для цього було розроблено кресленик IA81.300БАК.003 Д4. Також необхідно, побудувати додаток таким чином, що сторінки додатку мали змогу замінювати одна-одну, та правильно виконувати свої навігаційні функції. У додатку, як основа його користувацького представлення, буде використано елемент під назвою «Coordinator Layout». Coordinator Layout призначений для двох випадків використання:

- як декоратор верхнього рівня додатку;
- як контейнер для конкретної взаємодії з одним або декількома дочірніми представленнями.

У додатку елемент розмітки, буде використано для обох випадків, він буде відповідати і за верхній рівень додатку, і виконувати функції контейнера рисунок 4.3.

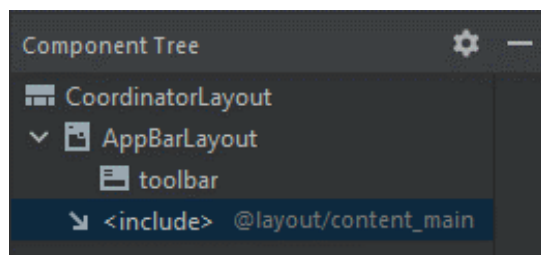


Рисунок 4.3 – Дерево компонентів основного представлення додатку

Як зазначено на дереві компонентів, зображеного на рисунку 4.3 Coordinator Layout, як основний елемент користувацького інтерфейсу включає в себе контейнер навігації який знаходиться у файлі на який посилається представлення.

Контейнер навігації складається з фрагменту, який за використання «nav_graph.xml», замінюється на потрібне представлення.

Система навігації що використана у проєкті має назву – «Navigation component». Система дає змогу створювати у графічному редакторі «Навігаційний графік». «Навігаційний графік» - графічне представлення навігації по екранам додатку, розроблену навігацію зображено на рисунку 4.4.

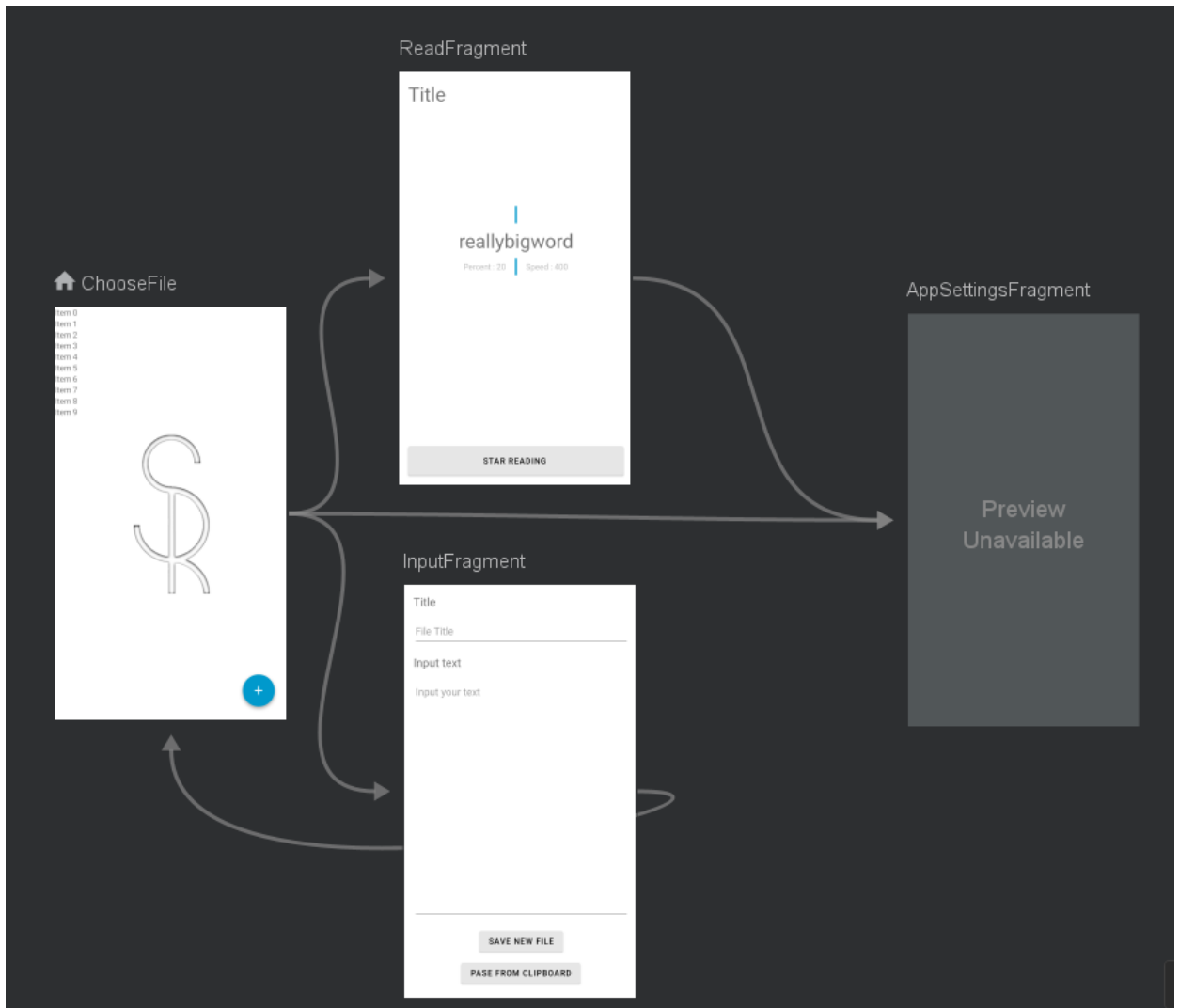


Рисунок 4.4 – Навігаційний графік мобільного додатку «SpeedRead»

Кожній стрілці з «Навігаційного графіку» відповідає поле id, за яким здійснюють перехід представлень. Приклад коду з id зображено на рисунку 4.5.


```
<action
  android:id="@+id/action_ChooseFile_to_readFragment"
  app:destination="@id/ReadFragment" />
```

Рисунок 4.5 – Фрагмент коду з «Навігаційного графіку»

Перша сторінка додатку , яку побачить користувач - сторінка вибору текстового файлу для читання, що також виконує функцію екрану завантаження. Екран завантаження представлений на рисунку 4.6.



Рисунок 4.6 – Реалізація екрану завантаження мобільного додатку

Під час того як користувач бачить екран завантаження, відбувається запит до бази даних сервісу «Firestore Cloud Database», де зберігається історія прочитаних файлів користувачем у минулому, або перший документ, що буде додано за вмовчуванням, щоб користувач мав змогу протестувати функціонал

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		43

додатку на готовому файлі, у подальшому він матиме змогу видалити цей файл.

Після запиту до бази даних, за умови успішності запиту, користувач побачить завантажені файли, та комплексну кнопку навігації, як зображено на рисунку 4.7.

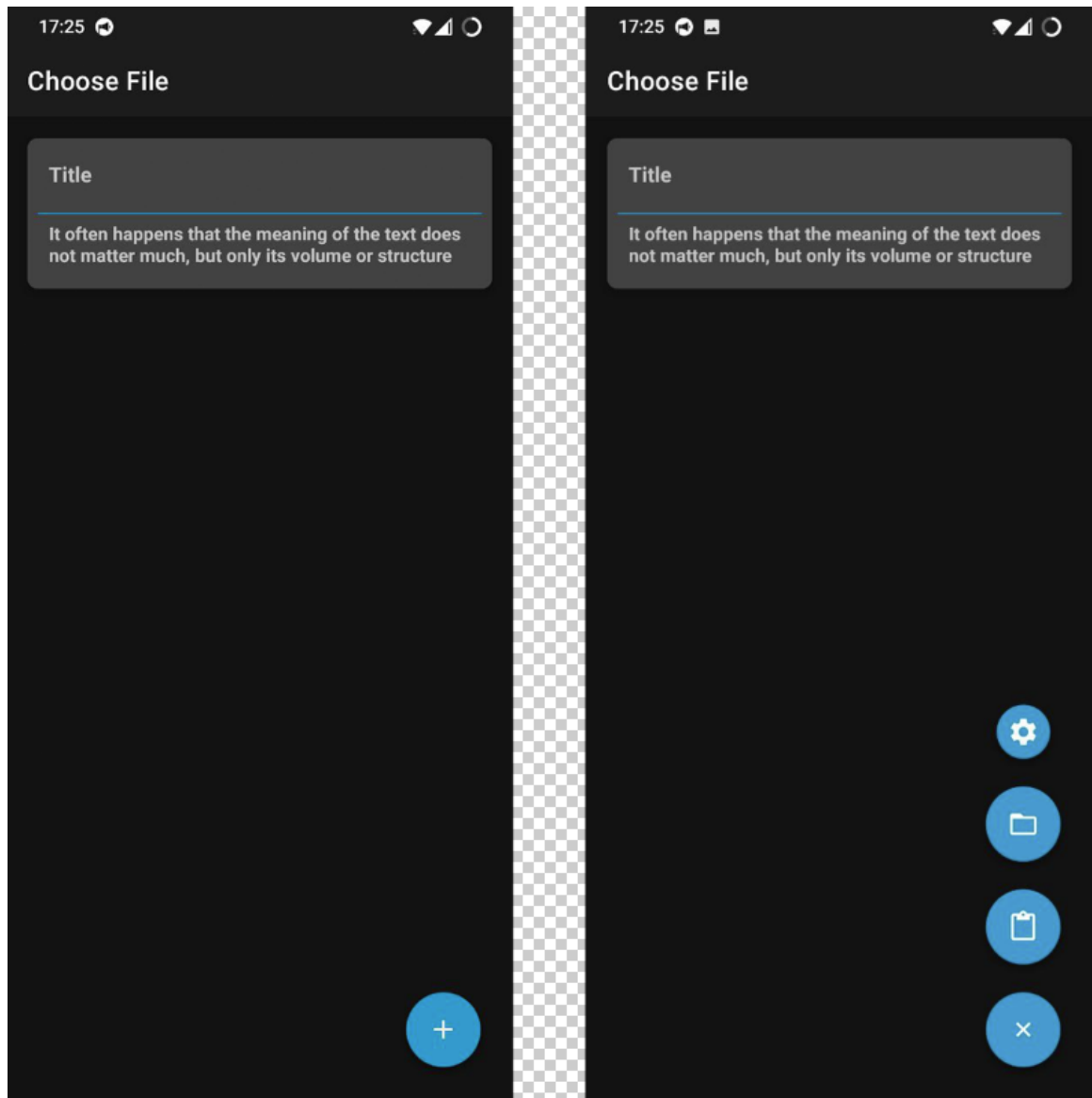


Рисунок 4.7 – Реалізація екрану вибору файлу для читання з неактивною та активною кнопкою навігації

Як видно з рисунку 4.4, комплексна кнопка навігації має 2 стани, активний та неактивний. Коли кнопка неактивна вона виконує лише функцію своєї активації.

Після натискання на кнопку, вона анімовано активується, та користувач має змогу скористатися однією з функцій кнопки:

- переходу до екрану налаштувань додатку;
- завантаження файлу з пристрою для читання або редагування;
- переходу до екрану створення нового текстового файлу для читання;
- деактивації.

При виборі функції переходу до будь-якого екрану, користувач бачить анімацію зміни активного екрану додатку на обраний, покадрове представлення переходу, зображене на рисунку 4.8

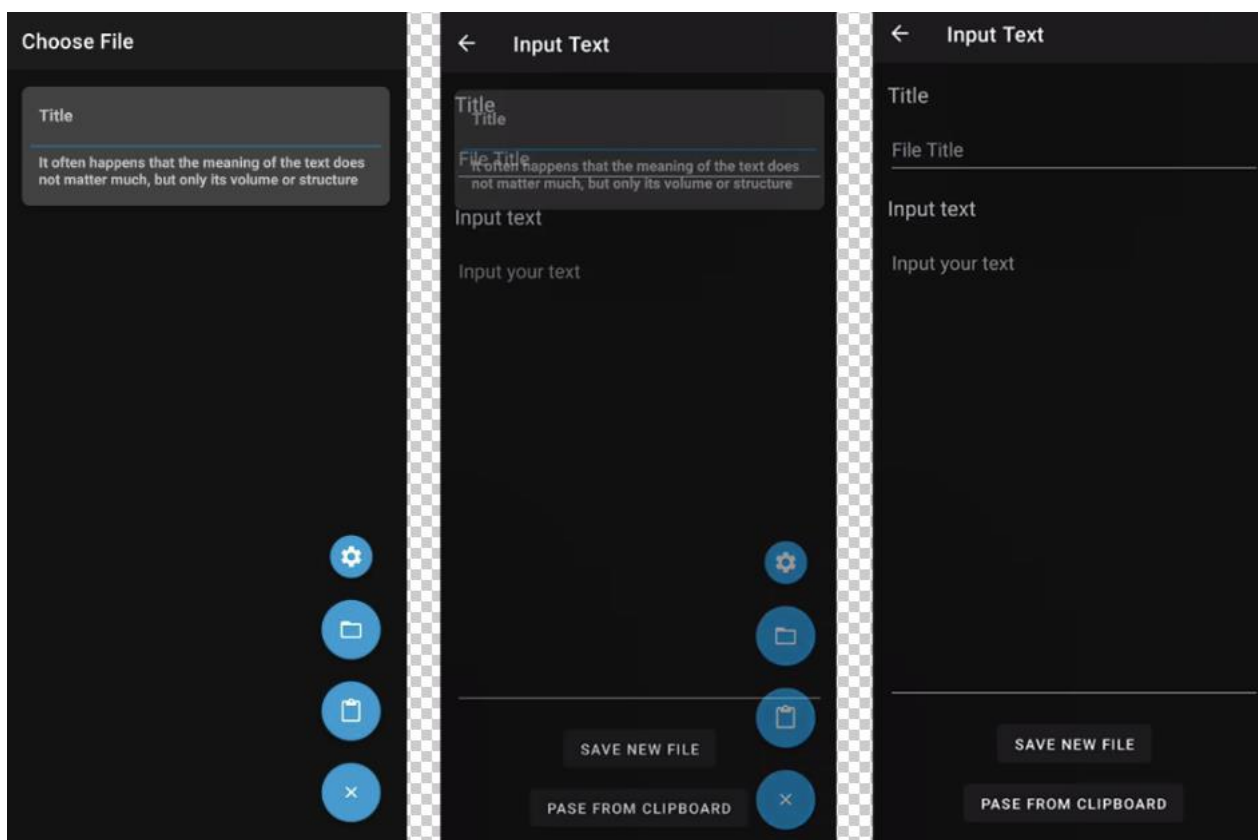


Рисунок 4.8 – Представлення анімації зміни активного екрану додатку

Фрагмент вибору файлу для читання включає в себе елемент «Card View». Елемент «Card View» - стандартний елемент додатків на базі операційної системи Android, який унаслідується від класу елемента розмітки «Layout». Тобто цей елемент слугує для розміщення у ньому дочірніх елементів.

Дочірніми у елемента є два текстових поля, та лінія поділу. Над лінією поділу розміщується заголовок або назва файлу, в залежності від типу додавання файлу у додаток, а під лінією поділу текстовим полем зображено частину першого речення певного файлу, фрагмент коду картки зображено на рисунку 4.9.

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:padding="8dp"
    app:cardCornerRadius="8dp"
    app:cardElevation="8dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/card1"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/cv_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:padding="8dp"
            android:text="Title"
            android:textSize="16sp"
            android:textStyle="bold"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <View
            android:id="@+id/divider"
            android:layout_width="0dp"
            android:layout_height="1dp"
            android:layout_marginStart="8dp"
            android:layout_marginEnd="8dp"
            android:background="@color/accent"
            app:layout_constraintBottom_toTopOf="@+id/cv_title" />
```

Рисунок 4.9 – Фрагмент XML код розмітки елемента «Card View»

Також на екрані вибору файлу знаходиться автоматично-заповнюваний список який складається з карток, які в свою чергу мають власну розмітку, графічну розмітку карток зображено на рисунку 4.10.

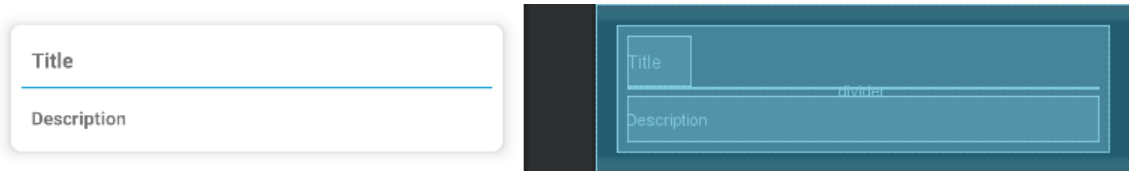


Рисунок 4.10 – Графічне представлення розмітки картки

Екран вводу тексту для читання має простий інтерфейс, що виконує свої функції, верхню частину екрану займає «App Bar» - стандартний елемент інтерфейсу пристроїв на базі операційної системи «Android», на якому знаходиться кнопка повернення до минулого екрану, зображена лівою стрілкою, та назву активного екрану.

Перше поле для вводу – поле для вводу заголовку файлу що створюється, інформація з цього поля вводу буде збережена у базі даних як «title» та використана на екрані вибору текстового файлу для читання.

Друге поле для вводу, це поле для вводу тексту створюваного файлу, обмеження по об'єму тексту створюваного файлу відсутнє, інформація з даного поля також буде направлена на збереження до бази даних.

Внизу екрану представлення знаходяться дві кнопки:

- перша для збереження нового файлу, інформація з полів введення зберігається у базу даних, для відображення на екрані вибору файлу для читання;
- друга для вставки скопійованого тексту, текст, заздалегідь скопійований користувачем до буферу обміну, вставляється до поля введення тексту.

Після наповнення полів представлення користувач має змогу зберегти файл у базу даних «Firebase», після збереження файлу, екран повертається до екрану вибору файлу.

Екран вводу текстового файлу зображено на рисунку 4.11.

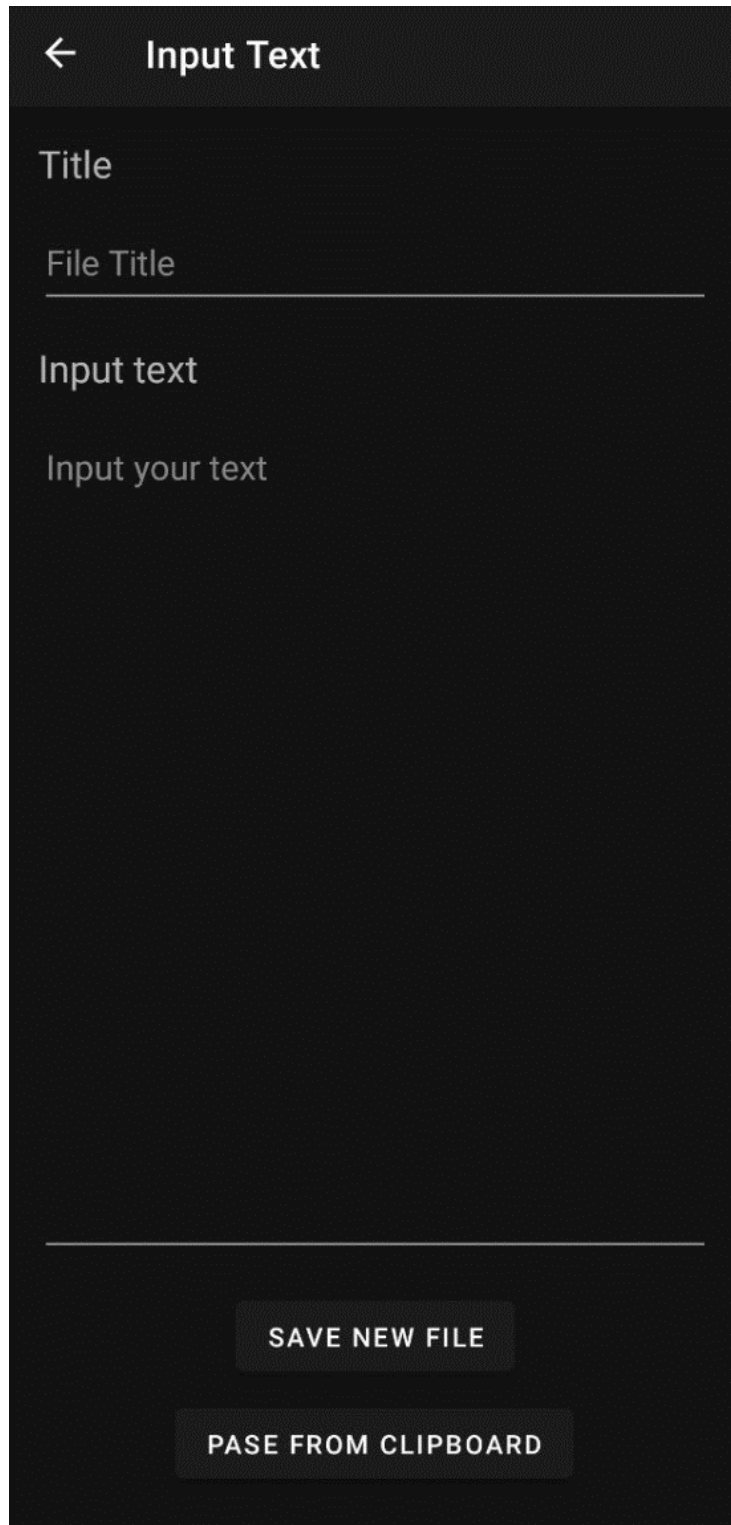


Рисунок 4.11 – Реалізація екрану вводу текстового файлу

При виборі, на екрані вибору текстового файлу для читання, функції переходу до екрану завантаження файлу, буде відкрито екран вибору текстового файлу для завантаження з пам'яті пристрою. Приклад реалізації екрану вибору файлу зображено на рисунку 4.12.

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		48

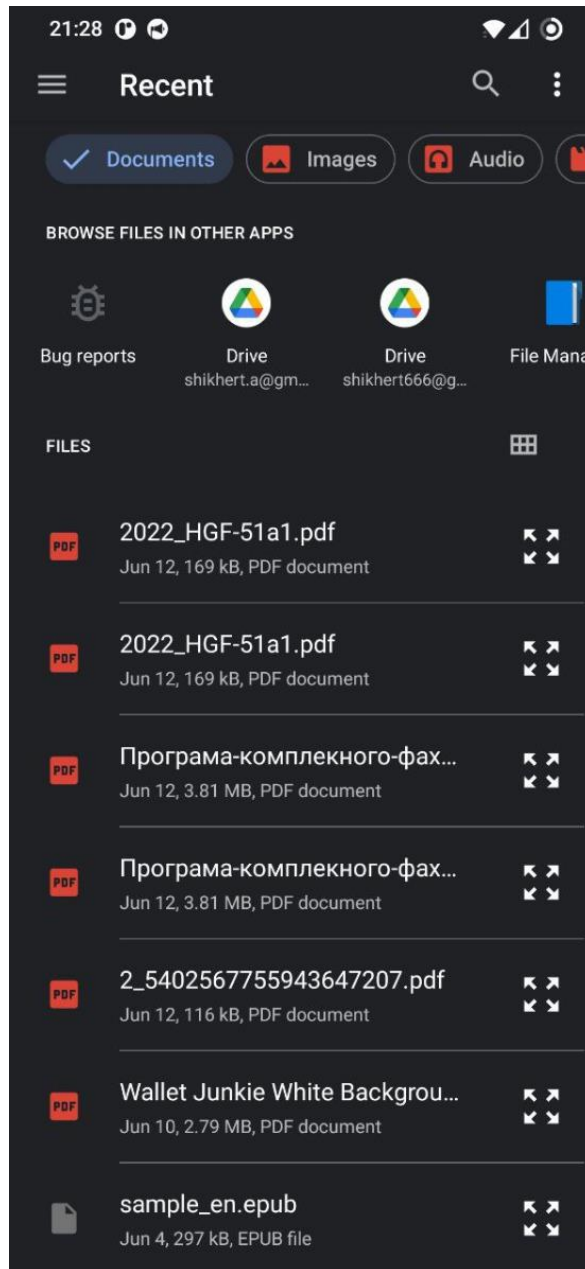


Рисунок 4.12 – Реалізація екрану вибору текстового файлу для завантаження з пам'яті пристрою

При доторканні до картки, на екрану вибору текстового файлу для читання, буде відкрито екран читання файлу. За умовчанням режим читання у мобільному додатку буде обрано «власний метод швидкого читання». Також з екрану читання, користувач має змогу перемкнутися на вікно налаштувань, дотиком до іконки, яка означає налаштування. Реалізація інтерфейсу екрану читання зображена на рисунку 4.13.

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		49

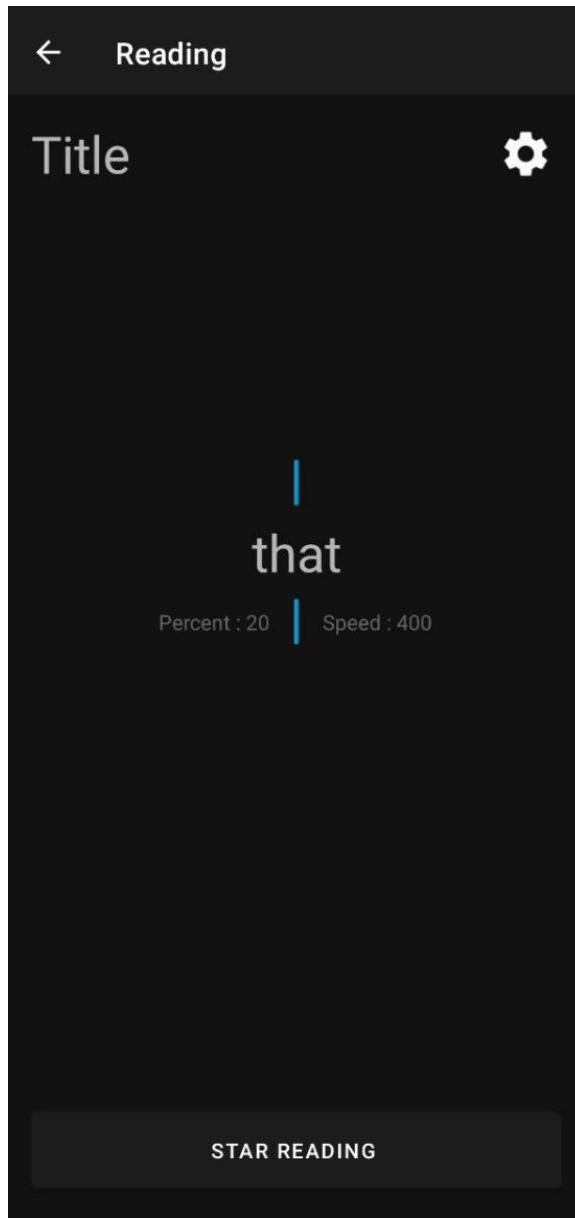


Рисунок 4.13 – Представлення реалізації екрану швидкого читання

При доторканні до швидкості читання, що знаходиться справа знизу від слова що читається, користувач має можливість змінити швидкість читання тексту.

Екран налаштування додатку, складається з наступних елементів:

- зміна кольорової теми додатку;
- зміна методу швидкого читання;
- зміна розміру шрифту, у вікні читання тексту;
- зміна акцентових кольорів.

Екран налаштувань зображено на рисунку 4.14.

					ІА81.300БАК.003 ПЗ	Лист
						50
Зм.	Лист	№ докум.	Підпис	Дата		

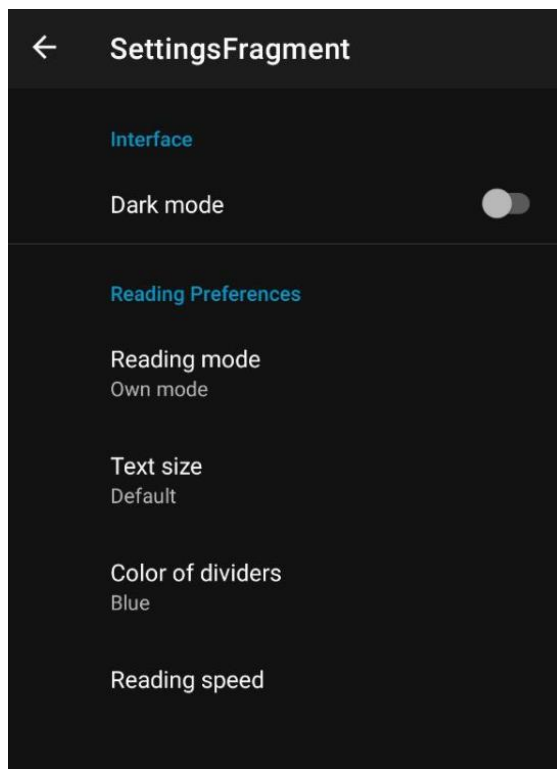


Рисунок 4.14 – Представлення реалізації екрану налаштувань

4.2 Програмна частина додатку

У розділі буде розглянуто створення програмної частини додатку та використаних технологій.

4.2.1 Основна діяльність

Діяльність (англ. Activity) - це єдина цілеспрямована річ, яку може робити користувач. Майже всі дії взаємодіють з користувачем, тому клас Activity подбає про створення вікна, в якому ви можете розмістити свій інтерфейс користувача[19].

До основної активності відноситься логіка навігації у додатку, та верхня панель навігації, створення навігації відбувається за використання компоненту «NavController», що значно спрощує виконання навігації у додатку, та використання AppBar – стандартного компоненту Android додатків. Для взаємодії з активними компонентами додатку необхідно щоб була змога

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		51

звертатись до цих компонентів з коду додатку, для цього був використаний DataBinding. Фрагмент коду основного екрану наведено на рисунку 4.15, інші екрани будуть вважатися дочірніми до нього.

```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var appBarConfiguration: AppBarConfiguration  
    private lateinit var binding: ActivityMainBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
  
        binding = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
        setSupportActionBar(binding.toolbar)  
  
        val navController = findNavController(R.id.nav_host_fragment_content_main)  
        appBarConfiguration = AppBarConfiguration(navController.graph)  
        setupActionBarWithNavController(navController, appBarConfiguration)  
    }  
  
    override fun onSupportNavigateUp(): Boolean {  
        val navController = findNavController(R.id.nav_host_fragment_content_main)  
        return navController.navigateUp(appBarConfiguration)  
            || super.onSupportNavigateUp()  
    }  
}
```

Рисунок 4.15 – Частина коду основної активності

4.2.2 Фрагмент вибору файлу

У фрагменті вибору файлу для читання відбувається взаємодія додатку з хмаровою базою даних, та у результаті додаток отримує інформацію про текстові файли, які має в себе включати представлення екрану вибору файлу.

Фрагмент коду звернення додатку до бази даних «Firebase Cloud Database» зображено на рисунку 4.16.

```

fun readFromDB() {
    textFilesModelArrayList = ArrayList()

    db.collection( collectionPath: "text_documents")
        .get()
        .addOnSuccessListener { result ->
            for (document in result) {
                textFilesModelArrayList?.add(TextFilesModel(document.data["title"] as String,
                    document.data["last"] as String))

                Log.d(TAG, msg: "readFromDB:${document.data["title"]} - ${document.data["last"]}")
            }
            Log.d(TAG, msg: "readFromDB: ${textFilesModelArrayList?.size}")
        }
        .addOnFailureListener { exception ->
            Log.w(TAG, msg: "Error getting documents.", exception)
        }
}
}

```

Рисунок 4.16 – Фрагмент коду звернення додатку до хмарової бази даних

Після отримання результатів звернення відбувається заповнювання списку з карток, елементами які місять в собі інформацію про назву текстового файлу, або заголовок, та коротке представлення першого речення файлу, довжиною в декілька слів, як це зображено на рисунку 4.4.

4.2.3 Фрагмент створення власного файлу

У цьому фрагменті користувач має змогу створити свій файл для читання, ввівши його заголовок, та сам текст що буде використовуватись у процесі читання. При цьому поля заголовку та наповнення текстом є обов'язковими для заповнення, при невиконанні цієї вимоги, файл не буде збережено.

На фрагменті знаходяться дві кнопки, перша для збереження файлу до бази даних, друга для вставки тексту з буферу обміну мобільного пристрою до поля введення тексту для читання, це зроблено для збільшення комфорту користування додатком, прототип інтерфейсу зображено на рисунку 4.11. Програмний код реалізації цього функціоналу зображено на рисунку 4.17.

```
private fun pasteText() {
    val myClipboard: ClipboardManager =
        activity?.getSystemService(CLIPBOARD_SERVICE) as ClipboardManager

    val pasteData: ClipData? = myClipboard.primaryClip
    val item = pasteData?.getItemAt(index: 0)
    val paste = item?.text.toString()

    binding.etInput.setText(paste, TextView.BufferType.EDITABLE)
}
```

Рисунок 4.17 – Програмний код реалізації функціоналу кнопки вставки тексту

4.2.4 Діяльність додавання файлу з пам’яті пристрою

Щоб додати файл для читання до додатку, необхідно дати змогу користувачеві вибрати, який саме файл користувач хоче завантажити, для цього у додаткові реалізовано діяльність вибору файлу для завантаження з пам’яті пристрою.

У цьому випадку реалізація екрану відбувається не фрагментом, тому що фрагмент не може втручатись до користувацького інтерфейсу операційної системи.

Для виклику діяльності завантаження використовуються методи наведені у таблиці 4.1.

Таблиця 4.1 – Опис методів завантаження файлу.

Назва методу	Призначення методу
startActivityResult()	Отримання інформацію від однієї діяльності до іншої
onActivityResult()	Обробка отриманих результатів від діяльності та перевірка їх наявності.

4.2.5 Фрагмент швидкого читання

У фрагменті швидкого читання, реалізовано основну функцію додатку, а саме швидке читання текстової інформації, за допомогою одного з методу що були розглянуті вище у науковому дослідницькому проєкті.

Фрагмент коду який використовується для власного методу читання, що встановлено як метод читання за умовчуванням зображено на рисунку 4.18.

```
binding.btnStartReading.setOnClickListener { it: View!

    val speech = reading_text
    val splitResult = speech.split(...delimiters: " ").toTypedArray()
    binding.tvWordToRead.text = splitResult[0]

    try {
        val handler = Handler()
        val thread = Thread {
            while (x < splitResult.size) {
                binding.btnStartReading.isClickable = false
                try {
                    handler.post { binding.tvWordToRead.text = (splitResult[x]) }
                    if (splitResult[x].length > 15) {
                        Thread.sleep(millis: readingSpeed + 100)
                    }
                    Thread.sleep(readingSpeed)
                    x++
                } catch (e: Exception) {
                    e.printStackTrace()
                }
            }
            if (x == splitResult.size) {
                binding.btnStartReading.isClickable = true
                x = 0
            }
        }
        thread.start()
    } catch (e: IOException) {
        e.printStackTrace()
    }
}
```

Рисунок 4.18 – Фрагмент коду реалізації методу швидкого читання

Також з рисунку 4.18, видно що слова довжина яких більша ніж 15 символів будуть відображатись на сто мілісекунд довше, щоб читач мав змогу розібрати це слово.

Висновки до розділу 4

У цьому розділі було створено логотип проєктованого додатку, та його прототип, за використання сервісу «Figma». Так як макет було створено з

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		55

урахуванням більшості нюансів створення додатку та з використанням бібліотеки стандартних компонентів операційної системи Android, то фінальний зовнішній вигляд користувацького інтерфейсу майже ідентичний до того що було запроєктовано.

Також у розділі було розглянуто повною мірою створення користувацького інтерфейсу мобільного додатку, кожен екран додатку було розібрано, та визначено ціль знаходження елементів на ньому.

Було розглянуто реалізацію програмної частини додатку, та наведено приклади коду, що використовувався під час створення мобільного додатку, також було розглянуто взаємодію додатку з хмаровою базою даних «Firebase Cloud Database».

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		56

5 ТЕСТУВАННЯ ТА АНАЛІЗ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Метою цього розділу є виявлення недоліків створеного мобільного додатку, та перевірка його працездатності

5.1 Адаптивність інтерфейсу

Тестування адаптивності екранів було проведено за допомогою середовища програмування «Android Studio», для цього було використано інструмент для роботи з інтерфейсом «Layout Validation», результати перевірки зображено у додатку А.1.

5.2 Тестування програмного забезпечення

У розділі розглянуто тестування мобільного додатку засобами тестування середовища програмування «Android Studio», та методом ручного тестування.

5.2.1 Тестування засобами тестування Android Studio

Під час тестування додатку засобом для тестування «Profiler», помилок чи нераціонального використання ресурсів тестованого девайсу не було виявлено.

Є невелике збільшення використання енергії під час запуску додатку, але всі показники знаходяться у допустових значеннях.

Також було проведено тестування використання інтернет підключення мобільного додатку, показники фонового інтернет підключення рівняються нулю, а при оновленні інформації з бази даних використання інтернету дорівнює приблизно 1Кб(показник може варіюватися в залежності від

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		57

кількості збережених файлів) та якості інтернет підключення. Графік тестування інтернет підключення зображено на рисунку 5.1.

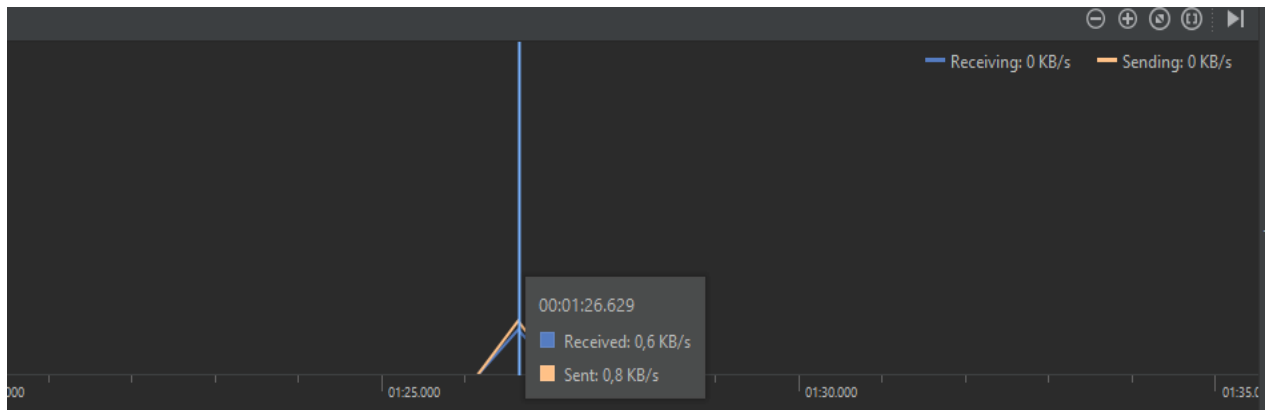


Рисунок 5.1 – графік тестування інтернет підключення

За умови відсутності інтернет з'єднання, додаток також можливо використовувати, а дані в хмаровій базі даних оновляться, коли пристрій буде підключено до інтернету.

5.2.2 Тестування ручним методом

Тестування ручним методом – процес технічного дослідження, коли додаток перевіряється на предмет помилок, тестувальником який відіграє роль користувача програми і знаходить помилки у його роботі.

Було проведено тести на анімації, та створено несприятливі для роботи умови, неочікувану поведінку користувача.

Після тестування цим методом було встановлено:

- на всіх екранах додатку анімація спрацювала правильно, конфліктів інтерфейсу виявлено не було;
- при зміні теми додатку інтерфейс відображається правильно;
- при відсутності інтернет з'єднання додаток працює і не виникає критичних помилок;

– якщо користувач виходить з додатку під час читання, то читання тексту припиняється, а помилок не виникає, після повернення до додатку, читання можливо продовжити.

Також було встановлено, що всі реалізовані функції читання, працюють правильно, та додаток зручно використовувати у буденному житті.

Висновки до розділу 5

У розділі описано процес тестування мобільного додатку. Тестування проводилось ручним методом та автоматичним. Для автоматичного тестування мобільного додатку було використано «Profiler», що є засобом для тестування вмонтованим до середовища розробки «Android Studio»

У результаті тестування не було виявлено критичних помилок, які б викликали передчасне закриття додатку, додаток є стабільним. Автоматичне тестування також не показало відхилень у використанні ресурсів мобільного пристрою, всі ресурси використовуються оптимально. Помилки у користувацькому інтерфейсі виявлено не було.

При ручному тестуванні одним з тестів, був тест використання додатку без інтернет підключення, та було встановлено що всі функції додатку працюють коректно, а дані в хмаровій базі даних оновляться щойно пристрій буде підключено до мережі інтернет.

Після проведення тестування додатку було встановлено, що додаток придатний до використання.

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		59

ВИСНОВКИ

В ході виконання дипломного проєкту було спроектовано та розроблено мобільний додаток для швидкого читання «SpeedRead».

Розробка мобільного додатку почалась з вивчення теорії читання, також було розглянуто низку методів швидкого читання, обрано методи які доцільно використовувати при створенні мобільного додатку, та розроблено власний метод швидкого читання, актуальність якого буде досліджено після релізу додатку, та початку використання його користувачами, для збору даних буде використано сервіс «Firebase», дані будуть збиратись анонімно та лише з дозволу користувача.

Також було розглянуто існуючі рішення додатків для швидкого читання і проаналізовано їх особливості, переваги деяких існуючих рішень було використано при створенні дипломного мобільного додатку.

На етапі проектування було створено наступні діаграми: діаграма розгортання, діаграма прецедентів, діаграма діяльності, та діаграма послідовностей які наведено у додатках до дипломного проєкту, та за використання сучасних засобів проектування розроблено прототип користувацького інтерфейсу.

Також було обрано мови програмування клієнтської та програмної частини додатку, які використовувалась під час реалізації, також детально розглянуто сервіс «Firebase» та його компоненти його хмарних послуг, доцільним було прийнято рішення використовувати наступні послуги: керованої серверної інфраструктури, релізу та моніторингу, супроводження.

Додаток має переваги над існуючими аналогами:

- можливість використання різних методів швидкого читання;
- можливість створення нових текстових файлів.

В останньому розділі було проведено тестування створеного мобільного додатку, та його відповідність заявленим функціональним вимогам. Під час

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		60

тестування не було виявлено критичних помилок, та помилок які б заважали використанню додатку.

Створений додаток є стабільним та відповідає вимогам розроблення, також мета створення додатку, була досягнута, додаток дає змогу використовувати переваги швидкого читання без спеціальних навичок, допомагає розвинути техніку читання та спрощує процес навчання цьому.

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		61

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. «Types of reading» URL: <https://schoolandtravel.com/types-of-reading/>
2. «Speed Reading» URL: <https://www.mindtools.com/speedrd.html>
3. «Прорив у швидкості читання» URL: <https://kniga.biz.ua/pdf/23122-Proriv-u-shvidkosti-chytannya.pdf>
4. «Повільне читання - Slow reading» URL: https://uk.wikiето.ru/wiki/Slow_reading#Research
5. «Прихильники скорочитання обіцяють, що за допомогою спеціальних технік кожен зможе швидко читати та запам'ятовувати об'ємні тексти.» URL: <https://lala.lanbook.com/4-tekhniki-skorochteniya-dlya-studentov>
6. «Швидка серійна візуальна презентація» URL: https://en.wikipedia.org/wiki/Rapid_serial_visual_presentation
7. «Spritz (Спиритз) - техніка быстрого чтения» URL: <http://magicspeedreading.com/k/spiritz.html>
8. «Reedy. Intelligent reader» URL: <https://play.google.com/store/apps/details?id=azagroup.reedy&hl=ru&gl=US>
9. «Speed reading» URL: <https://play.google.com/store/apps/details?id=com.speedreading.alexander.speedreading&hl=ru&gl=US>
- 10.«Speed Reader» URL: <https://play.google.com/store/apps/details?id=com.vanniktech.speedreading&hl=ru&gl=US>
- 11.«Balto Speed Reading» URL: <https://progsoft.net/ru/software/balto-speed-reading>
- 12.«Balto Speed Reading» URL: <https://www.amazon.com/Balto-Wolf-Studio-Speed-Reading/dp/B00J21HNOO>
- 13.«Kotlin» URL: <https://uk.wikipedia.org/wiki/Kotlin>
- 14.«XML» URL: <https://uk.wikipedia.org/wiki/XML>
- 15.«Android Studio» URL: <https://developer.android.com/studio>

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		62

16.«Firebase» URL: <https://en.wikipedia.org/wiki/Firebase>

17.«Gradle» URL: <https://uk.wikipedia.org/wiki/Gradle>

18.«Material Design» URL: <https://material.io/develop/android>

19.«Activity» URL: <https://developer.android.com/reference/android/app/Activity>

20.«App bars: top» URL : <https://material.io/components/app-bars-top>

					ІА81.300БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		63

Додаток А

Результати тестування користувацького інтерфейсу на екранах мобільних пристроїв

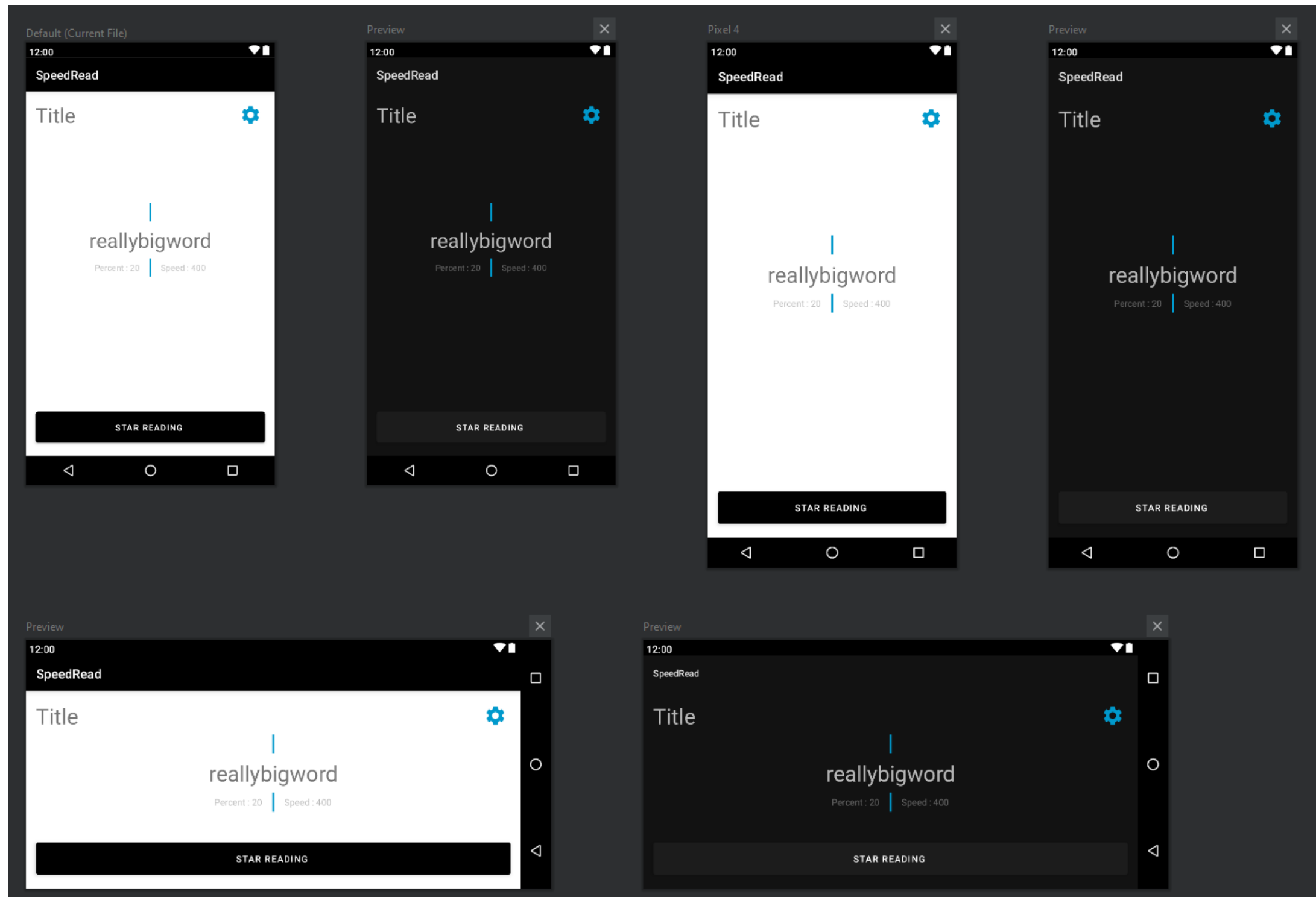


Рисунок А1 – представлення реалізації користувацького інтерфейсу на різних мобільних пристроях