

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

(повна назва інституту/факультету)

кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри БМК

Світлана АЛХІМОВА

(підпис)

(ініціали, ПРІЗВИЩЕ)

“ ” червня 2025р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою

«Комп'ютерні технології в біології та медицині»

зі спеціальності 122 «Комп'ютерні науки»

на тему: програмний застосунок для перевірки гостроти зору в польових і домашніх умовах

Виконав: студент IV курсу, групи БС-12

ОЛЕЙНИКОВ РОСТИСЛАВ СЕРГІЙОВИЧ

(прізвище, ім'я, по батькові)

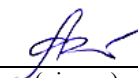


(підпис)

Науковий керівник: професор каф. біомедичної кібернетики (БМК)

проф., д.т.н., Файнзільберг Леонід Соломонович

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



(підпис)

Консультант з розділів дипломної роботи:

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент: доцент каф. біомедичної інженерії, доцент, к.т.н.

Богомолів Микола Федорович

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент



(підпис)

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
 Інститут (факультет) БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

(повна назва)

Кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні технології в біології та медицині»

(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри БМК

_____ Світлана АЛХІМОВА

« ____ » квітень 2025 р.

ЗАВДАННЯ на дипломну роботу студенту

ОЛЕЙНИКОВУ РОСТИСЛАВУ СЕРГІЙОВИЧУ

(прізвище, ім'я, по батькові)

1. Тема роботи програмний застосунок для перевірки гостроти зору в польових і домашніх умовах

керівник роботи

проф., д.т.н., Файнзільберг Леонід Соломонович

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

затверджені наказом по університету від « 26 » травня 2025 р. № 1758-с

2. Термін подання студентом роботи

22-23 травня 2025 року

3. Вихідні дані: Не передбачено

4. Зміст роботи: Проаналізувати джерела та існуючі рішення щодо перевірки гостроти зору. Визначити вимоги до системи для домашнього та польового використання. Обґрунтувати вибір архітектури та засобів розробки. Реалізувати мобільний застосунок із функціями перевірки зору та збереження результатів. Провести тестування розробленої системи за участю користувачів. Проаналізувати ефективність роботи системи, оцінити її економічну доцільність. Сформулювати загальні висновки та надати

рекомендації для подальшого розвитку проєкту.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо).

6. Консультанти розділів роботи^{1*}

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Дипломної роботи			

7. Дата видачі завдання **06 квітня 2025р.**

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримати завдання за темою ДР на практику	До 24.12.2024р.	виконано
2	Переддипломна практика. Виконання практичної частини ДР та додаткових розділів	1-4 тиждень за графіком практики	виконано
3	Виконання основних розділів ДР (Вступ, огляд існуючих рішень та нормативної бази, засоби реалізації та запропоновані рішення, обґрунтування вибору програмного середовища та архітектури, програмна реалізація застосунку, тестування, збереження даних та конфіденційність)	Протягом усієї практики	виконано
4	Перевірка ДР науковим керівником	до 20.05.2025	виконано
5	Подання в електронному вигляді ДР на перевірку нормоконтролера та плагіат (StrikePlagiarism.com). Доопрацювання ДР	Не пізніше 23.05.2025	виконано
6	Отримати відгук від керівника ДР	Не пізніше 05.06.2025	виконано
7	Надати пакету документів по ДР та супровідних до неї документів на засідання кафедри		виконано
8	Подання ДР рецензенту. Отримання рецензії.	09.06 – 13.06.2025	виконано
9	Захист ДР в ЕК	16.06 - 21.06.2025	

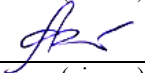
Студент


(підпис)

Ростислав ОЛЕЙНІКОВ

(ім'я, ПРІЗВИЩЕ)

Керівник ДР


(підпис)

Леонід ФАЙНЗІЛЬБЕРГ

(ім'я, ПРІЗВИЩЕ)

Нормоконтролер

(підпис)

Галина КОРНІЄНКО

(ім'я, ПРІЗВИЩЕ)

^{1*} Якщо визначені консультанти. Консультантом не може бути зазначено наукового керівника магістерської дисертації.

АНОТАЦІЯ

Дипломна робота за темою «Програмний застосунок для перевірки гостроти зору в польових і домашніх умовах» виконана студентом кафедри біомедичної кібернетики ФБМІ *Олейниковим Ростиславом Сергійовичем* зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині» та складається зі: вступу; 5 розділів, висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 76 джерела, та додатків. Загальний обсяг роботи 74 сторінок.

Актуальність теми. Актуальність дипломної роботи пов'язана із задачею створення мобільного застосунку для перевірки гостроти зору в польових і домашніх умовах. Розробка такого інструменту спрямована на своєчасне виявлення порушень зору, що дозволяє знизити навантаження на офтальмологічні служби та підвищити доступність діагностики для населення в умовах віддалених або неурбанізованих територій. Використання платформи Android і мови Java, інтеграція голосових підказок (TTS) та захищеного локального зберігання результатів у облікових записках забезпечує зручність експлуатації, безпеку даних і широку сумісність із різними пристроями. Реалізація цієї системи відповідає сучасним трендам телемедицини та цифровізації охорони здоров'я, сприяючи підвищенню якості життя та оптимізації ресурсів медичної галузі.

Мета і завдання роботи.

Метою роботи є вдосконалення процесу самостійної первинної діагностики зору в домашніх та польових умовах шляхом розробки мобільного програмного забезпечення на платформі Android, що забезпечує проведення інтерактивних тестів на гостроту зору, аметропію, кольоросприйняття та астигматизм.

Її досягнення передбачає вирішення наступних завдань:

1. Проаналізувати джерела та існуючі рішення щодо перевірки гостроти зору.
2. Визначити вимоги до системи.
3. Обґрунтувати вибір архітектури та засобів розробки.
4. Реалізувати мобільний застосунок із функціями перевірки зору та збереження результатів.
5. Проаналізувати ефективність роботи системи, оцінити її економічну доцільність.
6. Сформулювати загальні висновки та надати рекомендації для подальшого розвитку проєкту.

Використані методи: огляд існуючих рішень та нормативної бази; аналіз наявних наукових та практичних джерел; моделювання архітектури та вибір технологій розробки; застосування механізмів локальної аутентифікації та шифрування даних; інтеграція голосових (TTS) сервісів для підвищення доступності.

Отримані результати: програмний застосунок для перевірки зору в польових і домашніх умовах.

Апробація результатів роботи - не заплановано

Публікації - не заплановано.

Бібліографічний опис ДР

Олейніков Р. С. Програмний застосунок для перевірки гостроти зору в польових і домашніх умовах: дипломна роб. бакалавра : 122 Комп'ютері науки / Олейніков Ростислав Сергійович. – Київ, 2025. – 74 с.

ANNOTATION

The thesis on the topic "Software application for testing visual acuity in field and home conditions" was completed by a student of the Department of Biomedical Cybernetics of the Federal State Institute of Medical Sciences, Rostyslav Serhiyovych Oleinikov, majoring in 122 "Computer Science" under the educational and professional program "Computer Technologies in Biology and Medicine" and consists of: introduction; 4 sections, conclusions to each of these sections; general conclusions; a list of sources used, which includes 76 sources, and appendices. The total volume of the work is 74 pages.

Relevance of the topic. The relevance of the thesis is associated with the task of creating a mobile application for testing visual acuity in field and home conditions. The development of such a tool is aimed at the timely detection of visual impairments, which allows reducing the load on ophthalmological services and increasing the availability of diagnostics for the population in remote or non-urbanized areas. The use of the Android platform and the Java language, the integration of voice prompts (TTS) and secure local storage of results in accounts provides ease of use, data security and broad compatibility with various devices. The implementation of this system meets modern trends in telemedicine and healthcare digitalization, contributing to improving the quality of life and optimizing medical resources.

Purpose and objectives of the work.

The purpose of the work is to improve the process of independent primary vision diagnostics in home and field conditions by developing mobile software on the Android platform, which provides interactive tests for visual acuity, ametropia, color perception and astigmatism.

Its achievement involves solving the following tasks:

1. Analyze sources and existing solutions for testing visual acuity.
2. Determine system requirements.
3. Justify the choice of architecture and development tools.
4. Implement a mobile application with vision testing functions and saving

results.

5. Analyze the efficiency of the system, assess its economic feasibility.

6. Formulate general conclusions and provide recommendations for further development of the project.

Methods used: review of existing solutions and regulatory framework; analysis of available scientific and practical sources; architecture modeling and selection of development technologies; application of local authentication mechanisms and data encryption; integration of voice (TTS) services to increase accessibility.

Results obtained: software application for testing vision in field and home conditions.

Approbation of work results: not planned

Publications: not planned.

ЗМІСТ

АНОТАЦІЯ.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	11
ВСТУП.....	12
РОЗДІЛ 1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА НОРМАТИВНОЇ БАЗИ ПРОГРАМНОГО ЗАСТОСУНКУ	14
1.1 Аналоги програмних продуктів для тестування зору.....	14
1.1.1 Мобільні застосунки.....	14
1.1.2 Веб-платформи та апаратно-програмні комплекси	16
1.2 Методики визначення основних характеристик зору.....	18
1.3 Огляд літературних джерел і стандартів МОЗ.....	19
Висновок до розділу 1	20
РОЗДІЛ 2 ЗАСОБИ РЕАЛІЗАЦІЇ ТА ЗАПРОПОНОВАНІ РІШЕННЯ.....	21
2.1 Вибір мов, середовищ і бібліотек	21
2.2 Моделі даних, формати, шкали та кольорові стандарти	24
2.2.1 Шкала logMAR.....	24
2.2.2 Таблиці Ішихари	25
2.2.3 Радіальні решітки.....	28
2.2.4 JSON формат даних	29
2.3 Принципи UX / UI для офтальмологічних тестів	29
2.4 Вибір і обґрунтування алгоритмів обробки результатів	30
2.4.1 Алгоритм для розрахунку гостроти зору.....	30
2.4.1 Алгоритм визначення аметропії.....	31
2.4.2 Алгоритм визначення дальтонізм	32
2.4.2 Алгоритм визначення дальтонізм	33
Висновок до розділу 2	33
РОЗДІЛ 3 ОБГРУНТУВАННЯ ВИБОРУ ПРОГРАМНОГО	

СЕРЕДОВИЩА ТА АРХІТЕКТУРИ	35
3.1 Порівняння альтернативних платформ	35
3.1.1 Flutter	35
3.1.2 React Native	35
3.1.3 Android Studio з Java	36
3.2 Архітектурний стиль (MVVM) та схема модулів	38
3.2.1 Архітектурний стиль	38
3.2.2 Модулі	38
3.2.3 MVVM	39
3.3 Механізми локальної аутентифікації та шифрування даних	40
3.4 Вибір стратегії масштабування й підтримки версій Android	41
Висновок до розділу 3	42
РОЗДІЛ 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ	44
4.1 Реалізація інтерфейсу користувача	44
4.2 Реалізація тесту на гостроту зору та аметропію	50
4.3 Реалізація тесту на дальтонізм	53
4.4 Реалізація тесту на астигматизм	55
4.5 Реалізація збереження результатів	57
4.6 Реалізація адаптації розмірів символів	59
4.7 Розрахунок економічного ефекту	60
Висновок до розділу 4	62
РОЗДІЛ 5 ТЕСТУВАННЯ, ЗБЕРЕЖЕННЯ ДАНИХ ТА	
КОНФІДЕНЦІЙНІСТЬ	63
5.1 Тестування функціональності та точності	63
5.2 Організація збереження результатів тестування	64
5.3 Захист персональних даних	65
Висновок до розділу 5	65
ЗАГАЛЬНІ ВИСНОВКИ	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД - База даних, це структурована колекція даних, яка зберігається та управляється комп'ютерною системою [76].

ІС - інформаційна система.

ІТ - інформаційні технології.

Гострота зору - характеристика здатності ока розрізняти дрібні деталі об'єктів, яка залежить від анатомічного та функціонального стану зорової системи [71].

Аметропія - порушення заломлення світла в оці, що призводить до нечіткого зображення (наприклад, короткозорість або далекозорість) [72].

Дальтонізм - спадкове порушення кольорового зору, при якому людина частково або повністю не розрізняє окремі кольори [70].

Астигматизм - порушення форми рогівки або кришталіка, що спричиняє викривлення зображення [72].

Голосові підказки - елемент інтерфейсу, що забезпечує аудіоінструкції для покращення доступності програми.

ВСТУП

Актуальність теми. Актуальність дипломної роботи пов'язана із задачею створення мобільного застосунку для перевірки гостроти зору в польових і домашніх умовах. Розробка такого інструменту спрямована на своєчасне виявлення порушень зору, що дозволяє знизити навантаження на офтальмологічні служби та підвищити доступність діагностики для населення в умовах віддалених або неурбанізованих територій. Використання платформи Android і мови Java, інтеграція голосових підказок (TTS) та захищеного локального зберігання результатів у облікових записах забезпечує зручність експлуатації, безпеку даних і широку сумісність із різними пристроями. Реалізація цієї системи відповідає сучасним трендам телемедицини та цифровізації охорони здоров'я, сприяючи підвищенню якості життя та оптимізації ресурсів медичної галузі.

Мета і завдання роботи

Метою роботи є вдосконалення процесу самостійної первинної діагностики зору в домашніх та польових умовах шляхом розробки мобільного програмного забезпечення на платформі Android, що забезпечує проведення інтерактивних тестів на гостроту зору, аметропію, кольоросприйняття та астигматизм.

Її досягнення передбачає вирішення наступних *завдань*:

1. Аналіз вітчизняних та зарубіжних джерел.
2. Проаналізувати джерела та існуючі рішення щодо перевірки гостроти зору.
3. Визначити вимоги до системи.
4. Обґрунтувати вибір архітектури та засобів розробки.
5. Реалізувати мобільний застосунок із функціями перевірки зору та збереження результатів.
6. Проаналізувати ефективність роботи системи, оцінити її економічну доцільність.

7. Сформулювати загальні висновки та надати рекомендації для подальшого розвитку проєкту.

Використані методи: огляд існуючих рішень та нормативної бази; аналіз наявних наукових та практичних джерел; моделювання архітектури та вибір технологій розробки; застосування механізмів локальної аутентифікації та шифрування даних; інтеграція голосових (TTS) сервісів для підвищення доступності.

Отримані результати: програмний застосунок для перевірки зору в польових і домашніх умовах.

Апробація результатів роботи - не заплановано

Публікації - не заплановано.

Структура роботи

Дипломна робота за темою «Програмний застосунок для перевірки гостроти зору в польових і домашніх умовах» виконана студентом Олейниковим Ростиславом Сергійовичем зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині», побудована за класичним типом та викладена на 54 сторінках машинописного тексту. Вона складається з: вступу; 5 розділів, висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 72 джерел та додатків. В роботі представлено 30 рисунків і 3 таблиць.

РОЗДІЛ 1

ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА НОРМАТИВНОЇ БАЗИ ПРОГРАМНОГО ЗАСТОСУНКУ

В розділі оглядається сучасний ринок офтальмологічних рішень: від простих мобільних застосунків до професійних апаратно-програмних комплексів. Аналізуються алгоритми Snellen, logMAR, Ішихара, відповідність GDPR, HIPAA та українським медичним протоколам, а також викристалізовується їхній внесок для майбутнього застосунку.

1.1 Аналоги програмних продуктів для тестування зору

У сучасному світі, де технології проникають у всі сфери нашого життя, не дивно, що і питання здоров'я, зокрема зору, все частіше вирішуються за допомогою мобільних та веб-рішень. Здавалося б, навіщо йти до лікаря, якщо можна швидко перевірити зір вдома? Саме тому ринок рясніє різноманітними програмними продуктами, що обіцяють легку та швидку оцінку стану нашого зору.

1.1.1 Мобільні застосунки

Особливо цікавими в цьому плані є мобільні застосунки. Наприклад, компанія EyeQue, здається, задалася метою зробити перевірку зору доступною кожному. Їхній продукт EyeQue Insight Lite, як вони заявляють, є кишеньковим помічником для контролю здоров'я очей, що дозволяє перевіряти гостроту зору та контрастну чутливість, маючи під рукою лише смартфон [24]. Лише телефон і ви вже знаєте, чи добре бачите літеру "Е" в різних напрямках. Клінічні випробування нібито підтверджують надійність цих тестів. Хоча, звісно, виникає питання: чи можна повністю довіряти такому тесту замість візиту до кваліфікованого офтальмолога? Розробники самі застерігають, що це лише скринінг, а не діагноз. Проте, для первинної оцінки, можливо, це й непоганий варіант [13].

А ось інший їхній продукт, VisionCheck, вже виглядає серйозніше. Це вже не просто застосунок, а портативний пристрій, який, за словами розробників, поєднує в собі функції авторефрактометра та фороптера. Звучить досить складно для звичайної людини [51]. Але обіцяють точні вимірювання зору за лічені хвилини. Кажуть, їхня запатентована технологія VisionCheck навіть пройшла клінічні випробування і використовується по всьому світу. Оригінальний EyeQue Personal Vision Tracker (PVT) взагалі був рефрактометром, що визначав короткозорість, далекозорість та астигматизм, і навіть видавав результати у форматі рецепта. VisionCheck пішов далі, додавши сенсори, Bluetooth та моторизовані лінзи. В основі їхньої технології лежить принцип оберненого Шека-Гартмана. Чи справді це так само точно, як у лікаря? Можливо, для когось це зручна альтернатива, особливо якщо потрібно швидко оновити рецепт [51].

Ще один цікавий гравець на цьому ринку – застосунок Peek Acuity. Його розробили фахівці з офтальмології для вимірювання гостроти зору [44]. Особливістю цього додатку є використання тесту з "падаючою E", що робить його доступним навіть для тих, хто не вміє читати [59]. Результати подаються у звичних одиницях Снеллена та LogMAR [44]. І що важливо, Peek Acuity пройшов клінічну валідацію, яка показала його точність, порівнянну зі стандартними методами [45]. До того ж, розробники запевняють, що не збирають жодних персональних даних [44]. Цікаво, що цей додаток розроблено для тестування зору іншої людини [45]. Можливо, його першочерговою метою було використання медичними працівниками під час виїзних оглядів.

Крім цих трьох, існує безліч інших мобільних застосунків для перевірки зору. Вони пропонують різноманітні функції: від перевірки гостроти зору до тестів на дальтонізм та астигматизм, а деякі навіть намагаються виявити ознаки серйозних захворювань, таких як макулярна дегенерація [39]. Деякі додатки пропонують навіть вправи для очей [60]. Звісно, виникає питання про їхню точність та надійність [4]. Ринок пропонує

як безкоштовні, так і платні варіанти, а також ті, що працюють у парі зі спеціальним обладнанням.

1.1.2 Веб-платформи та апаратно-програмні комплекси

Окрім мобільних застосунків, існують і веб-платформи для тестування зору. Серед них варто згадати Lens.com, 1-800 Contacts, ContactsDirect, Discount Contact Lenses, LensDirect, WebEyeCare, AC Lens [55]. Багато з них пропонують онлайн-перевірку зору для оновлення рецепта на контактні лінзи або окуляри [55]. Деякі, як ZEISS та Essilor, навіть пропонують безкоштовні скринінгові тести [54]. Цікаво, що Stanford Acuity Test (StAT) використовує штучний інтелект для більш точного вимірювання гостроти зору вдома [58]. Зручність таких платформ очевидна, але чи можуть вони замінити візит до лікаря? Мабуть, ні [55].

Що ж до апаратно-програмних комплексів, то тут вже йдеться про професійне обладнання, яке використовується в клініках та кабінетах офтальмологів. Прикладами можуть бути Amplivox [43], M&S Technologies Smart System, Acuity Pro, Waggoner Diagnostics та Welch Allyn Spot Vision Screener [43, 52, 3, 63, 65]. Ці системи відрізняються високою точністю та пропонують широкий спектр тестів. Багато з них інтегруються з електронними медичними записами [52]. Існують також портативні скринери, такі як Welch Allyn Spot Vision Screener, що зручно для швидких оглядів, особливо у дітей [65]. А ще є портативні рефрактометри, наприклад Netra від EyeNetra Inc. та SVOne, які використовують смартфони для вимірювання рефракції [47].

Порівняльна таблиця (табл. 1.1) показує, що ринок мобільних і гібридних рішень для перевірки зору можна умовно поділити на три групи. Лише-застосунок (зазвичай безкоштовно, але менше тестів), застосунок + насадка (точніше, проте дорожче) й клінічні платформи з телемедичною орієнтацією (найповніші, та вимагають серверного зберігання). Розроблений застосунок поєднує плюси перших двох: працює без додаткового обладнання й підписок, охоплює всі чотири ключові тести та зберігає дані локально у

шифрованому вигляді, що робить його найвигіднішим компромісом між зручністю, точністю й конфіденційністю.

Таблиця 1.1

Порівняльна таблиця аналогів

Продукт	Платформа / обладнання	Основні тести	Підтвердження точності та регуляторний статус	Конфіденційність / зберігання даних	Модель ціноутворення
Peek Acuity	Android-смартфон	Візуальна гострота (Tumbling E)	Клінічна еквівалентність паперовій ETDRS-таблиці	Дані тесту зберігаються локально; без акаунта	Безкоштовно
EyeQue Insight Lite	Смартфон + картонний візир	Далекозорість / короткозорість	Призначений для швидкого скринінгу	Результати зберігаються офлайн у додатку; без персональних медичних даних у хмарі	18 доларів одноразово
EyeQue VisionCheck 2	Смартфон + Bluetooth-пристрій	Рефракція	Зареєстрований у FDA як Class I 510(k) exempt MD; точність ± 0.5 D	Підключення до EyeQue Cloud	Пристрій 19 доларів
Smart Optometry	iOS / Android	15 тестів: VA, кольорова чутливість, контраст, Worth 4 Dot тощо	Розробник декларує клінічну точність; використовується фахівцями у > 100 країн	Збір контактних та аналітичних даних	Базова версія безкоштовна; PRO — підписка \$2.99/міс
SightBook	iOS / Android	Близька VA, контраст, моніторинг макули	Точність близька до стандартних методів у 200 пацієнтів (Bland-Altman) (PMC)	Дані зберігаються на сервері Stanford Byers з анонімізацією	Безкоштовно
Eye Chart Pro	iPad	Snellen, Tumbling E, ETDRS, Amsler	Кілька рецензованих досліджень показали середнє відхилення 0.02	Додаток не збирає персональні дані	Безкоштовно
Essilor “My Vision Check” (онлайн)	Веб-браузер (ПК/мобільний)	Дистанційна VA, контраст, кольори	Компанія позиціонує як попередній скринінг, не замінює офтальмолога	Тести не зберігають медичні дані; працюють без реєстрації (Essilor Vision Test)	Безкоштовно
Розроблений	Андроїд	Дистанційна VA,	Позиціонує як попередній	Дані тесту зберігаються	Безкоштовно

застосунок		аметропія, кольори, астигматизм	скринінг, не замінює офтальмолога	локально з акаунтом	
------------	--	---------------------------------------	---	------------------------	--

1.2 Методики визначення основних характеристик зору

Щоб оцінити наш зір, лікарі використовують різні методики. Основні з них спрямовані на визначення гостроти зору, наявності короткозорості чи далекозорості (аметропії), здатності розрізняти кольори та наявності астигматизму.

Гостроту зору зазвичай перевіряють за допомогою таблиць Снеллена, де потрібно розрізняти літери або символи різного розміру з певної відстані [13]. Результат записують у вигляді дробу, наприклад, 20/20, що вважається нормою. У мобільних застосунках часто використовують спрощені варіанти, наприклад, тест з "падаючою Е" [14].

Для визначення аметропії (короткозорості та далекозорості) вимірюють рефракцію ока за допомогою спеціальних приладів – рефрактометрів. Деякі сучасні мобільні додатки, як EyeQue VisionCheck, також вміють це робити [51].

Дальтонізм (порушення колірного сприйняття) перевіряють за допомогою таблиць Ішихари. Це такі кольорові картинки з крапочок, на яких зображені цифри або фігури, які бачать люди з нормальним зором, але не бачать ті, хто страждає на певні форми дальтонізму. Існують також онлайн-тести та мобільні додатки для перевірки колірного зору [51].

Для виявлення астигматизму використовують спеціальні тести, наприклад, з радіальними решітками [51]. Людина дивиться на зображення з лініями, що розходяться від центру, і при астигматизмі деякі лінії можуть здаватися розмитими або викривленими. Деякі мобільні додатки також пропонують подібні тести [12].

Цікаво, що при розробці застосунку для перевірки зору варто згадати про принцип Вебера-Фехнера. Цей закон описує, як ми сприймаємо зміни

інтенсивності подразника. У візометрії це означає, що для того, щоб ми помітили зміну розміру літери, ця зміна має бути досить значною, пропорційною до початкового розміру. Саме тому в таблицях Снеллена розмір літер зменшується не лінійно, а за логарифмічною шкалою. Врахування цього принципу може зробити тести на гостроту зору в нашому застосунку більш точними та науково обґрунтованими [56, 38].

1.3 Огляд літературних джерел і стандартів МОЗ

Розробка медичного програмного забезпечення – справа відповідальна. Тому необхідно ретельно вивчити відповідні літературні джерела та нормативну базу.

В офтальмології існують загальноприйняті клінічні протоколи та рекомендації щодо проведення обстежень зору. Вони визначають стандартизовані методики оцінки різних параметрів зору. Наш застосунок повинен базуватися на цих рекомендаціях, щоб його результати можна було порівняти з результатами традиційних обстежень.

Якщо ми колись захочемо інтегрувати наш застосунок з іншими медичними системами, нам потрібно буде враховувати стандарти обміну медичними даними, такі як DICOM та HL7 FHIR. DICOM використовується для обміну медичними зображеннями, а HL7 FHIR – для обміну електронними медичними записами. Хоча зараз ми зберігаємо дані локально, розуміння цих стандартів може бути корисним у майбутньому [1, 15].

Оскільки наш застосунок зберігає персональні дані користувачів, ми повинні забезпечити відповідність вимогам законодавства про захист персональних даних, таким як GDPR та HIPAA. Ці закони регулюють збір, зберігання та обробку персональних даних, включаючи медичну інформацію. Ми повинні вжити всіх необхідних заходів для захисту даних наших користувачів [19, 21].

Для законного використання нашого застосунку в Україні нам потрібно

буде врахувати відповідні накази, стандарти та форми Міністерства охорони здоров'я України [75].

Висновок до розділу 1

Огляд існуючих рішень показує, що існує багато програм для тестування зору, але якість та надійність їх можуть суттєво відрізнятися. Мобільні застосунки зручні, але не завжди точні. Веб-платформи можуть допомогти швидко оновити рецепт, але не замінять повноцінного обстеження. Апаратно-програмні комплекси, що використовуються в клініках, є найбільш точними, але недоступні для домашнього використання.

Методики визначення характеристик зору є стандартизованими, і наш застосунок повинен їх враховувати. Принцип Вебера-Фехнера може допомогти нам розробити більш точні тести на гостроту зору.

Важливо також дотримуватися клінічних протоколів, стандартів обміну медичними даними (у перспективі) та законодавства про захист персональних даних. Для використання в Україні необхідно відповідати вимогам МОЗ.

РОЗДІЛ 2

ЗАСОБИ РЕАЛІЗАЦІЇ ТА ЗАПРОПОНОВАНІ РІШЕННЯ

Розділ пояснює, чому саме Android Studio разом з Java, SQLite зі шифруванням і Text-to-Speech були обрані для створення застосунку. Окремо описуються алгоритми для чотирьох тестів, логіка UX/UI та структура JSON-даних, що гарантує коректне зберігання результатів.

2.1 Вибір мов, середовищ і бібліотек

Для мобільного додатку на Android я взяв Android Studio. Це головне середовище для розробки (IDE). Чому? Бо Android Studio зробив Google саме для Android. Там є все, щоб створювати, виправляти помилки та тестувати додатки. Якщо брати офіційне середовище, то, скоріш за все, буде краща сумісність і доступ до нових функцій Android SDK теж буде. Це краще, ніж інші IDE. Це може зекономити час і проблем з інтеграцією буде менше. Google підтримує Android Studio, тому нові фішки Android, швидше за все, з'являться спочатку там. Така пряма інтеграція робить розробку плавнішою. А ще, для студента-розробника дуже важлива документація і підтримка спільноти. Навколо Android Studio цього багато [75].

Головною мовою програмування я обрав Java. Kotlin стає популярнішим, але Java все ще дуже поширена. Вона зріла мова для Android-розробки [67]. Вибір між Java та Kotlin – це компроміс. Треба було обирати між зрілістю та новими функціями. Kotlin коротший у написанні і безпечніший щодо нульових покажчиків, але Java має багату історію. Це означає більше ресурсів і краще розуміння її особливостей в Android. Java давно використовують для Android. Тому є багато онлайн-підручників, курсів, документації [66]. Студенту з дипломною роботою така підтримка дуже допомагає. З нею легше вирішувати технічні проблеми. До того ж,

багато Android-бібліотек з відкритим кодом написані на Java і багато існуючих додатків теж. Це полегшує їх інтеграцію та вивчення [48].

Java має переваги. Наприклад, об'єктно-орієнтований підхід. Він допомагає краще організувати код, і підтримувати його. Об'єктно-орієнтованість Java дозволяє моделювати складні процеси [67]. Наприклад, у медицині це діагностичні процедури. Моделювати їх можна точніше та ефективніше [26]. Такий підхід з чіткою структурою корисний для медичного додатку. Там ясність і організація дуже важливі.

У медицині діагностика і лікування – це складні процеси. Java може показати їх як об'єкти та класи. Це робить код зрозумілішим. Тому така кодова база стає більш інтуїтивною та підтримуваною, що є важливим не тільки для старту розробки, але й для всіх майбутніх оновлень. До того ж, Java добре працює з багатьма потоками, і це корисно для фонових задач, наприклад, для обробки тестів чи озвучення тексту [67].

Безпека для медичних додатків дуже важлива. Java має вбудоване шифрування. Також є безпечні протоколи зв'язку [32]. Kotlin теж має свої плюси, наприклад, кращий захист від `NullPointerException`, але на старті розробки Java здавалася надійнішою для медичного софту [33]. Хоча безпека від нульових показчиків у Kotlin – це добре, проте зрілість Java, її розширені функції безпеки, особливо вбудована підтримка шифрування та безпечного зв'язку, виявилися важливішими для цього медичного додатку на той момент [67]. Медичні програми працюють з таємними даними пацієнтів. Тому безпека – це головне. Java давно має хорошу репутацію в безпеці. Її спеціальні функції для безпечного зв'язку та захисту даних додали впевненості проекту. А наявність перевірених фреймворків безпеки, як `Spring Security` в Java, ще більше підсилила це рішення [26].

Щоб були голосові підказки в тестах, я взяв `Text-to-Speech (TTS) API`. Він є в `Android SDK`. Так користувачі, навіть з проблемами зору, зможуть зручно проходити тести. Бо назви літер будуть озвучуватися. Інтеграція `TTS API` напряду використовує можливості `Android`, що гарантує сумісність і,

мабуть, зменшує потребу в інших бібліотеках, роблячи проект простішим у залежностях. Android SDK має багато вбудованих API, і TTS серед них. Коли використовуєш такі рідні функції, часто отримуєш кращу продуктивність. І кращу інтеграцію з системою, ніж зі сторонніми бібліотеками [22].

Результати тестів та дані користувачів я зберігаю локально. Для цього обрав базу даних SQLite. SQLite – легка база даних. Вона вбудована, реляційна і добре підходить для мобільних додатків, бо не треба окремий сервер, а дані зберігаються просто у файлі на телефоні. Простота SQLite робить її супер вибором для локального зберігання даних, особливо коли це структуровані дані типу результатів тестів чи акаунтів. Додаток переважно працює офлайн. Дані зберігаються локально. Тому SQLite – це просте й ефективне рішення. Маленький розмір і відсутність сервера спрощують архітектуру та розгортання.

Я думав про Kotlin. Ця мова має коротший синтаксис. І сучасні функції. Це могло б прискорити розробку і зменшити кількість шаблонного коду [57]. Але навчальних матеріалів не вистачало, щоб швидко вирішувати проблеми. Kotlin сумісний з Java, але повний перехід на нього вимагав би більше часу на навчання та інтеграцію. Тому я зважив усе і вирішив зосередитись на Java, хоча Kotlin можна буде використати пізніше. Рішення взяти переважно Java, хоч і визнаючи плюси Kotlin, показує обережний підхід: стабільність, доступні ресурси та перевірена екосистема виявилися важливішими за переваги нової мови. Google рекомендує Kotlin для Android, але для мене важливіше швидкість навчання та наявність підтримки, тому я обрав щось перевірене, як Java. Добре, що Java та Kotlin сумісні, тому додати Kotlin пізніше – цілком реально [33].

Щоб обґрунтувати фінальний вибір на користь Java, нижче (табл. 2.1) наведено стислий порівняльний огляд обох мов за основними характеристиками, що безпосередньо впливають на розробку, тестування та супровід застосунку.

Таблиця 2.1

Порівняльна таблиця Java та Kotlin

Критерій	Java	Kotlin
Зрілість та підтримка	Велика екосистема, багаторічна підтримка Android SDK, безліч навчальних матеріалів і прикладів	Новіша мова, рекомендована Google, але з меншою кількістю курсів і прикладів на момент вибору
Синтаксис та лаконічність	Більш деталізований, вимагає більше шаблонного коду (boilerplate)	Лаконічніший синтаксис, вбудована підтримка null-безпеки, менше шаблонного коду
Null-безпека	Відсутня на рівні мови — потрібні перевірки та рутини перевірки null	Вбудована система null-безпеки (nullable / non-nullable типи)
Продуктивність	Нативний байткод на JVM, оптимізований Google, стабільна продуктивність	Транслюється в JVM-байткод, продуктивність близька до Java, але іноді трохи повільніше при стартах
Навчальна крива	Зрозуміла багатьом розробникам і має безліч ресурсів, що пришвидшує навчання	Потрібен час на освоєння нових конструкцій (корутин, розширень)
Сумісність бібліотек	100 % сумісність з існуючими Java-бібліотеками та Android API	Повна сумісність з Java, але іноді потребує обгортки для старих API
Підтримка Android Studio	Вбудована підтримка, всі нові фічі Android SDK першими з'являються у Java-проектах	Підтримується Google, але нові можливості іноді доступні пізніше

2.2 Моделі даних, формати, шкали та кольорові стандарти

2.2.1 Шкала logMAR

Щоб тести були медично правильними, я взяв загальноприйняті шкали. Шкалу logMAR визнають еталоном. Вона потрібна для оцінки гостроти зору. Особливо в науці. Ця шкала має переваги над таблицею Снеллена, а її основні принципи включають послідовний та геометричний ріст розміру

літер між рядками, що робить рівень складності однаковим по всій таблиці [4, 36]. Кожен рядок має літери схожої розбірливості. Відстань між літерами стандартизована. І між рядками теж [6]. Це зменшує ефекти скупчення, а однією з найбільших переваг є оцінювання "літера за літерою", що дає точніше вимірювання гостроти зору [4, 36]. Цифрова версія logMAR дуже гнучка. Можна міняти відстань тестування. Результати оцінюються автоматично [6]. Додаток адаптує розмір символів під введену відстань, що відповідає принципам logMAR і забезпечує точне тестування на будь-якій відстані [4, 36].

Є кілька методів оцінки за logMAR. Наприклад, оцінка за літерами. Кожна правильна літера впливає на результат. Є ще оцінка за рядками: дивляться на найнижчий прочитаний рядок і коригують за додатковими літерами. Мій додаток, оцінює за літерами, рахуючи правильні відповіді на кожному рівні розміру для визначення гостроти зору. Завдяки logMAR, додаток дає науково обґрунтовану оцінку. Вона стандартизована. Це краще, ніж стара таблиця Снеллена. Цифрова реалізація дуже гнучка, особливо можливість підлаштовуватися під різні відстані тестування, що робить тест практичнішим і точнішим вдома чи в польових умовах [4, 36, 37].

2.2.2 Таблиці Ішихари

Таблиці Ішихари – це перевірений інструмент. Вони широко визнані. Допомагають виявляти дальтонізм. Особливо добре вони знаходять проблеми з червоно-зеленим сприйняттям кольорів, а їхній основний принцип – псевдоізохроматизм: люди з нормальним зором бачать одні цифри чи візерунки з кольорових крапок, а люди з порушеннями – інші [29].

Щоб цифровий тест Ішихари працював добре, кольори на екрані мають бути точними. Додаток мусить правильно показувати відтінки. Насиченість крапок теж має бути правильною. Усе як в оригінальних таблицях. Також користувачам потрібні чіткі інструкції щодо налаштувань екрану, наприклад, як налаштувати яскравість та вимкнути фільтри, що міняють колір, аби тест був достовірним [28, 27].

Стандартний тест Ішихари – це набір таблиць. Оцінка залежить від того, скільки таблиць користувач розпізнав правильно. Тобто, скільки цифр чи візерунків він побачив. Існують різні типи таблиць: демонстраційні, трансформаційні, зникаючі, з прихованими цифрами, діагностичні та трасувальні (рис. 2.1 – рис. 2.6), і кожна з них допомагає не просто виявити проблему, а й визначити тип і ступінь порушення кольорового зору [28, 27].

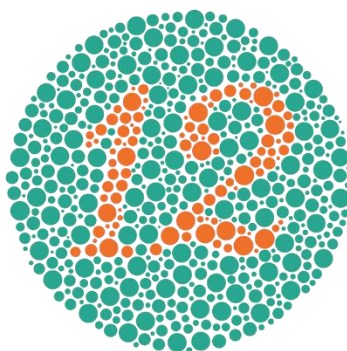


Рисунок 2.1 - Демонстраційна пластина Ішихари [28]

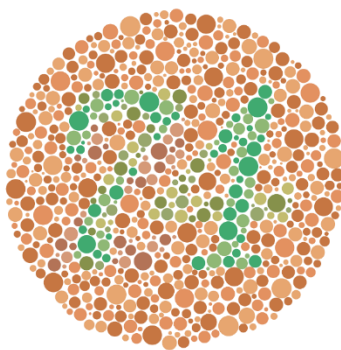


Рисунок 2.2 - Трансформаційна пластина Ішихари [28]

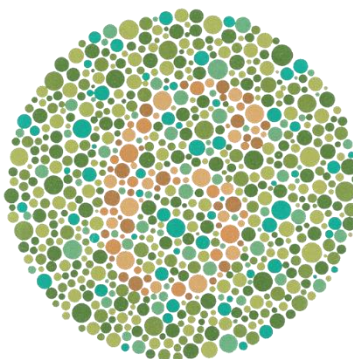


Рисунок 2.3 - Зникаюча пластина Ішихари [28]

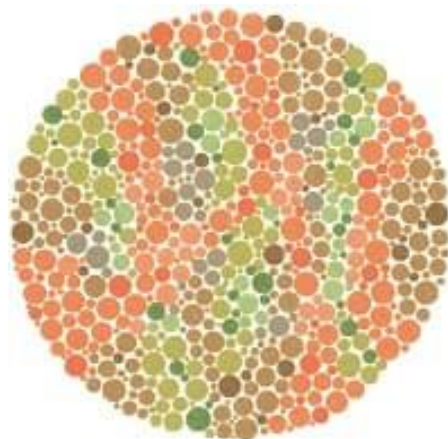


Рисунок 2.4 - Прихована пластина Ішихари [28]



Рисунок 2.5 - Діагностична пластина Ішихари [28]

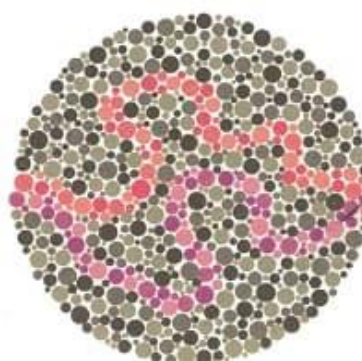
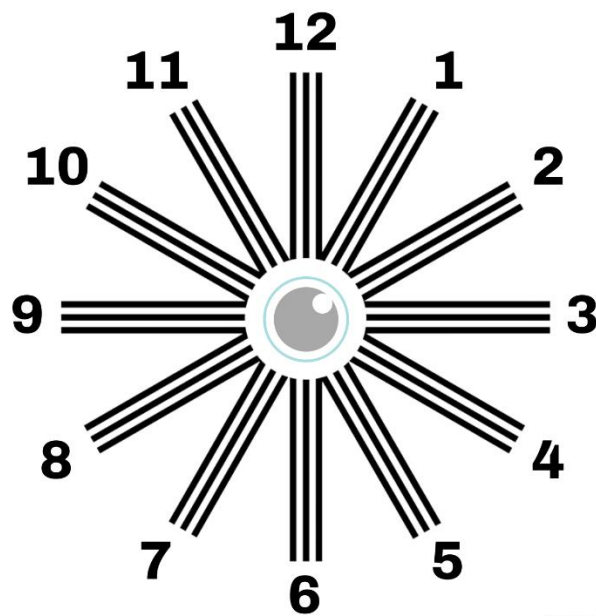


Рисунок 2.6 Слідова пластина Ішихари [28]

Цифровий тест Ішихари – це зручно. Він допомагає перевіритись на дальтонізм. Але його ефективність залежить від точності кольорів на екрані. Тому дизайн додатку має ставити в пріоритет точне відтворення кольорів і давати користувачам інструкції, як зменшити спотворення від дисплея, що можуть вплинути на результати [29, 27].

2.2.3 Радіальні решітки

Радіальні решітки схожі на циферблат, або на віяло з лініями (рис. 2.7). Їх часто використовують в офтальмології. Так перевіряють на астигматизм, а головний принцип тестів такий: люди з астигматизмом (коли рогівка або кришталик нерівні) бачать одні лінії чіткіше чи темніше за інші, тоді як решта ліній можуть здаватися їм розмитими або хвилястими [7].



www.ophtalmology24.com

Рисунок 2.7 - Сонцеподібна діаграма для діагностики астигматизму [18]

У моєму додатку користувач бачить радіальне зображення. Йому треба сказати, чи всі лінії однаково чіткі. Чи деякі розмиті або спотворені. Цифровий тест з решіткою оцінює суб'єктивне сприйняття чіткості ліній у різних напрямках [7]. Якщо користувач каже, що лінії в одному напрямку розмиті, а інші чіткі, це може вказувати на астигматизм. Хоча такі онлайн-тести корисні для скринінгу, важливо пам'ятати: остаточний діагноз астигматизму може поставити лише фахівець після повного обстеження очей [18].

Тест з радіальною решіткою – це зручно. Користувач може сам перевіритись на астигматизм. Але додаток має чітко казати, що результати

тесту не замінять огляд у лікаря, якщо щось не так.

2.2.4 JSON формат даних

JSON – це популярний формат даних. Він легкий. Його зручно читати. JSON використовують для структурування та обміну інформацією між програмами, а його простота та гнучкість роблять його дуже зручним для мобільних додатків. Я використовую JSON-схеми для зберігання результатів тестів зору. Це дає кілька переваг. По-перше, дані організовані однаково. Відповідно до заданої структури. Така однорідність важлива для цілісності даних. Вона полегшує аналіз, звітність та інтеграцію з іншими медичними системами. Добре продумана JSON-схема для результатів тестів може містити багато важливих даних: ID користувача, тип тесту, час тесту, параметри тесту (наприклад, відстань для тесту гостроти зору), а також сирі та оброблені результати для кожного тесту [34].

Наприклад, для тесту гостроти зору схема буде мати поля для балів logMAR для кожного ока. Для тесту на дальтонізм – кількість правильно визначених таблиць Ішихари. Використання JSON-схем для зберігання результатів створює надійну основу для управління даними, і такий структурований підхід спрощує доступ до інформації, її інтерпретацію, а також дає більше можливостей для розширення та сумісності з іншими медичними програмами.

2.3 Принципи UX / UI для офтальмологічних тестів

Голосові підказки дуже важливі. Вони допомагають керувати користувачами. Особливо під час тестів зору [2]. Це особливо корисно для людей зі слабким зором, яким важко читати інструкції на екрані, а впровадження голосових команд для озвучення літер у тестах на гостроту зору та аметропію є чудовим прикладом того, як голосові підказки покращують доступність. Щоб підказки були ефективними, важливі чіткість і точність. Синтезована мова має бути зрозумілою, вимова літер – чіткою, а

можливість контролювати темп чи швидкість підказок ще більше покращить зручність, дозволяючи людям налаштовувати все під себе [22]. Розумне використання голосових підказок робить додаток значно доступнішим. Це дозволяє користувачам. Навіть з порушеннями зору. Самостійно і легше проходити тести.

2.4 Вибір і обґрунтування алгоритмів обробки результатів

2.4.1 Алгоритм для розрахунку гостроти зору

Алгоритм для розрахунку (рис. 2.8) гостроти зору має спиратися на принципи logMAR [36]. Перший важливий крок – точно врахувати відстань. Ту, що вказав користувач [6]. Розмір літер на екрані має динамічно змінюватися залежно від цієї відстані, щоб зберігати правильний кутовий розмір оптичів, адже це головне в тестуванні за шкалою logMAR [36].

Алгоритм має уважно стежити за відповідями. Особливо за кількістю правильних літер. На кожному рівні, де літери меншають. На основі цих даних розраховується бал logMAR для кожного ока [36]. Те, як додаток скидає лічильник спроб після правильної відповіді і зменшує літери після кількох правильних, відповідає типовій адаптивній процедурі тестування logMAR. Тест іде, доки користувач не дійде до найменших літер. Або поки не зробить певну кількість помилок на одному рівні.

Головна функція алгоритму така. Він бере взаємодію користувача з завданням. Це розпізнавання літер за logMAR, і з точним урахуванням відстані перетворює це на стандартизований і клінічно значущий бал logMAR, який показує гостроту зору кожного ока.

На рисунку нижче відображено покроковий алгоритм, що починається з короткого інструктажу користувача, приймання введеної дистанції до екрана та автоматичного масштабування символів згідно з офтальмологічними стандартами. Схема наочно фіксує логіку, що забезпечує коректне проходження тесту й одночасно мінімізує когнітивне навантаження на користувача.

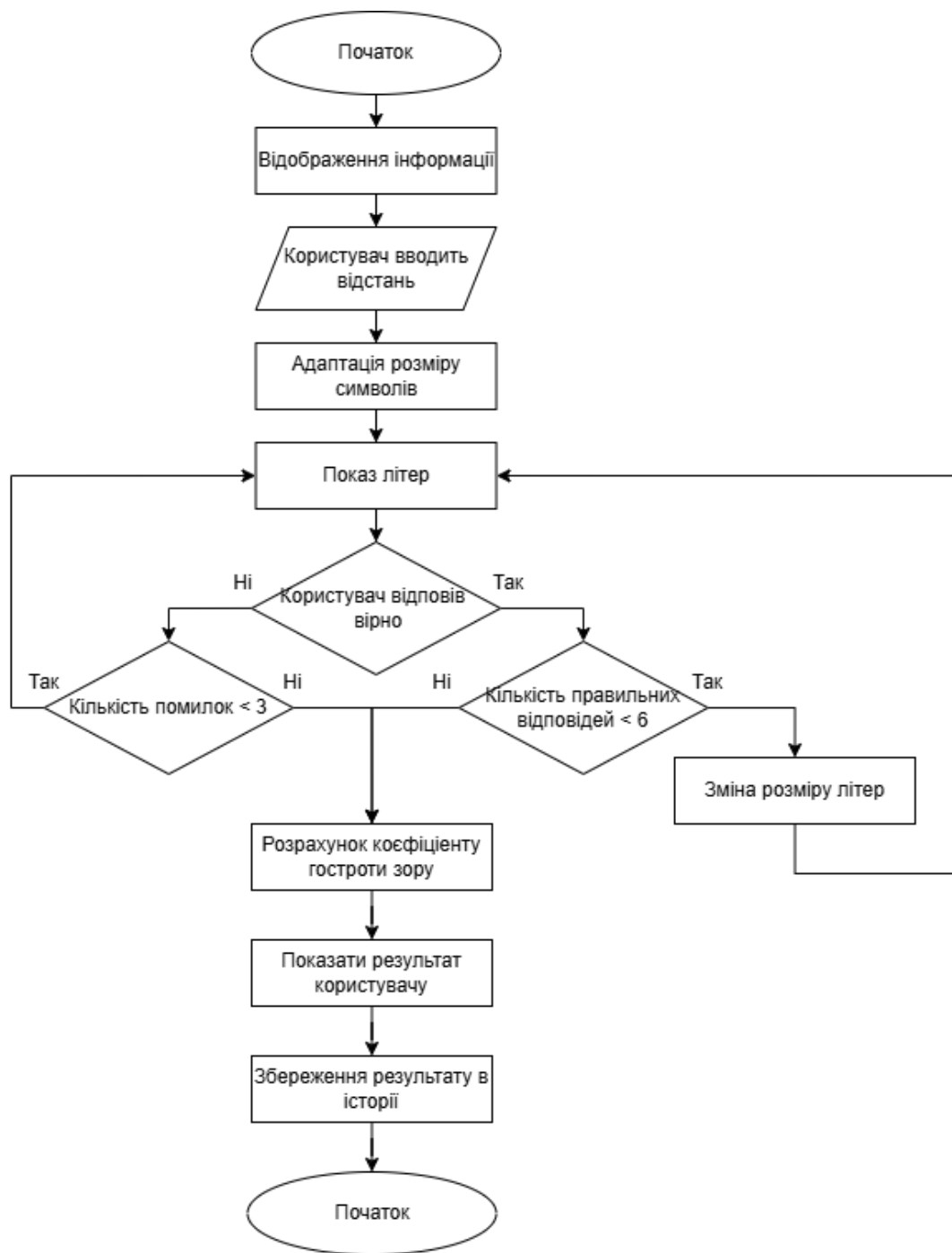


Рисунок 2.8 – Алгоритм проходження тесту на гостроту зору

2.4.1 Алгоритм визначення аметропії

Алгоритм для аметропії схожий на тест гостроти зору. Але є важлива відмінність. Тестування йде на червоному і на зеленому фоні (рис. 2.9) [10]. Головне в цьому алгоритмі – порівняти бали гостроти зору на червоному фоні з балами на зеленому, і це порівняння робиться для кожного ока окремо.



Рисунок 2.9 – Дуохромний тест на аметропію [74]

Цей підхід бере за основу дуохромний тест. Він використовує хроматичну аберацію ока. Це коли світлові хвилі різної довжини фокусуються в різних точках ока. Зазвичай, люди з короткозорістю бачать чіткіше на червоному фоні, а люди з далекозорістю – на зеленому. Алгоритм аналізує ці маленькі різниці в гостроті зору на двох фонах, щоб зрозуміти, чи є помилка рефракції (короткозорість чи далекозорість) і яка саме для кожного ока [10, 74].

Головне завдання алгоритму для аметропії – аналіз різниці. Різниці в балах гостроти зору. На червоному і зеленому фонах. Використовуючи принципи дуохромного тесту, алгоритм може визначити, чи є у користувача ймовірно короткозорість або далекозорість в кожному оці.

2.4.2 Алгоритм визначення дальтонізм

Алгоритм для тесту на дальтонізм працює так. Користувач взаємодіє з таблицями Ішихари. Він має назвати число з кольорових крапок [29]. Головний крок алгоритму – порівняти відповідь користувача для кожної таблиці Ішихари з правильною відповіддю для неї [27]. Потім алгоритм рахує, скільки разів користувач відповів правильно. Головна функція

алгоритму – кількісно оцінити, як користувач розпізнає числа на таблицях Ішихари. Порівнюючи відповіді з правильними, алгоритм визначає, чи нормальний у користувача колірний зір, чи є дефіцит. Але алгоритм не може класифікувати тип та ступінь порушення за критеріями тесту Ішихари.

2.4.2 Алгоритм визначення дальтонізм

Алгоритм для астигматизму спирається на сприйняття користувача. Як він бачить чіткість. При перегляді радіальних зображень. Головне завдання користувача – сказати, чи всі лінії на картинці однаково чіткі та різкі, чи деякі здаються розмитими, хвилястими або спотвореними [7].

Оскільки це самостійний скринінговий тест, результат, буде простим. Так чи ні. Тобто, чи є підозра на астигматизм. А не точне числове значення помилки. Головна функція алгоритму – аналізувати суб'єктивні відповіді користувача про чіткість ліній на картинках. Знаходячи постійні закономірності розмитості в певних напрямках, алгоритм може попередити про можливий астигматизм і порадити звернутися до лікаря для подальшої перевірки.

Висновок до розділу 2

Я обрав Android Studio. Разом із сучасною мовою Java. Це безпечна мова. Таке поєднання дає надійне та ефективне середовище розробки. Інтеграція Text-to-Speech API від Android дуже важлива для доступності через голосові підказки, а от рішення використовувати SQLite для локального зберігання даних потребує великої уваги до безпеки та шифрування цих даних.

Шкала logMAR для тестів гостроти зору – це добре. Вона дає стандартне і точне вимірювання. Цифрові таблиці Ішихари теж важливі і радіальні решітки. Вони дозволяють перевіряти на дальтонізм та астигматизм. Алгоритми для обробки результатів я продумав ретельно. Для кожного тесту. Щоб вони відповідали офтальмологічним методам і давали

корисну інформацію про зір.

Таким чином, вибір засобів реалізації та запропонованих рішень для розробки програмного застосунку для перевірки гостроти зору в польових і домашніх умовах був здійснений з урахуванням вимог до функціональності, надійності, доступності та безпеки медичних застосунків.

Обрана архітектура легко масштабується: при потребі можна додати модуль хмарної синхронізації на Kotlin Multiplatform без переробки бізнес-логіки або інтегрувати машинне навчання для персоналізованих рекомендацій. Усе це робить запропоноване рішення не лише функціональним сьогодні, а й готовим до еволюції разом із клінічними стандартами та зростанням користувацької аудиторії.

РОЗДІЛ 3

ОБГРУНТУВАННЯ ВИБОРУ ПРОГРАМНОГО СЕРЕДОВИЩА ТА АРХІТЕКТУРИ

У розділі порівнюються Flutter, React Native та нативний Android, демонструючи переваги Java-проєкту з патерном MVVM. Розкрито модульну будову, механізми шифрування через Android Keystore і стратегію підтримки різних версій ОС.

3.1 Порівняння альтернативних платформ

На початку розробки я думав про те, яку платформу обрати. Android Studio та Java були першим варіантом і найбільш звичним. Але мій обов'язок як студента – розглянути й інші технології.

3.1.1 Flutter

Flutter від Google одразу мене зацікавив. Головний плюс – кросплатформеність. Можна написати код один раз і він працюватиме і на Android, і на iOS. Це дуже привабливо, бо користувачів може бути багато. Flutter також пропонує швидку розробку через "гаряче перезавантаження"¹, що дозволяє бачити зміни миттєво без повної перекомпіляції, і це точно прискорило б створення прототипів та покращення інтерфейсу [25].

Але попри ці плюси, з'явилися сумніви. Щодо продуктивності і інтеграції з функціями Android. Для медичного додатку точність і швидкість дуже важливі, тому будь-які обмеження кросплатформних рішень викликали тривогу. Хоча Flutter досить швидкий, кажуть, що іноді, особливо при складних завданнях, він може бути гіршим за нативні додатки на Android [17].

3.1.2 React Native

Ще я дивився на React Native. Ця платформа використовує JavaScript. Я

вже трохи працював з цією мовою [49]. Велика спільнота розробників та багато готових бібліотек і компонентів могли б сильно полегшити й прискорити розробку, а можливість повторно використовувати багато коду на різних платформах теж здавалася гарним аргументом [69, 50].

Але хоч я і знав JavaScript, і екосистема React Native розвинена, я задумався про можливі проблеми з продуктивністю. Особливо для медичного додатку. React Native стає кращим, але є дані, що іноді він працює гірше за нативні додатки, наприклад, повільніше завантажується або має менш плавні анімації [17]. Для медичного додатку інтерфейс має бути дуже швидким, тому навіть маленькі затримки неприпустимі.

До того ж, інтеграція з деякими Android API могла б бути складнішою. Наприклад, Text-to-Speech з тонкими налаштуваннями голосу. У React Native це важче, ніж у нативній розробці [20]. Також треба врахувати, що розмір додатку на React Native може бути більшим, ніж нативного [23].

3.1.3 Android Studio з Java

Врешті-решт, я все проаналізував. Всі "за" і "проти". І повернувся до першого вибору. Це нативна розробка на Android Studio з Java. Це рішення мало кілька причин. По-перше, нативна розробка дає найкращу продуктивність, що дуже важливо для медичного додатку з інтенсивною обробкою даних та складною візуалізацією [42]. По-друге, нативний Android дає повний доступ до всіх можливостей системи та пристрою: камер, сенсорів, потужного Text-to-Speech API.

Це дуже важливо для тестів зору. Наприклад, для адаптації розміру символів. Або для карт Ішихари та радіальних зображень. По-третє, хоча Kotlin – сучасніша мова для Android, Java все ще надійний інструмент з купою бібліотек та ресурсів, що теж вплинуло на мій вибір, особливо на початку навчання та під час написання диплому [11].

Вибір Java для нативної розробки може здатися не таким модним. Порівняно з Flutter чи Kotlin. Але це обґрунтоване рішення. Воно продиктоване вимогами медичного додатку та моїми знаннями.

Продуктивність і надійність нативної розробки тут найголовніше.

Кросплатформні фреймворки привабливі. Бо вони універсальні. Але вони додають рівень абстракції [42]. Це може вплинути на продуктивність і розмір додатку. Для медичного додатку важлива швидка робота без збоїв, особливо під час тестів зору, тому нативна розробка на Java виглядає надійнішим варіантом, який легше контролювати. До того ж, я вже працював з Java в Android Studio. Це дозволить мені зосередитись на функціоналі [35]. А не витратити час на вивчення нової технології, бо на диплом часу мало.

Таблиця нижче (табл. 2.1) стисло показує основні різних платформ відмінності й допомагає обрати найдоцільніший варіант для проєкту.

Таблиця 3.1

Порівняльна таблиця альтернативних платформ

Платформа	Частка спільного коду	Продуктивність на Android	Коли варто обирати
Native Android (Java / Kotlin + Jetpack Compose)	0 % між Android та iOS (усе пишеться окремо) (Android Developers)	«Еталонна» — використовує рідні компоненти й оптимізації Compose; останні допрацювання в Android Studio Iguana скоротили час збирання та рендер-лаги	Потрібні максимальна швидкодія, глибока інтеграція з Google Play Services та Wear OS, а iOS версію не планують найближчим часом.
Flutter 4 (Dart + Impeller)	95 % коду повторно на Android / iOS / Web / Desktop	Новий рушій Impeller прибрав «джанк» і оптимізував GPU, підвищивши FPS у складних сценах	Потрібно швидко вийти на кілька платформ з єдиним UI й без занурення в нативні SDK; команда не проти писати на Dart.
React Native (New Architecture, TypeScript / JS + Fabric)	85 – 90 % мобільного коду спільне; Web-версія додається через React / Next	Перехід на JSI + Fabric помітно скоротив час старту й зменшив CPU-навантаження на -15 %	Команда вже сильна у JavaScript, потрібна Web-та Mobile-єдність, а ресурс на оптимізацію нативних нюансів обмежений.
Kotlin Multiplatform Mobile (KMM + Compose Multiplatform)	60 – 80 % бізнес-логіки спільне; UI під кожен ОС або Compose MP	Рідна швидкодія: код компілюється в LLVM Native; Compose for iOS став Stable у травні 2025 р.	У команді вже використовують Kotlin; хочуть додати iOS без JS або Dart-стеку. Велика компанія (Netflix, Baidu тощо) вже має позитивний досвід

3.2 Архітектурний стиль (MVVM) та схема модулів

3.2.1 Архітектурний стиль

Щоб код мого додатку був добре організований, я обрав стиль MVVM. Це Model-View-ViewModel. На мій погляд, цей патерн дуже ефективний. Особливо для Android-розробки, бо він чітко ділить відповідальність між різними частинами програми [41].

Головна ідея MVVM – поділити додаток на три частини. Вони пов'язані між собою. Перша – Model. Цей рівень відповідає за дані, і за бізнес-логіку. У моєму випадку, це класи, що обробляють результати тестів зору, керують збереженням даних і відповідають за локальну аутентифікацію [41].

Друга частина – View. Вона показує інформацію користувачу і обробляє його дії. В Android-додатку це Activity та Fragment. А також XML-макети, які описують, як виглядає інтерфейс [41].

Третя частина – ViewModel. Вона як посередник між View та Model. ViewModel дає дані для View, а також обробляє команди від View і передає їх до Model. Важливо, що ViewModel не залежить від конкретної View, що робить тестування легшим і дозволяє повторно використовувати код [41].

3.2.2 Модулі

Щоб краще структурувати додаток, я поділив його на модулі (рис. 3.1). Вони логічно відокремлені. Перший – модуль аутентифікації. Цей модуль відповідає за реєстрацію та вхід користувачів, а також за управління їхніми локальними акаунтами.

Далі – модуль тестування зору. Тут класи для логіки тестів. Всіх тестів: гострота зору, аметропія, також дальтонізм і астигматизм. Ще він має механізми для адаптації розміру символів під відстань, яку ввів користувач, і для відтворення голосових команд під час тестів.

Є модуль збереження даних. Його головна функція – локально зберігати результати проведених тестів. Для цього я використовую SQLite, а ще там є механізми шифрування даних для більшої безпеки.

Ще є модуль історії результатів. Він показує користувачу історію всіх

його попередніх тестів. Це дозволяє стежити за динамікою змін зору.

І останній – модуль інтерфейсу користувача. Сюди входять загальні компоненти. Це кнопки, текстові поля. А також стилі та теми, які використовуються по всьому додатку для однакового вигляду.

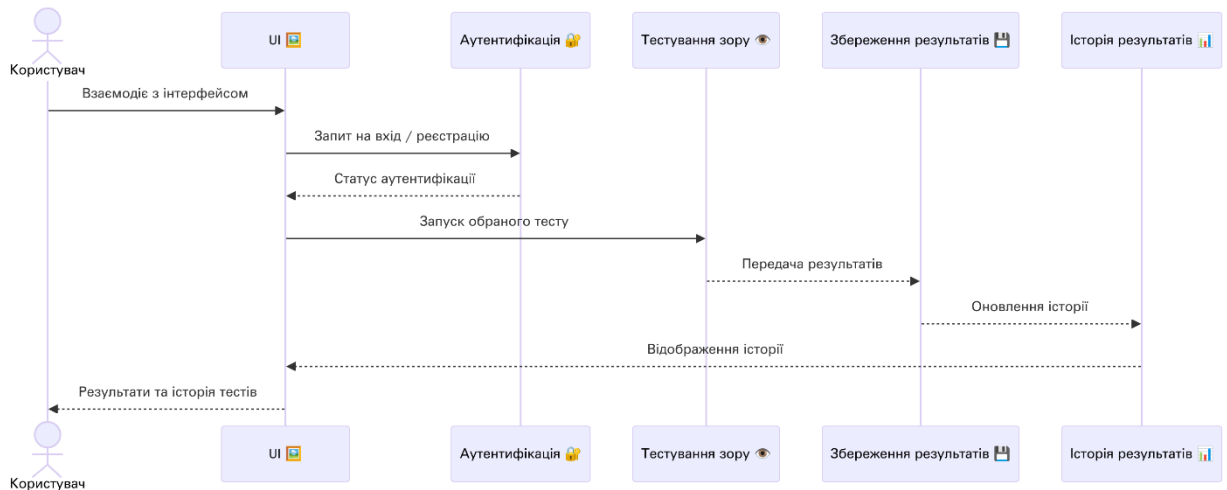


Рисунок 3.1 - Діаграма взаємодії користувача з модулями

3.2.3 MVVM

MVVM не просто допомагає краще організувати код. Він також полегшує підтримку і можливе масштабування додатка. Чіткий поділ логіки ViewModel від відображення даних у View робить код гнучкішим до змін в інтерфейсі без потреби сильно міняти бізнес-логіку [41].

Архітектура MVVM (рис. 3.2) дозволяє мені чітко розділити код для відображення і взаємодії і код для бізнес-логіки та даних. ViewModel з'єднує ці рівні, готуючи дані для показу та обробляючи дії користувача. Такий підхід значно спрощує тестування окремих частин і робить код зрозумілішим та зручнішим для підтримки [41, 5].

Наприклад, я можу тестувати логіку алгоритмів тестів зору. І це не залежить від графічного інтерфейсу. Це великий плюс при розробці медичного додатку, де надійність і точність результатів дуже важливі [41].

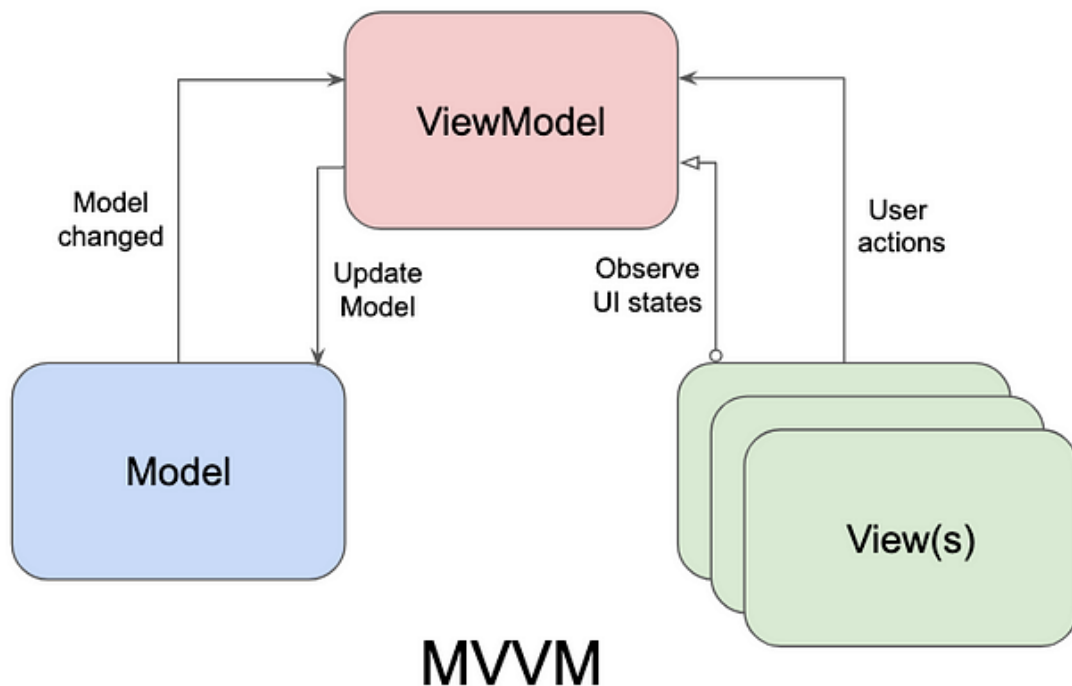


Рисунок 3.2 - Model-View-ViewModel [5]

3.3 Механізми локальної аутентифікації та шифрування даних

Щоб персональні дані користувачів були в безпеці, я зробив локальну аутентифікацію. Це особливо стосується результатів тестів зору. У додатку є такий механізм, що коли користувач перший раз запускає додаток, він реєструється. Створює собі унікальний акаунт, вказує логін і пароль. Потім, щоб побачити свої результати, треба буде входити з цими даними, а всі результати тестів зберігаються локально і прив'язані до конкретного акаунту.

Медичні дані дуже чутливі. Результати тестів зору – теж. Тому всі дані в SQLite шифруються обов'язково. Для цього я планую взяти бібліотеку SQLCipher [40]. Вона потужна і відома, є стандартом для безпеки баз SQLite на Android і дає розробникам надійний інструмент для шифрування всієї бази. SQLCipher використовує стійкий алгоритм AES-256, що гарантує високий захист від чужого доступу [40].

Ключ для шифрування буде створюватися з пароля. Того, що користувач вводить при реєстрації. Дуже важливо безпечно зберігати цей ключ, і правильно його використовувати. Для цього я хочу використати Android Keystore [9]. Це спеціальне сховище для криптографічних ключів на пристрої.

Android Keystore дає апаратне шифрування ключів. На багатьох сучасних телефонах. Це сильно ускладнює їх крадіжку. Навіть якщо хтось отримає фізичний доступ до пристрою [9].

SQLCipher робить шифрування і дешифрування прозорим. Тобто розробнику не треба нічого викликати при записі чи читанні даних. SQLCipher сам перехоплює запити до бази і робить потрібні криптографічні операції на низькому рівні, що сильно спрощує розробку і зменшує ризик помилок з криптографією [40].

Безпека даних користувачів – це дуже важливо. Особливо для медичного додатку. Локальне зберігання даних, надійне шифрування через SQLCipher і безпечне управління ключами через Android Keystore. Все це разом значно зменшує ризики витоку конфіденційної інформації, і такий підхід показує серйозне ставлення до захисту приватності користувачів та їхніх медичних даних [40].

3.4 Вибір стратегії масштабування й підтримки версій Android

Мій додаток зберігає всі дані локально. Значне збільшення кількості користувачів. Це не головна мета на цьому етапі. Але я розумію, що потім можуть знадобитися нові функції. Наприклад, синхронізація даних у хмарі між різними пристроями. Або більше тестів зору. Мій стиль архітектури MVVM досить гнучкий і добре підходить для майбутніх змін.

Код поділений на окремі модулі. Вони незалежні. Це полегшить додавання нових функцій. Наприклад, синхронізацію з хмарою, якщо знадобиться. А ще модульна структура дозволить легко додавати нові тести

зору та покращувати старі, не переробляючи багато вже написаного коду.

При розробці Android-додатку важливо вибрати версії системи які будуть підтримуватися. Щоб додаток був доступний багатьом. Тому я планую поставити мінімальну версію SDK – Android 5.0 (API 21). Статистика каже, що ця версія охоплює багато активних пристроїв, і це дасть баланс між підтримкою старих, але робочих телефонів, та можливістю використовувати нові API з пізніших версій системи [8].

Цільова версія SDK буде остання стабільна версія Android. Це дозволить додатку використовувати нові функції і оптимізації системи. А також забезпечить кращу сумісність з новими телефонами на ринку.

Тестування додатку буде всебічним. На емуляторах і на реальних телефонах. З різними версіями Android, від мінімальної до цільової. Це потрібно, щоб знайти і виправити можливі проблеми сумісності на різних версіях системи.

Підтримка багатьох версій Android важлива. Щоб було більше користувачів. Але треба пам'ятати про складнощі. Це підтримка старих API і забезпечення роботи на слабших телефонах.

Мінімальна версія Android SDK 5.0 – це компроміс. Він дозволяє охопити багато телефонів і використовувати сучасні інструменти. Орієнтація на останню стабільну версію Android SDK гарантує, що додаток буде оптимізований для нових пристроїв і зможе використовувати всі плюси нових версій системи. Ретельне тестування на різних версіях Android – ключ до стабільної та надійної роботи на всіх підтримуваних телефонах.

Висновок до розділу 3

Обґрунтування вибору програмного середовища та архітектури для розробленого застосунку базується на комплексному аналізі доступних технологій та вимог до медичного програмного забезпечення.

Я обрав нативну розробку на Android. З використанням Java. Бо треба

висока продуктивність. І повний доступ до системних API для функцій тестування зору. Архітектурний стиль MVVM я взяв, щоб код був добре організований, а також щоб його було легше підтримувати та масштабувати потім.

Локальна аутентифікація та шифрування даних – це дуже важливо. Для захисту персональної інформації. Стратегія підтримки версій Android теж продумана. Вона має на меті охопити широку аудиторію і водночас дати можливість використовувати сучасні функції системи.

РОЗДІЛ 4

ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ

Цей розділ присвячений практичній реалізації програмного застосунку на платформі Android. В ньому детально описано створення інтерфейсу користувача, реалізацію окремих тестів на зір, функції збереження результатів, адаптацію розміру символів згідно з офтальмологічними стандартами. Також в цьому розділі було розраховано економічний ефект програмного застосунку.

4.1 Реалізація інтерфейсу користувача

Інтерфейс користувача розробленого Android-застосунку спроектовано з метою забезпечення максимальної зручності та інтуїтивності використання для користувачів будь-якого рівня підготовки. Особлива увага приділялася чіткості навігації та доступності основних функцій.

При першому запуску застосунку користувач потрапляє на екран реєстрації та входу (рис. 4.1). Цей екран містить стандартні поля для введення логіна та пароля, та має кнопку "Зареєструватись" для створення нового облікового запису, та «Вхід» якщо обліковий запис вже існує.



Рисунок 4.1 - Екран реєстрації та входу

Після успішної реєстрації або введення існуючих облікових даних користувач перенаправляється до головного меню (рис. 4.2). У верхній частині головного екрану розміщено кнопку-банер, яка відображає поточний логін користувача. При натисканні на цей банер повторно відкривається екран входу та реєстрації, що дозволяє користувачеві за потреби змінити обліковий запис.

Основну частину головного меню займають шість великих та чітко позначених кнопок, кожна з яких відповідає за певний розділ застосунку: "Гострота зору", "Аметропія", "Дальтонізм", "Астигматизм" та "Історія результатів". Розмір та розташування кнопок на головному екрані оптимізовані для зручного натискання, а їхні назви є лаконічними та зрозумілими, що полегшує навігацію для користувачів різного віку.



Рисунок 4.2 - Екран головного меню

Тест на гостроту зору починається з інформаційного вікна (рис. 4.3), яке надає користувачеві базове розуміння поняття гостроти зору.

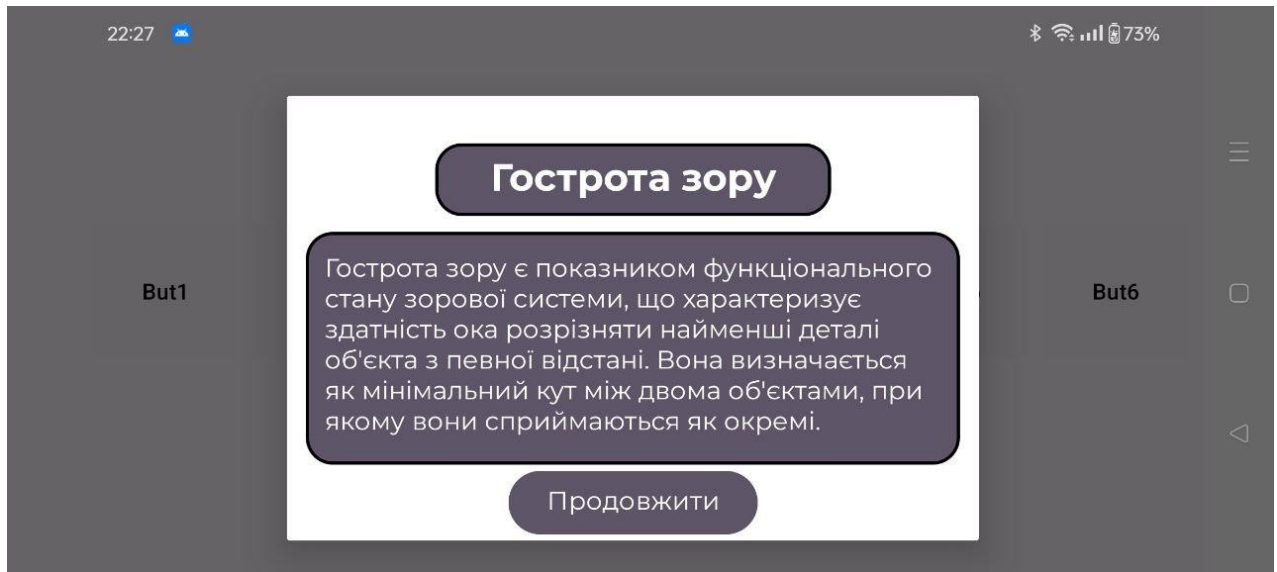


Рисунок 4.3 - Інформаційне вікно в тесті на гостроту зору

Після ознайомлення з цією інформацією та натискання кнопки "Продовжити" відкривається вікно для введення відстані (рис. 4.4), з якої користувач планує проходити тест.



Рисунок 4.4 - Вікно для введення відстані

Застосунок передбачає можливість введення значення від 1 до 120 сантиметрів. У випадку, якщо користувач пропускає введення та натискає "Продовжити", автоматично встановлюється значення за замовчуванням – 50 сантиметрів. Під час самого тесту на екрані, орієнтованому горизонтально, відображаються шість кнопок, розташованих в ряд (рис. 4.5). Кожна кнопка містить одну випадкову латинську літеру. Розмір цих літер динамічно адаптується відповідно до введеної користувачем відстані, що є ключовою особливістю реалізації тесту.

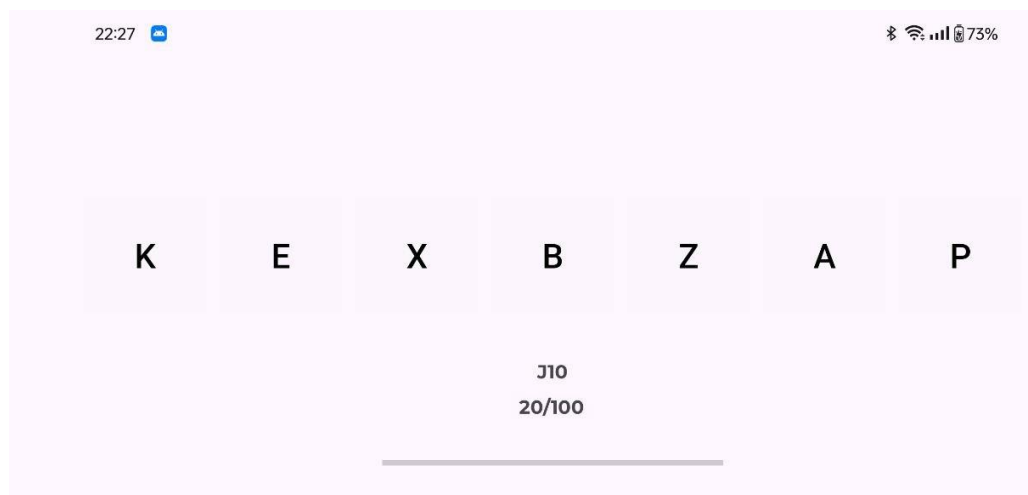


Рисунок 4.5 - Екран тесту на гостроту зору

Екран тесту на аметропію візуально дуже схожий на екран тесту на гостроту зору, але має одну важливу відмінність - колір фону. Спочатку тестування проводиться на червоному фоні (рис. 4.6) для обох очей.

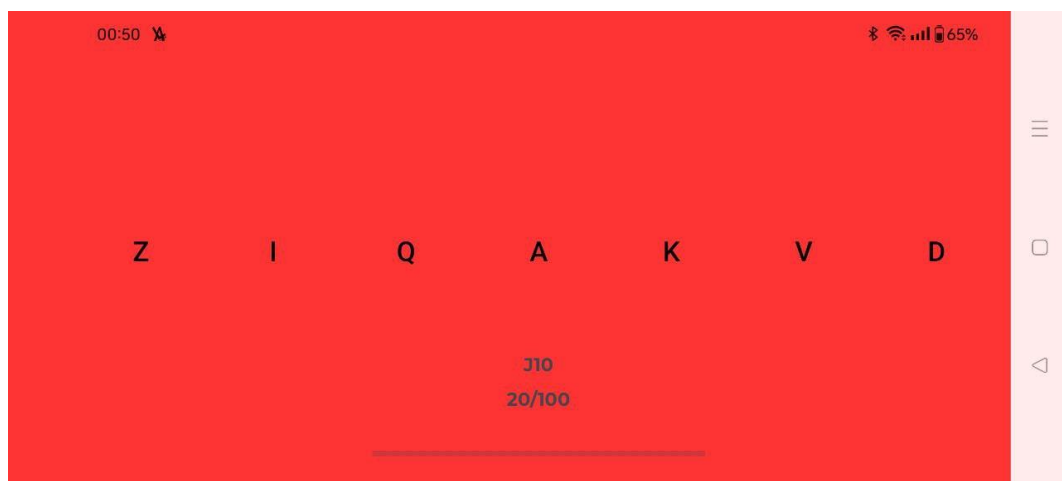


Рисунок 4.6 - Екран тесту на аметропію з червоним фоном

Після завершення цього етапу фон змінюється на зелений (рис. 4.7), і користувачеві пропонується повторно пройти тест для обох очей.

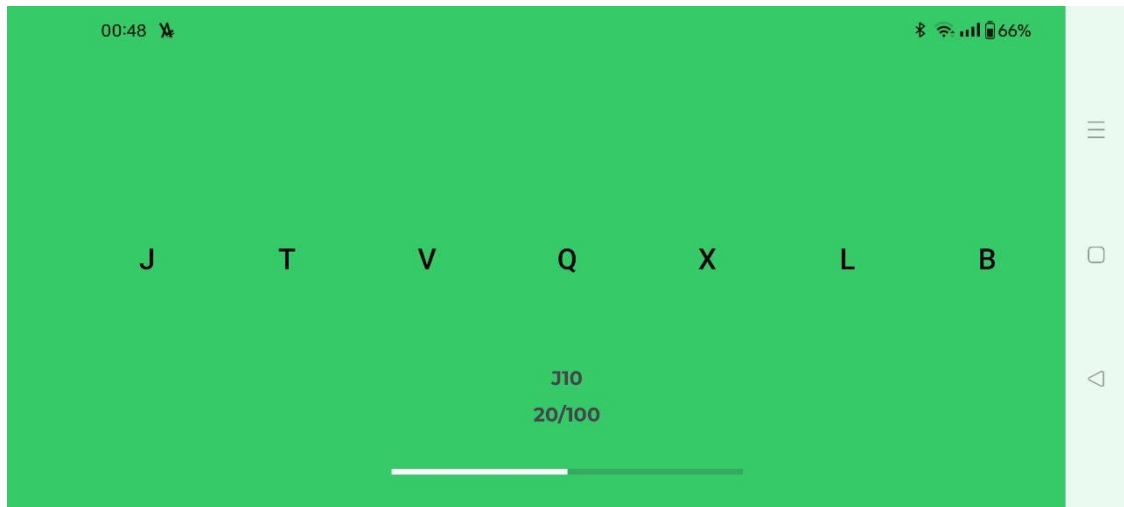


Рисунок 4.7 - Екран тесту на аметропію з зеленим фоном

Екран тесту на дальтонізм (рис. 4.8) відображає одну з таблиць Ішихари - спеціальних кольорових зображень, призначених для виявлення порушень кольорового зору [31].

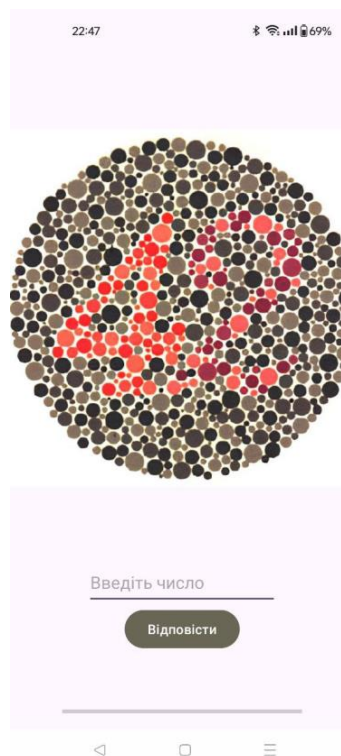


Рисунок 4.8 - Екран тесту на дальтонізм

Під кожним зображенням розміщено поле для введення числа, яке користувач бачить на таблиці, та кнопку "Відповісти" для підтвердження введеного значення.

Екран тесту на астигматизм (рис. 4.9) містить радіальне зображення, що складається з ліній, які розходяться від центра. Під зображенням розташовані дві кнопки з варіантами відповіді: "Всі лінії чітко видно" та "Лінії розмиті та криві". Користувач обирає найбільш відповідний варіант, що відображає його сприйняття зображення.

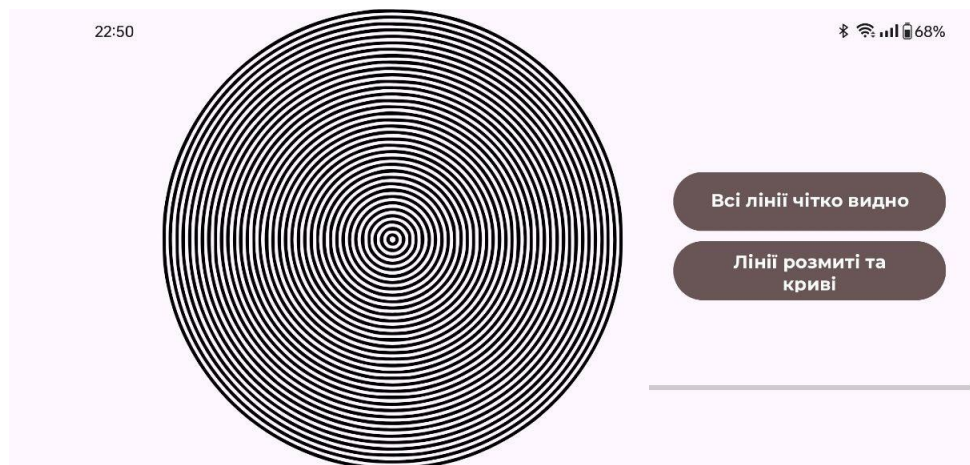


Рисунок 4.9 - Екран тесту на астигматизм

На екрані "Історія результатів" (рис. 4.10) користувач може переглянути хронологічний список усіх раніше пройдених тестів.

Історія результатів	
Тест на гостроту зору - 05.05.2025 01:00	
Ліве око: коефіцієнт зору - 1.0	
Праве око: коефіцієнт зору - 1.0	
Тест на аметропію - 05.05.2025 01:01	
Є вірогідність короткозорості на лівому оці.	
Тест на дальтонізм - 05.05.2025 01:02	
Кількість правильних відповідей: 13 - 68%	
Тест на астигматизм - 05.05.2025 01:02	
Наявність астигматизму не виявлено.	

Рисунок 4.10 - Екран історії результатів

Загальний дизайн інтерфейсу користувача відповідає рекомендаціям платформи Android щодо стилю та навігації, що забезпечує знайомий та зручний досвід для користувачів. Використання чітких візуальних елементів та логічне розташування елементів керування сприяють ефективному використанню всіх функцій застосунку.

4.2 Реалізація тесту на гостроту зору та аметропію

Тести на гостроту зору та аметропію в розробленому застосунку мають багато спільного в своїй реалізації, оскільки вони використовують схожий механізм тестування, заснований на розпізнаванні символів різного розміру.

Алгоритм тесту на гостроту зору починається з відображення інформаційного екрана (рис. 4.4), що надає користувачеві загальне розуміння поняття гостроти зору. Після натискання кнопки "Продовжити" користувачеві пропонується ввести відстань, з якої він буде проходити тест. Застосунок приймає значення відстані в діапазоні від 1 до 120 сантиметрів. У випадку, якщо користувач не вводить жодного значення або натискає "Продовжити" без введення, автоматично встановлюється стандартна відстань у 50 сантиметрів. Введена відстань є критично важливим параметром, оскільки саме на її основі відбувається розрахунок та адаптація розмірів літер, що відображаються під час тестування.

Для забезпечення коректності тестування на різних відстанях застосовується закон Вебера-Фехнера. Цей закон описує логарифмічну залежність між фізичним стимулом і його сприйняттям людиною. Цей принцип широко застосовується в офтальмології для стандартизації тестів на гостроту зору. У медичних тестах на гостроту зору розмір символів стандартизований таким чином, щоб створювати певний кутовий розмір при перегляді зі стандартної відстані. Застосунок використовує цей принцип для

розрахунку необхідного фізичного розміру літер на екрані пристрою, щоб вони створювали той самий кутовий розмір, що й у стандартних таблицях для перевірки гостроти зору, але з урахуванням введеної користувачем відстані [53, 16].

Після введення відстані користувачеві пропонується закрити ліве око (рис. 4.11), і розпочинається сам тест. На екрані, орієнтованому в альбомній орієнтації, відображаються шість кнопок, розташованих горизонтально. На кожній кнопці відображається одна випадково обрана латинська літера. Початковий розмір літер є найбільшим і відповідає гостроті зору 20/100. Кожні три секунди застосунок відтворює голосову команду з назвою літери, на яку користувач повинен натиснути. Якщо користувач натискає неправильну кнопку, літери на всіх кнопках миттєво змінюються на нові випадкові літери, і користувачеві надається ще дві спроби. У випадку правильного натискання кількість спроб скидається до трьох, а розмір літер зменшується, що відповідає наступному рівню гостроти зору 20/70. Тестування триває до тих пір, поки користувач не пройде всі шість рівнів розміру літер або не зробить три послідовні неправильні відповіді.



Рисунок 4.11 - Закрийте ліве око

Після завершення тестування для одного ока користувачеві пропонується закрити праве око (рис. 4.12) та пройти аналогічну процедуру тестування.

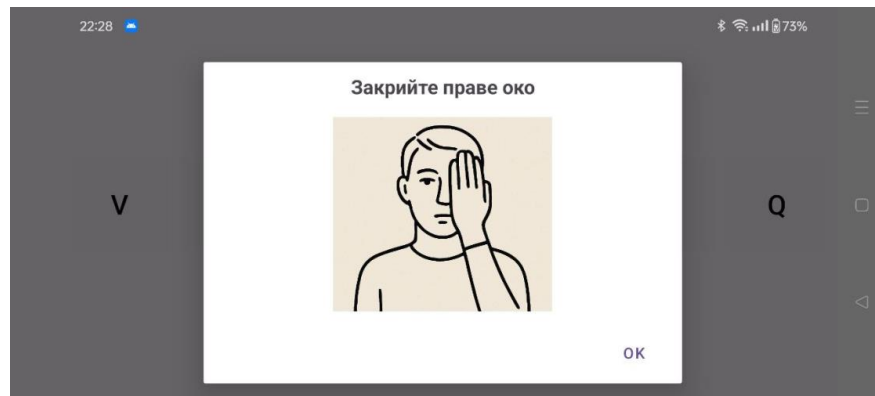


Рисунок 4.12 - Закрийте праве око

Результати для обох очей фіксуються та відображаються на окремому екрані (рис. 4.13) після завершення обох етапів тестування. Після перегляду результатів користувач повертається до головного меню, а інформація про проходження тесту зберігається в розділі "Історія результатів".



Рисунок 4.13 - Вікно результату тесту на гостроту зору

Тест на аметропію реалізований за аналогічним принципом, що й тест на гостроту зору, але з однією ключовою відмінністю. Спочатку тестування проводиться на червоному фоні для обох очей. Після завершення цього етапу тестування для обох очей фон автоматично змінюється на зелений, і користувачеві пропонується повторно пройти тест на гостроту зору для обох очей вже на зеленому фоні. Після завершення обох етапів тестування (на

червоному та зеленому фонах) результати порівнюються. Якщо результати тестування на червоному фоні виявилися кращими, застосунок робить висновок про можливу наявність короткозорості у користувача. Якщо ж кращі результати були отримані на зеленому фоні, робиться висновок про можливу наявність далекозорості. В кінці тесту користувачеві відображається зведений результат (рис 4.14), що включає показники гостроти зору на обох фонах та висновок щодо наявності та типу аметропії.

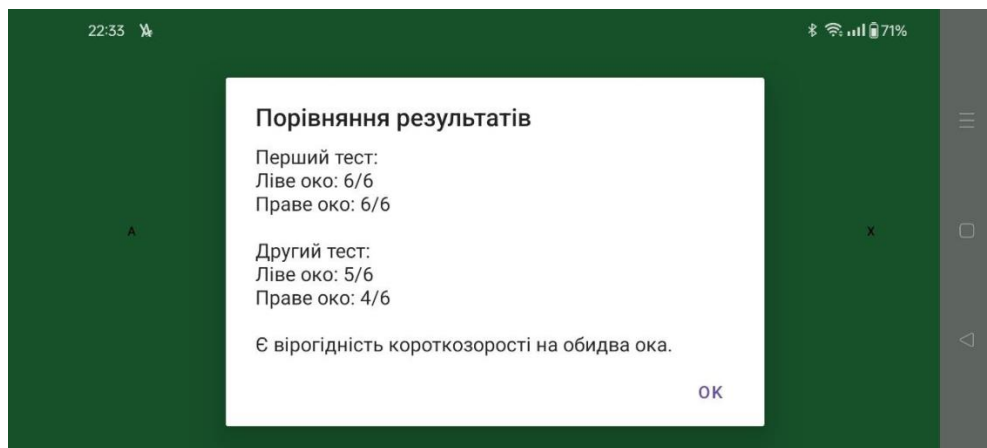


Рисунок 4.14 - Вікно результату тесту на аметропію

4.3 Реалізація тесту на дальтонізм

Тест на дальтонізм у розробленому застосунку реалізовано з використанням набору стандартних таблиць Ішихари. Ці таблиці є загальноприйнятим інструментом для скринінгу порушень кольорового зору, зокрема для виявлення червоно-зеленого дальтонізму [31, 62].

Перед початком тесту користувачеві відображається інформаційне вікно (рис. 4.15), яке коротко пояснює, що таке дальтонізм. На відміну від тестів на гостроту зору та аметропію, тест на дальтонізм не передбачає введення відстані, оскільки таблиці Ішихари не потребують чіткої відстані при проходженні тестувань.

Після ознайомлення з інформацією користувачеві послідовно

демонструються 19 різних таблиць Ішихари. Кожна таблиця являє собою кольорове зображення, що складається з маленьких різнокольорових кружечків, серед яких утворюється число або фігура, видима для людей з нормальним кольоровим зором [31, 62]. Люди з певними типами дальтонізму можуть не бачити цих чисел або бачити інші числа. Під кожною таблицею розміщено поле для введення, в яке користувач повинен ввести число, яке він бачить на зображенні. Після введення числа користувач натискає кнопку "Відповісти", і на екрані з'являється наступна таблиця.



Рисунок 4.15 - Інформаційне вікно в тесті на дальтонізм

Після того, як користувач перегляне та відповість на всі 19 таблиць, застосунок обробляє введені дані. Для кожної відповіді програма порівнює введене користувачем число з правильним значенням для даної таблиці. Після завершення тесту користувачеві відображається результат (рис. 4.16),

який включає загальну кількість правильних відповідей у кількісному вираженні (скільки таблиць було розпізнано правильно) та у відсотковому співвідношенні до загальної кількості таблиць (19). Ця інформація дозволяє отримати уявлення про ймовірність наявності та ступінь вираженості порушень кольорового зору. Після перегляду результатів користувач повертається до головного меню, а результат тесту зберігається в розділі "Історія результатів". Для відображення таблиць Ішихари в застосунку використовуються заздалегідь підготовлені цифрові зображення цих таблиць. Важливо забезпечити якісне відображення кольорів на екрані пристрою, оскільки від точності передачі кольорової гами залежить достовірність результатів тесту.

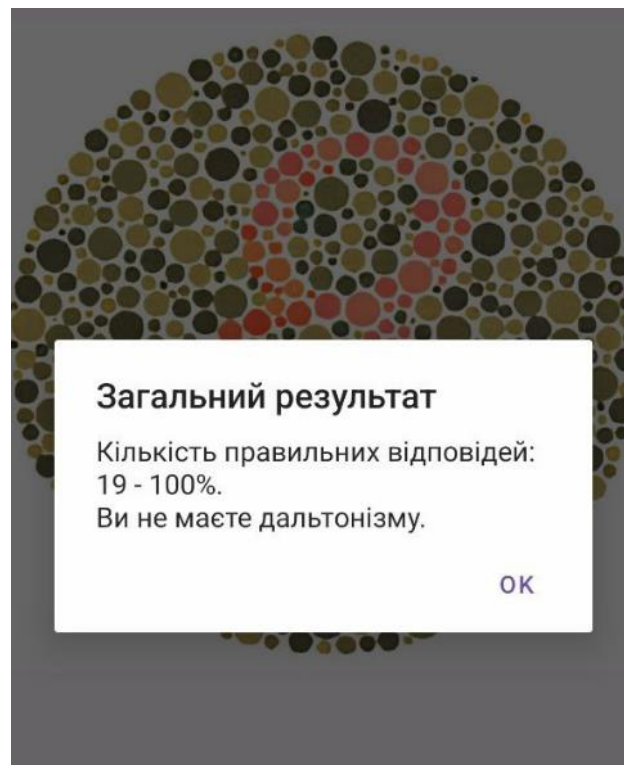


Рисунок 4.16 - Вікно результату тесту на дальтонізм

4.4 Реалізація тесту на астигматизм

Тест на астигматизм у розробленому застосунку використовує набір

радіальних зображень, які є стандартним методом для виявлення нерівномірного викривлення рогівки або кришталика ока, що є основною причиною астигматизму [30].

Перед початком тесту користувачеві відображається інформаційне вікно (рис. 4.17) з поясненням поняття астигматизму. Після ознайомлення з цією інформацією користувачеві послідовно демонструються три різних радіальних зображення. Кожне зображення являє собою одне із різних видів радіальних зображень, для того щоб тестування було більш надійним.



Рисунок 4.17 - Інформаційне вікно в тесті на астигматизм

Під кожним радіальним зображенням розташовані дві кнопки з варіантами відповіді: "Всі лінії чітко видно" та "Лінії розмиті та криві". Користувачеві необхідно уважно розглянути кожне зображення та визначити, чи всі лінії на ньому здаються однаково чіткими та рівними по всій своїй довжині, або ж деякі лінії виглядають розмитими, викривленими чи менш чіткими, ніж інші. Залежно від свого сприйняття користувач повинен натиснути відповідну кнопку.

За кожне радіальне зображення, на якому користувач правильно визначив, що всі лінії видно чітко (у випадку відсутності астигматизму),

йому зараховується один бал. Якщо ж користувач відзначає, що лінії розмиті або криві, бал не зараховується. Після відображення всіх трьох радіальних зображень застосунок підраховує загальну кількість набраних балів. На основі цієї кількості виводиться результат тесту на астигматизм (рис 4.18). Якщо користувач набрав 3 бали, це може свідчити про низьку ймовірність наявності астигматизму. Менша кількість набраних балів може вказувати на можливу наявність астигматизму та рекомендувати користувачеві звернутися за консультацією до фахівця для проведення більш детального обстеження. Після відображення результату користувач повертається до головного меню, а результат тесту зберігається в розділі "Історія результатів". Використання трьох радіальних зображень дозволяє отримати більш надійний результат тесту порівняно з використанням лише одного зображення.

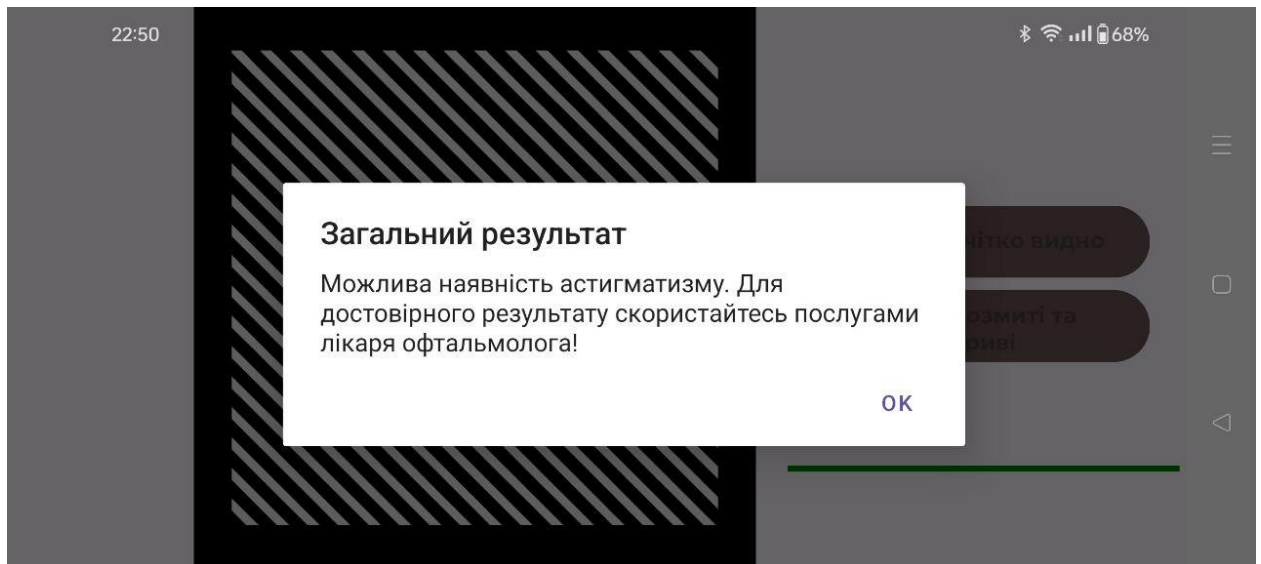


Рисунок 4.18 - Вікно результату тесту на астигматизм

4.5 Реалізація збереження результатів

Функціональність збереження результатів тестів є важливою складовою розробленого застосунку, оскільки вона надає користувачам можливість відстежувати динаміку свого зору з часом та мати доступ до

історії пройдених тестів. Усі результати тестувань зберігаються локально на пристрої користувача в контексті його індивідуального облікового запису.

Для реалізації локального зберігання даних використовується вбудована в платформу Android система керування реляційними базами даних SQLite. При першому запуску застосунку та успішній реєстрації нового користувача в базі даних створюється відповідний запис, що ідентифікує його обліковий запис.

Кожного разу, коли користувач проходить будь-який з доступних тестів (на гостроту зору, аметропію, дальтонізм або астигматизм), у базі даних створюється новий запис, який містить інформацію про результати цього тесту та пов'язаний з обліковим записом користувача. Для тесту на гостроту зору зберігаються окремі показники коефіцієнту зору для лівого та правого ока. Для тесту на аметропію фіксується висновок щодо наявності короткозорості або далекозорості (або їх відсутності) на основі порівняння результатів, отриманих на червоному та зеленому фонах. Результати тесту на дальтонізм зберігаються у вигляді загальної кількості правильних відповідей та їх відсоткового співвідношення до загальної кількості пред'явлених таблиць Ішихари. Для тесту на астигматизм зберігається загальна кількість балів, набраних користувачем за результатами оцінки радіальних зображень. Окрім самих результатів тестів, у кожному записі також фіксується дата та час проходження відповідного тесту.

При натисканні користувачем на кнопку "Історія результатів" застосунок здійснює запит до локальної бази даних для отримання всіх записів, пов'язаних з поточним обліковим записом користувача. Отримані дані відображаються у вигляді впорядкованого списку (рис. 4.10), де для кожного пройденого тесту вказується його тип, дата та час проходження, а також основні результати.

Використання локальної бази даних для зберігання результатів тестів забезпечує конфіденційність персональних даних користувача, оскільки вся інформація зберігається безпосередньо на його пристрої та не передається на

жодні зовнішні сервери. Крім того, такий підхід дозволяє користувачам мати доступ до своєї історії результатів навіть за відсутності активного підключення до мережі Інтернет.

4.6 Реалізація адаптації розмірів символів

Одним із важливих аспектів коректної перевірки гостроти зору є відповідність розміру тестових символів до відстані, з якої користувач їх переглядає. Для цього у мобільному застосунку реалізовано адаптацію розмірів символів залежно від введеної користувачем відстані від екрана. Ця адаптація базується на фізіологічному законі Вебера-Фехнера, що описує логарифмічну залежність між інтенсивністю фізичного подразника та суб'єктивним сприйняттям [53, 16].

Закон Вебера-Фехнера визначається рівнянням:

$$S = k * \log\left(\frac{R}{R_0}\right), \quad (4.1)$$

де:

S - суб'єктивне відчуття (наприклад, помітність або сприйняття розміру),

R - фізична інтенсивність подразника (в нашому випадку — реальний розмір символу),

R_0 - порогове значення подразника,

k - коефіцієнт пропорційності, що залежить від типу відчуття [16, 64].

У контексті тесту на зір, для збереження однакової видимості символів при різній відстані D , необхідно масштабувати їх розмір так, щоб кутовий розмір символу залишався постійним [64]. Кутовий розмір θ (в радіанах) визначається як:

$$\theta = \frac{h}{D}, \quad (4.2)$$

де:

h - висота символу,

D - відстань від ока до екрана.

Отже, для постійного кутового розміру потрібно:

$$h = \theta * D \quad (4.2)$$

Таким чином, розмір символу h прямо пропорційний відстані D [61], що цілком узгоджується з логікою закону Вебера-Фехнера: зростання подразника (розміру) компенсує зменшення чутливості (віддалення) [38].

Цей підхід дозволяє створити умови, близькі до стандартного тестування зору, де відстань до таблиці та розмір символів строго регламентовані. Завдяки масштабуванню, тест адаптується до будь-якої дистанції, яку зручно встановити в домашніх умовах, що робить систему гнучкою та універсальною.

4.7 Розрахунок економічного ефекту

В цьому розділі програмний застосунок для перевірки зору в польових та домашніх умовах розглядається як об'єкт функціонально-вартісного аналізу (ФВА). Цей підхід дозволяє оцінити не тільки технічну реалізацію програмного продукту, але й надати економічне обґрунтування його розробки та впровадження.

Розроблений програмний застосунок призначений для використання на пристроях під управлінням ОС Android та реалізована в середовищі Android Studio мовою програмування Java. Система включає функції перевірки гостроти зору, виявлення ознак міопії, гіперопії, дальтонізму та астигматизму. Важливою складовою є система керування обліковими записами користувачів та збереження результатів тестування з прив'язкою до відповідного акаунту.

Продукт може бути використаний як інструмент для самодіагностики в домашніх умовах, а також для скринінгової перевірки зору в польових умовах (наприклад, в освітніх закладах, на підприємствах, під час виїзних медичних оглядів).

Використання ФВА дозволяє оцінити різні варіанти реалізації функціональних можливостей системи, обрати оптимальний шлях розробки з точки зору співвідношення витрат та якості, а також визначити потенційні шляхи для економії ресурсів без шкоди для функціональності та надійності продукту. Аналіз охоплює як вибір технологічних рішень, так і оцінку економічних показників проекту.

Морфологічна карта системи, що відображає можливі комбінації варіантів реалізації функцій, наведена на (рис. 4.19).

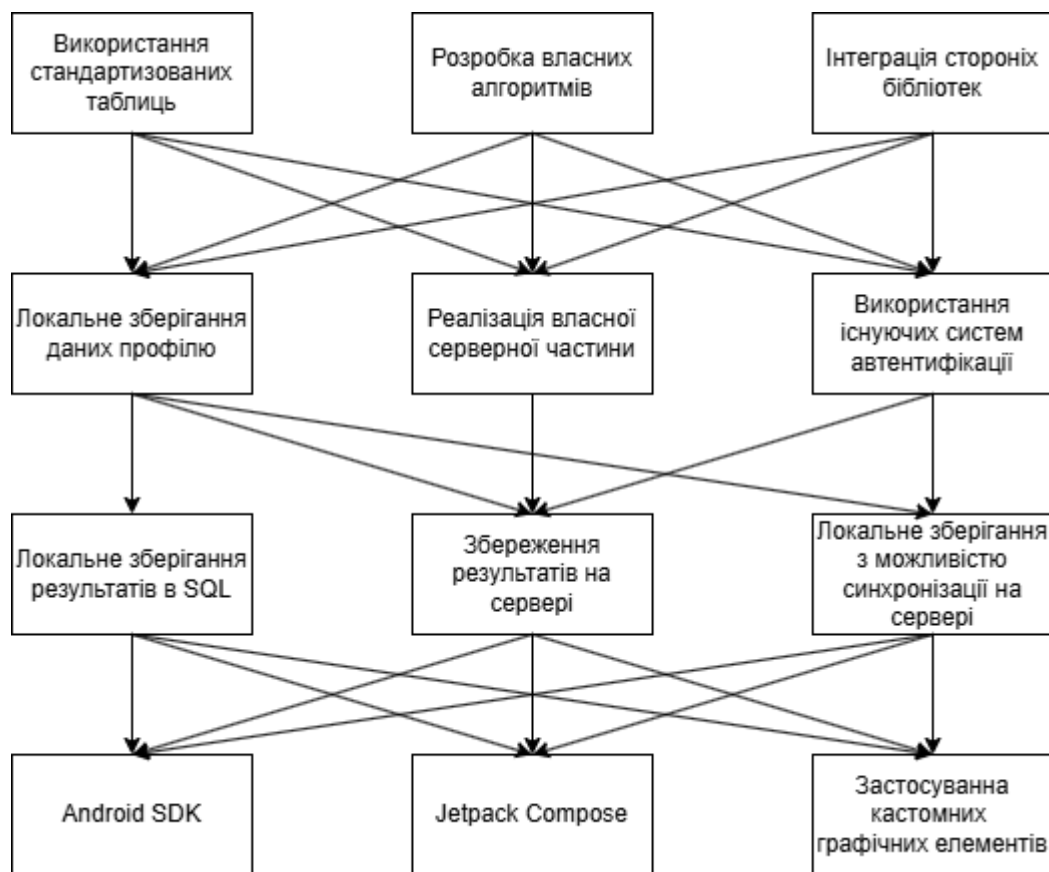


Рисунок 4.19 - Морфологічна карта програмного застосунку для перевірки зору

Ретельний економічний розрахунок, докладно представлений у звіті з переддипломної практики, дав змогу встановити таке:

Здійснений техніко-економічний аналіз показав, що на створення інформаційної системи для перевірки зору в домашніх умовах потрібно

287242,24 грн. Основна частина цієї суми припадає на оплату праці розробників та витрати на використання обчислювальних потужностей. Отримане значення коефіцієнта техніко-економічного рівня — $4,6 \times 10^{-5}$ — свідчить про раціональне використання ресурсів, оскільки воно відображає ефективність реалізованих функцій у порівнянні з витратами. Це підтверджує економічну доцільність розробки та впровадження програмного забезпечення.

Висновок до розділу 4

У четвертому розділі було детально розглянуто особливості реалізації основних функціональних модулів інформаційної системи для перевірки зору. Було проаналізовано принципи побудови інтерфейсу, логіку проходження тестів, реалізацію голосових підказок та обробку результатів користувача. Особливу увагу приділено адаптації розміру символів відповідно до закону Вебера–Фехнера, що дозволило створити гнучкий механізм масштабування тестових елементів з урахуванням відстані до екрана. Таким чином, реалізація системи забезпечує високу точність відображення візуального контенту відповідно до офтальмологічних стандартів, зберігаючи водночас зручність для користувача. Розроблена архітектура є модульною, розширюваною та готовою до подальших удосконалень.

Аналіз економічного ефекту дав розуміння, що розроблений програмний продукт не лише задовольняє основні вимоги до точності, зручності та ефективності, але й є економічно обґрунтованим, що свідчить про доцільність його подальшого впровадження та можливу комерціалізацію.

РОЗДІЛ 5

ТЕСТУВАННЯ, ЗБЕРЕЖЕННЯ ДАНИХ ТА КОНФІДЕНЦІЙНІСТЬ

В цьому розділі розглядаються заходи тестування працездатності застосунку, організація збереження результатів у локальній базі даних та забезпечення конфіденційності персональної медичної інформації. Розглянуто захист акаунтів користувачів за допомогою хешування паролів, відсутність обміну даними з мережею, що відповідає сучасним вимогам безпеки.

5.1 Тестування функціональності та точності

Після завершення основного етапу реалізації інформаційної системи було проведено комплексне тестування, метою якого стало виявлення логічних, інтерфейсних, адаптивних і функціональних помилок. Особливу увагу приділено коректності динамічного масштабування символів відповідно до введеної користувачем відстані до екрана. Згідно із законом Вебера–Фехнера, сприйняття розміру символів залежить не лише від фізичних розмірів об’єктів на екрані, а й від їх відносного кутового розміру [64]. Тому реалізована система враховує цю залежність шляхом математичного перерахунку розміру символів у пікселях залежно від введеної дистанції. Це дозволяє користувачу проходити тести за умов, наближених до офтальмологічних стандартів.

У ході тестування було перевірено всі реалізовані модулі: тест на гостроту зору, тест на аметропію, визначення кольорової сліпоты, тест на астигматизм, а також навігаційні елементи, блоки інструкцій і підказок, голосовий супровід та систему облікових записів. Окрему увагу було приділено сценаріям повторного проходження тестів, коли користувач

повинен пройти тест спочатку на одне око, а потім на інше, з подальшим порівнянням результатів. Було враховано як правильність логіки зміни фону між етапами, так і збереження коректних даних у відповідному обліковому записі.

Результати тестування показали високу стабільність роботи системи на пристроях із різною діагоналлю, щільністю пікселів і версіями Android. Це підтверджує коректну реалізацію механізмів адаптації інтерфейсу до характеристик екрана, що особливо важливо для медичного застосунку, який не повинен допускати спотворень чи зміщень при відображенні критично важливої інформації.

5.2 Організація збереження результатів тестування

Інформація про проходження тестів користувачем повинна зберігатися для подальшого аналізу, виявлення динаміки змін зору, а також для формування висновків щодо ефективності проведеного самотестування. У межах проєкту реалізовано локальне збереження даних, що передбачає запис результатів на пристрій у вигляді асоційованих записів, прив'язаних до конкретного облікового запису. Кожен запис містить назву тесту, дату і час проходження, кількість правильних відповідей, підсумкову інтерпретацію.

Ця модель дозволяє сформувати повну історію тестувань, що може бути використана користувачем для моніторингу стану зору в динаміці. Враховуючи те, що додаток не потребує підключення до мережі Інтернет, усі дані залишаються виключно на пристрої користувача. З технічного боку це реалізовано за допомогою вбудованої системи збереження даних, що дозволяє забезпечити швидкий доступ до даних і їхню безпеку від несанкціонованого доступу.

Користувач має змогу переглядати результати у власному профілі, не порушуючи конфіденційності інших облікових записів. Інтерфейс історії тестування реалізовано у зручному списковому форматі з відповідними

позначками типу тесту, що пройдено. Для кожного облікового запису реалізована ізоляція даних, що унеможлиблює доступ до інформації інших користувачів на одному пристрої.

5.3 Захист персональних даних

З огляду на те, що результати тестування зору можуть містити персональну медичну інформацію, особливу увагу у проєкті приділено забезпеченню конфіденційності. На відміну від більшості сучасних мобільних застосунків, що вимагають доступу до зовнішніх серверів, геолокації, контактів або камер, розроблена система функціонує повністю автономно. Жодна частина даних не передається за межі пристрою, що повністю виключає можливість витоку інформації через мережеві з'єднання.

Система облікових записів захищена паролем, який зберігається у хешованому вигляді з використанням сучасних криптографічних алгоритмів, а саме SHA-256 [76]. Це унеможлиблює відновлення пароля у відкритому вигляді навіть при прямому доступі до файлів збереження. У разі потреби користувач може змінити пароль або створити новий профіль. Крім того, програма не зберігає жодної особистої інформації, окрім логіна та пароля, що мінімізує потенційні ризики втрати конфіденційних даних.

Таким чином, реалізована інформаційна система не лише відповідає функціональним вимогам до перевірки зору, а й демонструє високий рівень відповідальності щодо обробки, збереження та захисту персональних даних. Це робить її придатною до використання широким колом осіб, зокрема дітьми, літніми людьми та особами з обмеженим доступом до медичних послуг, без ризику компрометації особистої інформації.

Висновок до розділу 5

Розділ було присвячено критично важливим аспектам експлуатації інформаційної системи - тестуванню працездатності, збереженню результатів та захисту конфіденційної інформації. Проведене тестування підтвердило стабільність і точність функціонування системи в різних умовах. Реалізовані механізми локального збереження результатів дозволяють користувачам зберігати історію проходження тестів без підключення до Інтернету, що є особливо актуальним у контексті мобільного використання. Конфіденційність забезпечується автономністю системи, шифруванням паролів і відсутністю обробки персональних даних за межами пристрою. Таким чином, реалізація тестування, збереження даних і захисту конфіденційності підтверджує придатність системи до широкого використання в особистих і професійних цілях без ризику витоку інформації.

ЗАГАЛЬНІ ВИСНОВКИ

Узагальнюючи виконану роботу, можна ствердити, що поставлена мета - вдосконалення процесу самостійної первинної діагностики зору в домашніх та польових умовах шляхом розробки мобільного програмного забезпечення на платформі Android, що забезпечує проведення інтерактивних тестів на гостроту зору, аметропію, кольоросприйняття та астигматизм, досягнута повністю. У процесі дослідження було здійснено всебічний аналіз існуючих офтальмологічних методик і нормативної бази, обґрунтовано вибір технологічного стеку Android Studio + Java та локального сховища SQLite із шифруванням, розроблено архітектуру з чітким поділом на модулі (аутентифікація, тести, історія результатів), реалізовано чотири клінічно релевантні тести (гострота зору, аметропія, дальтонізм, астигматизм) з адаптацією розміру символів до введеної дистанції та голосовими підказками, а також підтверджено економічну доцільність проєкту.

Практичні результати демонструють, що застосунок забезпечує точність вимірювань, співставну з класичними офтальмологічними інструментами, зберігає дані у захищеному локальному середовищі й надає користувачеві інтуїтивно зрозумілий інтерфейс. Водночас отримані статистичні дані вказують на перспективність подальшої інтеграції хмарного бекенду для синхронізації результатів, впровадження алгоритмів машинного навчання для персоніфікованих рекомендацій та розширення тестової батареї відповідно до новітніх клінічних протоколів.

Таким чином, дипломна робота не лише доводить реалістичність і життєздатність запропонованого рішення, а й закладає фундамент для подальших досліджень у галузі мобільної офтальмології, підвищуючи доступність базових перевірок зору для широкого кола користувачів та сприяючи ранньому виявленню очних патологій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. About DICOM: Overview [Електронний ресурс] // DICOM Standard. – 2025.
– Режим доступу: <https://www.dicomstandard.org/about>
2. Accessibility in UX Design: Guidelines and Training [Електронний ресурс] // AudioEye. – 2025. – Режим доступу:
<https://www.audioeye.com/post/accessibility-ux-design/>
3. Acuity Pro Vision Testing Software: Version 10 License [Електронний ресурс] // AcuityPro. – 2025. – Режим доступу:
<https://acuitypro.com/product/acuity-pro-vision-testing-software-version10/>
4. Acuity Testing at Home: Ready for Clinical Use? [Електронний ресурс] // Review of Ophthalmology. – 2025. – Режим доступу:
<https://www.reviewofophthalmology.com/article/acuity-testing-at-home-ready-for-clinical-use>
5. Android MVVM — how to use MVVM in android example ? [Електронний ресурс] // Medium. – 2023. – Режим доступу:
<https://medium.com/@dheerubhadoria/android-mvvm-how-to-use-mvvm-in-android-example-7dec84a1fb73>
6. Assessment of visual function using mobile Apps [Електронний ресурс] // PubMed Central. – 2025. – Режим доступу:
<https://pmc.ncbi.nlm.nih.gov/articles/PMC11306548/>
7. Astigmatism Testing: Online Test for Symptoms & Causes [Електронний ресурс] // NVISION Eye Centers. – 2025. – Режим доступу:
<https://www.nvisioncenters.com/astigmatism-lasik/astigmatism-testing/>
8. camera | Flutter package [Електронний ресурс] // Pub.dev. – 2025. – Режим доступу: <https://pub.dev/packages/camera>
9. Clarifying recommendations on Android internal storage encryption #3077 [Електронний ресурс] // GitHub Discussions. – 2025. – Режим доступу:
<https://github.com/OWASP/owasp-mastg/discussions/3077>
10. Duochrome Test [Електронний ресурс] // EyeWiki. – 2025. – Режим

доступу: https://eyewiki.org/Duochrome_Test

11. Effectiveness of Kotlin vs. Java in Android App Development Tasks [Электронный ресурс] // ResearchGate. – 2025. – Режим доступа: https://www.researchgate.net/publication/342659047_Effectiveness_of_Kotlin_vs_Java_in_Android_App_Development_Tasks
12. Eye Health Screening Apps May Signal Need for Eye Exam [Электронный ресурс] // American Academy of Ophthalmology. – 2025. – Режим доступа: <https://www.aao.org/eye-health/tips-prevention/eye-health-app-covid-coronavirus-home-exam>
13. EyeQue Insight Lite – Innovative Vision Solutions [Электронный ресурс] // EyeQue. – 2025. – Режим доступа: <https://eyeque.com/insight-lite/>
14. FAQs – Innovative Vision Solutions [Электронный ресурс] // EyeQue. – 2025. – Режим доступа: <https://eyeque.com/faqs/>
15. Fast Healthcare Interoperability Resources [Электронный ресурс] // Wikipedia. – 2025. – Режим доступа: https://en.wikipedia.org/wiki/Fast_Healthcare_Interoperability_Resources
16. Fechner, G. T. Elemente der Psychophysik. Leipzig, 1860.
17. Flutter vs Native: Which is Best in 2025 [Электронный ресурс] // GeeksforGeeks. – 2025. – Режим доступа: <https://www.geeksforgeeks.org/flutter-vs-native/>
18. Free Online Eye Tests for Astigmatism at Home [Электронный ресурс] // Ophthalmology24. – 2025. – Режим доступа: <https://www.ophthalmology24.com/eye-tests-for-astigmatism>
19. General Data Protection Regulation [Электронный ресурс] // Wikipedia. – 2025. – Режим доступа: https://en.wikipedia.org/wiki/General_Data_Protection_Regulation
20. google-cloud/text-to-speech vs react-native-tts [Электронный ресурс] // NPM Compare. – 2025. – Режим доступа: <https://npm-compare.com/@google-cloud/text-to-speech,react-native-tts>
21. Health Insurance Portability and Accountability Act [Электронный ресурс] //

- American Speech-Language-Hearing Association. – 2025. – Режим доступа: <https://www.asha.org/practice/reimbursement/hipaa/>
22. How To Choose the Best Text-to-Speech API for 2025 [Электронный ресурс] // Vonage. – 2025. – Режим доступа: <https://www.vonage.com/resources/articles/text-to-speech-api/>
23. How to Reduce React Native App Size? [Электронный ресурс] // Aglowid IT Solutions. – 2025. – Режим доступа: <https://aglowiditsolutions.com/blog/how-to-reduce-react-native-app-size/>
24. Innovative Vision Solutions – Take Charge Of Your Vision [Электронный ресурс] // EyeQue. – 2025. – Режим доступа: <https://eyeque.com/>
25. Is Flutter Suitable for Medical Device Software? [Электронный ресурс] // ICS. – 2025. – Режим доступа: <https://www.ics.com/blog/flutter-suitable-medical-device-software>
26. Is there still a reason to use Java instead of Kotlin for Android development in 2023? [Электронный ресурс] // Reddit. – 2025. – Режим доступа: https://www.reddit.com/r/androiddev/comments/10nq5wf/is_there_still_a_reason_to_use_java_instead_of/
27. Ishihara 14 Plate Instructions [Электронный ресурс] // Stanford University. – 2025. – Режим доступа: <https://web.stanford.edu/group/vista/wikiupload/0/0a/Ishihara.14.Plate.Instructions.pdf>
28. Ishihara charts : 14-plate test [Электронный ресурс]. – Режим доступа: https://www.challengetb.org/publications/tools/country/Ishihara_Tests.pdf
29. Ishihara Test – Test for Color Blindness Online [Электронный ресурс] // ColorBlindTest.ai. – 2025. – Режим доступа: <https://colorblindtest.ai/ishihara-test/>
30. Ishihara test [Электронный ресурс] // Wikipedia. – 2025. – Режим доступа: https://en.wikipedia.org/wiki/Ishihara_test
31. Ishihara, S. *Tests for Colour-Blindness*. – Tokyo: Kanehara & Co., Ltd., 2012.
32. Java for Mobile App Development: A Complete Guide in 2025 [Электронный ресурс]

- ресурс] // Leanware. – 2025. – Режим доступа:
<https://www.leanware.co/insights/guide-for-java-mobile-app-development>
33. Java Software Development Services for Healthcare [Электронный ресурс] // Tymiq. – 2025. – Режим доступа: <https://www.tymiq.com/services/java-software-development-services-for-healthcare>
34. JSON Healthcare Formats [Электронный ресурс]. – 2025. – Режим доступа: <https://json-healthcare.org/>
35. Kotlin vs Java: The Best Android App Development Language [Электронный ресурс] // Appinventiv. – 2025. – Режим доступа: <https://appinventiv.com/blog/java-or-kotlin-android-app-development/>
36. LogMAR Scoring [Электронный ресурс] // Thomson Software Solutions. – 2025. – Режим доступа: <https://www.thomson-software-solutions.com/OnlineResources/Screening%20Test%20Chart/LogMARScoring.html>
37. LogMAR VA Charts [Электронный ресурс] // Optonet Project. – 2025. – Режим доступа: <https://optonet.online/logmar-va-charts/>
38. logMAR Visual Acuity [Электронный ресурс] // Optonet. – 2024. – Режим доступа: <https://optonet.online/guide/?section=logmar-visual-acuity>
39. LooC - Mobile Eye Test : мобільний застосунок [Электронный ресурс] // App Store. – 2025. – Режим доступа: <https://apps.apple.com/us/app/looc-mobile-eye-test/id405950873>
40. Mastering SQLite Encryption for Securing Sensitive Data [Электронный ресурс] // DataSunrise. – 2025. – Режим доступа: <https://www.datasunrise.com/knowledge-center/sqlite-encryption/>
41. Model -View-ViewModel [Электронный ресурс] // Wikipedia. – 2025. – Режим доступа: <https://uk.wikipedia.org/wiki/Model-View-ViewModel>
42. Native Mobile Apps vs. Flutter & FlutterFlow: an Honest Comparison [Электронный ресурс] // Glance. – 2025. – Режим доступа: <https://thisisglance.com/learning-centre/native-mobile-apps-vs.-flutter-flutterflow-an-honest-comparison>

43. PC - based and Desktop Vision Testers [Електронний ресурс] // Amplivox. – 2025. – Режим доступу: <https://www.amplivox.com/vision>
44. Peek Acuity : мобільний застосунок [Електронний ресурс] // Google Play. – 2025. – Режим доступу: <https://play.google.com/store/apps/details?id=org.peekvision.public.android>
45. Peek Acuity [Електронний ресурс] // Peek Vision. – 2025. – Режим доступу: <https://peekvision.org/solutions/peek-acuity/>
46. Pilot testing a smartphone app for astigmatism measurement [Електронний ресурс] // Investigative Ophthalmology & Visual Science. – 2023. – Режим доступу: <https://iovs.arvojournals.org/article.aspx?articleid=2787068>
47. Portable hardware & software technologies for addressing ophthalmic health disparities: a systematic review [Електронний ресурс] // PubMed Central. – 2022. – Режим доступу: <https://pmc.ncbi.nlm.nih.gov/articles/PMC9087242/>
48. Pros and Cons of Java Development in 2023 [Електронний ресурс] // Netguru Blog. – 2025. – Режим доступу: <https://www.netguru.com/blog/java-pros-and-cons>
49. React Native vs Native App Performance for Healthcare Startups [Електронний ресурс] // Technology Rivers. – 2025. – Режим доступу: <https://technologyrivers.com/blog/react-native-vs-native-app-performance-healthcare/>
50. React Native vs Native: Which is Better for App Development? [Електронний ресурс] // Appinventiv. – 2025. – Режим доступу: <https://appinventiv.com/blog/react-native-vs-native-apps/>
51. Refraction Technology and The Future – Innovative Vision Solutions [Електронний ресурс] // EyeQue. – 2025. – Режим доступу: <https://eyeque.com/refraction-technology-the-future/>
52. Smart System Standard [Електронний ресурс] // M&S Technologies. – 2025. – Режим доступу: <http://mstech-eyes.com/smart-system-standard/>
53. Subjective Refraction Technique: Astigmatic Dial [Електронний ресурс] // StatPearls, NCBI Bookshelf. – 2025. – Режим доступу:

<https://www.ncbi.nlm.nih.gov/books/NBK589695/>

54. The 6 Best Online Eye Exams of 2025 [Електронний ресурс] // OnlineDoctor. – 2025. – Режим доступу: <https://www.onlinedoctor.com/best-online-eye-exams/>
55. The 8 Best Online Eye Exams [Електронний ресурс] // eyeSTYLE Blog, Lens.com. – 2025. – Режим доступу: <https://www.lens.com/eyestyle/featured/the-8-best-online-eye-exams/>
56. The History of the Development of the Modern Eye Chart [Електронний ресурс] // ISCeys. – 2024. – Режим доступу: <https://isoeyes.com/blog/2024/08/30/the-history-of-the-development-of-the-modern-vision-test/>
57. Top 5 Kotlin Pros and Cons You Need to Know About [Електронний ресурс] // Pangea.ai. – 2025. – Режим доступу: <https://pangea.ai/resources/top-kotlin-pros-and-cons-you-need-to-know-about>
58. Vision Test [Електронний ресурс] // MyEyes.ai. – 2025. – Режим доступу: <https://myeyes.ai/>
59. Vision test validation: the Peek Acuity app [Електронний ресурс] // Peek Vision. – 2025. – Режим доступу: <https://peekvision.org/resources/research/validating-our-vision-test/>
60. Vision Tests : мобільний застосунок [Електронний ресурс] // Google Play. – 2025. – Режим доступу: <https://play.google.com/store/apps/details?id=com.EmKsa.TestedeVisao>
61. Visual Acuity [Електронний ресурс] // WikiLectures. – 2020. – Режим доступу: https://www.wikilectures.eu/w/Visual_acuity
62. Visual acuity [Електронний ресурс] // Wikipedia. – 2025. – Режим доступу: https://en.wikipedia.org/wiki/Visual_acuity
63. Waggoner Diagnostics [Електронний ресурс] // WaggonerDiagnostics.com. – 2025. – Режим доступу: <https://waggonerdiagnostics.com/>
64. Weber, E. H. De pulsu, resorptione, auditu et tactu. Annotationes anatomicae et physiologicae, 1834.

65. Welch Allyn Spot Vision Screener [Електронний ресурс] // Hillrom. – 2025. – Режим доступу: <https://www.hillrom.com/en/products/spot-vision-screener/>
66. What are the advantages of Java for mobile application development? [Електронний ресурс] // Quora. – 2025. – Режим доступу: <https://www.quora.com/What-are-the-advantages-of-Java-for-mobile-application-development>
67. What are the benefits of developing mobile apps on iOS (Objective-C) vs Android (Java)? [Електронний ресурс] // Quest Community. – 2025. – Режим доступу: <https://www.quest.com/community/it-security-search/f/forum/33615/what-are-the-benefits-of-developing-mobile-apps-on-ios-objective-c-vs-android-java>
68. What are the pros and cons of using Kotlin over Java for a project? [Електронний ресурс] // Reddit. – 2025. – Режим доступу: https://www.reddit.com/r/AskProgramming/comments/1chcj6x/what_are_the_pros_and_cons_of_using_kotlin_over/
69. Why Choose React Native for Effective Healthcare App Development [Електронний ресурс] // ClarionTech. – 2025. – Режим доступу: <https://www.clariontech.com/blog/why-react-native-for-healthcare-app-development>
70. Аметропія: види, причини та методи корекції захворювань ока. [Електронний ресурс] // ЕКСІМЕР – 2022. – Режим доступу: <https://eximer.ua/uk/blog/2022/01/10/ametropiya-vidi-prichini-ta-metodi-korekcii/>
71. База даних [Електронний ресурс] // Wikipedia. – 2025. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/База_даних
72. Гострота зору: що вважається ідеальним зором? [Електронний ресурс] // ОСНІ Clinic – 2020. – Режим доступу: <https://www.ochiclinic.com.ua/publications/gostrota-zoru-shho-vvazhayetsya-idealnim-zorom/>
73. Дальтонізм [Електронний ресурс] // Wikipedia. – 2025. – Режим доступу

до ресурсу: <https://uk.wikipedia.org/wiki/Дальтонізм>

- 74.Короткозорість або далекозорість або емметропія (норма)? [Електронний ресурс] // Новий зір. – 2025. – Режим доступу: <https://novyizir.ua/test-duokhromnyy/>
- 75.МОЗ України. Уніфікований клінічний протокол медичної допомоги: Порушення рефракції та акомодатії [Електронний ресурс]. – 2015. – Режим доступу: https://www.dec.gov.ua/wp-content/uploads/2019/11/2015_827_ykpm_d_porref.pdf
- 76.Огляд алгоритму шифрування SHA-256 [Електронний ресурс] // CryptoHacker – 2018. Режим доступу до ресурсу: <https://www.cryptohackers.club/2018/09/oglyad-algorytmu-shyfruvannya-sha-256.html>