

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
імені ВАДИМА ГЕТЬМАНА»

Навчально-науковий інститут «Інститут інформаційних технологій в економіці»

Кафедра інформаційних систем в економіці

Форма навчання: заочна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ПРОЕКТ

на тему

«Мобільний застосунок розпізнавання тексту, об'єктів навколишнього середовища та грошей для людей с вадами зору»

здобувача Троян Ганни Костянтинівни _____

Науковий керівник:

к.е.н., доц.

_____ Зінов'єва І. С.

Кваліфікаційний бакалаврський проект допущений до захисту в Екзаменаційній комісії з атестації здобувачів вищої освіти

В.о. завідувача кафедри ІСЕ: к.е.н., доц. _____ Тішков Б.О.

Київ 2022

АНОТАЦІЯ

кваліфікаційного бакалаврського проекту студентки 4 курсу
Навчально-наукового інституту «Інститут інформаційних технологій в
економіці»

Троян Ганни Костянтинівни, виконаного на тему:
**«Мобільний застосунок розпізнавання тексту, об'єктів навколишнього
середовища та грошей для людей з вадами зору»**

Київ: кафедра інформаційних систем в економіці, 2022 р.

Кваліфікаційний бакалаврський проект спрямований на розроблення мобільного застосунку для людей з вадами зору, що має функції розпізнавання об'єктів, грошей, тексту.

Кваліфікаційний бакалаврський проект складається з трьох розділів, логічно пов'язаних між собою.

В першому розділі дана характеристика предметної галузі й об'єкта дослідження, наведено аналіз практичного досвіду використання ІС і технологій в предметній галузі, а також перелік існуючих програмних рішень у досліджуваній галузі.

Другий розділ присвячений розробці вимог і моделюванню інформаційної системи, аналізу і специфікації вимог, моделюванню процесу роботи та функціонування системи, трасуванню вимог.

Третій розділ містить результати проектування та приклади реалізації розробленого програмного продукту. В ньому наведено програмне забезпечення для мобільного застосування з функціями розпізнавання тексту, об'єктів навколишнього середовища та грошей для людей з вадами зору. Наведено проект інтерфейсу інформаційної системи. Описано етапи реалізації інформаційної системи та зазначено результати реалізації.

Висновки містять рекомендації щодо доцільності розроблення та впровадження мобільного застосування з функціями розпізнавання тексту, об'єктів навколишнього середовища та грошей для людей з вадами зору.

РЕФЕРАТ

Кваліфікаційний бакалаврський проект містить 78 сторінок, 5 таблиць, 34 рисунки, список літератури з 39 найменувань, 3 додатки.

Тема кваліфікаційного бакалаврського проекту **«Мобільний застосунок розпізнавання тексту, об'єктів навколишнього середовища та грошей для людей з вадами зору»**.

Ключові слова: програмний продукт, моделювання, вимоги, проектування, мобільний застосунок, технології, інформаційна система, розпізнавання, Android, JavaScript, Ruby.

Предметом дослідження є ІС (мобільний додаток) з функціями розпізнавання об'єктів, грошей, тексту для людей з вадами зору.

Об'єктом дослідження виступають інформаційні технології, які забезпечують процеси моделювання, проектування та розробки інформаційних систем.

Мета кваліфікаційного бакалаврського проекту полягає в моделюванні та розробці мобільного додатку, який користувач зможе застосувати як асистента, що дозволить швидко і легко розпізнати предмети (об'єкти), які оточують у повсякденному житті, розпізнати номінал українських гривень та прочитати український текст.

Завданнями кваліфікаційного бакалаврського проекту є:

- характеристика та аналіз предметної області;
- визначення вимог до системи, у т.ч. бізнес-вимог, функціональних та нефункціональних вимог із відповідними діаграмами;
- моделювання інформаційної системи, у т.ч. моделювання поведінки, структури, користувацького інтерфейсу системи;
- проектування та реалізація інформаційної системи.

Результати, досягнуті під час реалізації – готовий проект ІС з функціями розпізнавання об'єктів, грошей, тексту для людей з вадами зору.

Рік виконання і захисту курсового проекту: 2022.

ЗМІСТ

<u>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ</u>	5
<u>ВСТУП</u>	8
<u>РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ</u>	10
<u>1.1 Характеристика предметної галузі та об'єкта дослідження</u>	10
<u>1.2 Аналіз практичного досвіду використання ІС і технологій в предметній галузі</u>	14
<u>РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ</u>	21
<u>2.1 Аналіз і специфікація вимог до інформаційної системи</u>	21
<u>2.2 Постановка задачі</u>	27
<u>2.3 Моделювання інформаційної системи</u>	32
<u>2.3.1. Моделювання процесу роботи мобільного додатку</u>	32
<u>2.3.2. Моделювання функціонування системи</u>	35
<u>2.3.3. Моделювання структури системи</u>	38
<u>2.3.4. Трасування вимог до системи</u>	39
<u>РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ</u>	41
<u>3.1. Інформаційне забезпечення</u>	41
<u>3.2. Технічне забезпечення</u>	43
<u>3.3. Програмне забезпечення</u>	45
<u>3.4. Результати реалізації інформаційної системи</u>	51
<u>3.4.1. Прототипування інтерфейсу інформаційної системи</u>	51
<u>3.4.2. Результати реалізації мобільного застосунку</u>	56
<u>ВИСНОВКИ</u>	61
<u>ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ</u>	62
<u>ДОДАТКИ</u>	67

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ

- ІС – інформаційна система.
- ОС – операційна система.
- ОС Android – операційна система і платформа для мобільних телефонів та планшетних комп'ютерів.
- TalkBack – це програма зчитування з екрана Google, яка входить у пристрої Android. TalkBack дає змогу керувати пристроєм без очей.
- React Native – фреймворк для кросплатформної розробки мобільних додатків.
- API – (Application Programming Interface) прикладний програмний інтерфейс.
- Ruby – мова програмування.
- JavaScript – мова програмування.
- Java – мова програмування.
- JVM – Java Virtual Machine – віртуальна машина для виконання байт-коду Java.
- Kotlin – мова програмування.
- Ruby on Rails – фреймворк для розробки веб-додатків, створений мовою програмування Ruby.
- Digital Ocean – американська організація-постачальник хмарної інфраструктури.
- IDE – (Integrated development environment) Інтегроване середовище розробки.
- Google Play – магазин додатків від компанії Google, що дозволяє власникам пристроїв з операційною системою Android купувати і встановлювати різні програми.
- Vision AI – vision artificial intelligence – штучний інтелект спрямований на штучне бачення.
- ПЗ – програмне забезпечення.
- ПП – програмний продукт.
- Діаграма ВВ – Діаграма варіантів використання.
- Додаток – мобільне застосування.
- UML – (Unified Modeling Language) уніфікована мова моделювання.
- IDEF0 – методологія функціонального моделювання (англ. function modeling) та графічна нотація, призначена для формалізації та опису бізнес-процесів.
- TensorFlow – це безкоштовна бібліотека програмного забезпечення з відкритим вихідним кодом для машинного навчання та штучного інтелекту.
- ONNX – Open Neural Network Exchange – це екосистема штучного інтелекту з відкритим вихідним кодом технологічних компаній і дослідницьких організацій, які встановлюють відкриті стандарти для представлення алгоритмів машинного навчання та програмних засобів для просування інновацій та співпраці в секторі штучного інтелекту.
- Google Vision – програмний інтерфейс для машинного навчання створений компанією Google.

Cloud66 – служба для розгортання, керування та масштабування будь-якої програми в будь-якій хмарі або власних серверах.

DevOps (development і operations) – низка практик, призначених для поєднання взаємодії розробників із фахівцями інформаційно-технологічного обслуговування та зближення їхніх робочих процесів одне з одним.

iOS – це власницька мобільна операційна система від компанії Apple.

GitHub – один з найбільших веб-сервісів для спільної розробки програмного забезпечення.

Apache CouchDB – це документно-орієнтована база даних NoSQL з відкритим вихідним кодом, реалізована на мові Erlang.

Node.js – платформа, для розробки високопродуктивних мережевих застосунків.

Android Studio – це офіційне інтегроване середовище розробки для операційної системи Android від Google, побудоване на програмному забезпеченні IntelliJ IDEA від JetBrains і розроблене спеціально для розробки Android.

RubyMine – це спеціальне середовище розробки Ruby та Rails.

Figma – векторний онлайн-сервіс розробки інтерфейсів та прототипування з можливістю організації спільної роботи, що розробляється однойменною компанією.

Adobe Illustrator – професійний графічний редактор для створення та редагування векторної графіки від компанії Adobe.

Java Development Kit – це дистрибутив технології з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript.

DeviceAtlas – стандарт для виявлення пристроїв і даних, а також єдине рішення, яке індексується для Інтернету, програм і середовищ мобільних операторів.

Ruby-Talk – тематична технічна конференція.

TIOBE індекс (рейтинг мов програмування) – показник популярності мов програмування, створений і підтримуваний компанією TIOBE.

Ліцензія MIT (англ. MIT License) – група ліцензій, розроблених Массачусетським технологічним інститутом для розповсюдження вільного програмного забезпечення.

Шаблон Model-View-Controller – Модель-вигляд-контролер архітектурний шаблон, який використовує Java від Oracle Corporation. Він реалізує специфікацію мови Java і специфікацію віртуальної машини Java і надає стандартну версію інтерфейсу програмного програмування Java.

HTTP – протокол передачі даних, що використовується в комп'ютерних мережах. Назва скорочена від HyperText Transfer Protocol, протокол передачі гіпертекстових документів.

ВСТУП

Актуальність теми. Сучасні інформаційні технології вирішують багато задач та проблем у повсякденному житті людей, серед іншого – виступають у ролі допоміжних інструментів – асистентів. На цей час проводиться багато досліджень у галузі ІТ задля підвищення доступності та усунення бар'єрів для людей з вадами зору, в тому числі, проводиться розробка відповідних мобільних додатків. Необхідність таких розробок пов'язана з розв'язанням соціально-побутових проблем, які виникають у людей з вадами зору під час дослідження навколишнього світу. Серед них: читання будь-яких друкованих текстів, пошук та визначення предметів щоденного вжитку, розпізнавання номіналу грошових коштів. Одним зі способів розв'язання окреслених проблем є розробка мобільного застосунку «асистента», який допоможе людям із вадами зору розпізнавати предмети, які її оточують, зокрема, номінал грошей, допомагати читати та розпізнати текст.

Аналіз останніх досліджень і публікацій. Ступінь теоретичного вивчення сучасного стану розробок у сфері інформаційних технологій та, зокрема, мобільних застосунків для людей із вадами зору, їхнього ефективного застосування, наразі є не повною. Однак проблема підвищення доступності та усунення бар'єрів для людей з вадами зору за допомогою інформаційних технологій розглянута в працях: Аккара Дж.Д. (Akara J. D.) [26], Айзін В. [27], Зубченко С. [29], Мартінез К. (Martinez K.) [30], Носенко Ю. [28] та ін.

Мета і завдання дослідження. Метою кваліфікаційного бакалаврського проекту є розробка мобільного застосунку, що дозволяє людям із вадами зору швидко й легко розпізнавати предмети (об'єкти), які оточують у повсякденному житті, зокрема – номінал українських гривень та українськомовний текст. Для досягнення поставленої мети необхідно виконати наступні завдання:

- 1) визначити характеристику та здійснити аналіз предметної області;
- 2) розробити вимоги й змодельовати інформаційну систему;

3) спроектувати та реалізувати всі компоненти інформаційної системи;

4) надати методичні рекомендації та пропозиції щодо використання результатів дослідження у практиці.

Об'єктом дослідження є автоматизований процес розпізнавання зображень та текстів.

Предметом дослідження є створення інформаційної системи, що забезпечує автоматизацію процесу розпізнавання об'єктів, тексту.

Методологія та методи дослідження. Методологічною основою дослідження є теоретичні положення, висновки та праці українських та іноземних науковців із проблем, які пов'язані з розвитком інформаційних систем для людей з вадами зору. Для дослідження теоретичних аспектів розвитку галузі освіти та впровадження в неї інформаційних технологій було використано методи порівняння і аналізу. Також досліджено практичний досвід в галузі сучасних розробок для людей з вадами зору за допомогою методів порівняння, аналізу та синтезу. Для практичної частини роботи були використані комп'ютерні технології для створення мобільного додатку, що розпізнає об'єкти, гроші та текст, проектування і моделювання інформаційної системи.

Теоретична, методична та практична значущість отриманих результатів. Значущість роботи полягає у визначенні теоретичних аспектів розвитку інформаційних систем для людей з вадами зору, що дає змогу одержати теоретичні і практичні результати, які містять наукову новизну, зокрема, розроблено мобільний додаток асистент, що допомагає розпізнавати об'єкти навколишнього середовища, гроші та текст.

Апробація результатів досліджень. Ключові результати проведеного дослідження були винесені на обговорення під час участі у: Всеукраїнській науково-практичній конференції «Інформаційні технології та моделювання систем» (м. Житомир, 12 травня 2022 р.), 89-ї щорічній студентській науковій конференції з міжнародною участю «Інноваційні ідеї та проекти для економічного відродження України» (м. Київ, 01 червня 2022 року). В рамках

зазначених заходів підготовлено доповідь на тему «Проектування мобільного застосунку для людей с вадами зору» та тези на тему «Мобільний застосунок розпізнавання тексту, об'єктів навколишнього середовища та грошей для людей с вадами зору» (додаток В).

РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Характеристика предметної галузі та об'єкта дослідження

В Україні проживає близько 65 тисяч сліпих осіб [36]. Загалом у світі нараховується близько 314 мільйонів осіб, що мають порушення зору, викликані різними причинами. 45 мільйонів з них є сліпими. Серед них дітей у віці до 15 років за різними оцінками близько 1,4 мільйона. Більше 85 % осіб, які страждають на різноманітні порушення зору, мешкають в країнах з низьким і середнім рівнем доходу[37]. Для визначення сліпоти в традиційній медицині використовуються шкали.

Повна сліпота визначається як абсолютна відсутність реакції на світло. Тим не менш, у багатьох країнах введено поняття практичної сліпоти.

Практична сліпота — або часткова здатність до зору — стан, коли людина розрізняє світло і темряву, іноді володіє деякою здатністю до сприйняття візуальної інформації, однак ця здатність настільки незначна, що не має практичного значення. У США та багатьох європейських країнах практична сліпота визначається як здатність до зору 20/200, тобто пацієнт повинен перебувати від об'єкта на відстані 20 футів (дорівнює 609,6 см ~ 6 м), щоб бачити його так само чітко, як здатна людина із здоровим зором на відстані 200 футів (близько 60 м). У багатьох країнах, людина, яка має поле зору менше 20 градусів (норма — 180 градусів), також визнається практично сліпою.

Симуляція сліпоти при здоровому зорі зустрічається вкрай рідко і легко розпізнається відповідними контрольними способами перевірки зору. Водночас нерідко зустрічаються випадки перебільшення наявного зниження зору при реально існуючій патології органу зору [37].

Як свідчить практика, люди з вадами зору часто неохоче користуються спеціалізованими (медичними) пристосуваннями для покращення слабкого зору, зокрема: окуляри, ручні пристрої збільшення зображень тощо. Серед багатьох причин виділяються фактори соціальної стигматизації та

заперечення [26]. Тому, в сучасному світі на поруч людям із вадами зору може прийти смартфон, вміння користуватися яким може істотно розширити можливості таких людей.

Після спілкування з респондентами з вадами зору було виявлено що для деяких з них найважче у повсякденному житті це розпізнавання грошей, особливо металевих, нового зразка (1-10 гривень) та деякі паперові. Також важко розпізнавати ціnnики в магазинах та написи, що розташовані вище рівня очей. Фактично неможливо встановити номер маршрутного таксі або автобусу. Щодо грошей складність у тому, що монети майже однакові за розміром та однакові за кольором, не містять додаткових розпізнавальних знаків, які вказували б на номінал за дотиком. Інші респонденти також додали проблему пошуку входу до будівлі. Навігатори не вказують на вхід і не описують його, тому важко зорієнтуватись, коли прийшов до пункту призначення. Важко знаходити речі, які хтось переставив зі звичного місця в інше. Важко наливати гарячу воду у чашку. Щодо тексту, всі додатки що тестувались не розпізнають текст українською мовою.

Розпізнавання банкнот і тексту це надважливе питання для створення більш адаптованих умов взаємодії зі світом для людей з вадами зору.

Розпізнавання об'єктів — це техніка комп'ютерного зору для ідентифікації об'єктів на зображеннях або відео. Розпізнавання об'єктів є ключовим результатом алгоритмів глибокого навчання та машинного навчання. Коли люди дивляться на фотографію чи відео, ми можемо легко помітити людей, об'єкти, сцени та візуальні деталі. Мета – навчити комп'ютер робити те, що є природним для людей: отримати рівень розуміння того, що містить зображення.

Розпізнавання об'єктів є ключовою технологією безпілотних автомобілів, що дозволяє їм розпізнавати знак зупинки або відрізнити пішохода від ліхтарного стовпа. Він також корисний у різноманітних застосуваннях, таких як ідентифікація захворювань у біовізуалізації, промисловий контроль та роботизоване бачення [38].

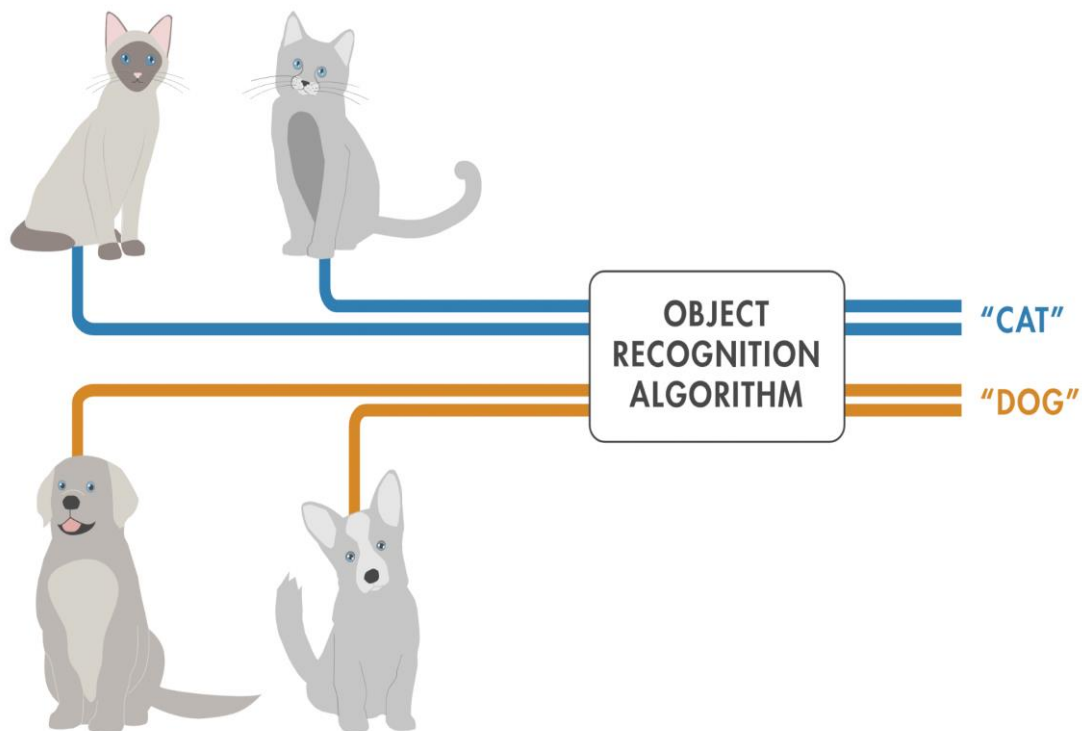


Рисунок 1.1 – Використання розпізнавання об'єктів для визначення різних категорій об'єктів

На даному етапі розвитку ринку програмних рішень для осіб із вадами зору існує велика кількість різноманітних програм, інформаційних систем та мобільних додатків як з відкритою / публічною ліцензією, так і на платній основі. Вони працюють під різними ОС, як то iOS, Android, Windows та ін.

Android (укр. Андроїд) – операційна система і платформа для мобільних телефонів та планшетних комп'ютерів [1]. У 84 % смартфонів, проданих у 3-му кварталі 2014 року, було встановлено операційну систему Android [1]. А з березня 2017 року ОС Android стала найпопулярнішою ОС, з якої виходили в інтернет. Так, 37,93 % користувачів заходили в інтернет із Android. В Азії показники ще вищі – 52,2 %.

Для пристроїв на ОС Android розроблено Android Accessibility Suite, який призначений для допомоги людям з вадами зору. Він забезпечує велике екранне меню керування багатьма найпоширенішими функціями смартфона. За допомогою цього меню можна заблокувати телефон, керувати гучністю та яскравістю, робити знімки екрана, отримати доступ до Google Assistant тощо. Хоча, функції доступності Android включають ряд додаткових

спеціалізованих інструментів, які значно покращують доступність, вони не вирішують всіх проблем, з якими стикаються у повсякденному житті люди, що мають вади зору.

Тому вже багато років проводиться розробка відповідних додатків, що допомагають вирішити питання, на яких не спеціалізуються інструменти операційної системи Android. Деякі з цих розробок допомагають розпізнавати кольори, інші – розпізнавати їжу, побутові предмети, тварин, текст та гроші. Кожен із цих додатків відрізняється своїм функціоналом і задачами, але є корисними так чи інакше.

Засоби призначені для допомоги людям з вадами зору, що постачаються розробником операційної системи Android використовуються більше для взаємодії з мобільним пристроєм, ніж для розв'язання конкретних задач. Ними можна користуватися в якості програмного забезпечення більшості мобільних телефонів. Можливості операційної системи Android для допомоги людям з вадами зору часто використовуються для вирішення більш конкретних задач при розробці мобільних додатків. Так, програмне забезпечення Talk Back замість того, щоб озвучувати, дії, які відбуваються у мобільному пристрої, – у мобільному додатку озвучує розпізнані за допомогою алгоритмів предмети навколишнього середовища, кольори, та таке інше.

Розпізнавання предметів здійснюється за допомогою камери мобільного телефону, шляхом фотографування або відеозапису навколишнього середовища користувача.

Отже, операційна система Android містить набір функціоналу для людей з вадами зору, проте у ній відсутні спеціальні додатки, які були б пристосовані для практичних повсякденних цілей людей з вадами зору.

В зв'язку з чим є доцільним та актуальним розробка мобільних застосувань для вирішення конкретних практичних задач, таких як розпізнавання тексту, об'єктів навколишнього середовища та грошей.

1.2 Аналіз практичного досвіду використання ІС і технологій в предметній галузі

Ринок мобільних додатків для людей з вадами зору порівняно, наприклад, з ринком офісного ПЗ, не дуже великий, але ознайомитись з усіма в рамках даного кваліфікаційного бакалаврського проекту неможливою. Так, в спеціалізованій літературі виділяють такі типи програмних продуктів для людей з вадами зору:

- програми для голосового управління;
- ігри;
- ідентифікатори та зчитувачі на основі камери;
- навігатори;
- електронні лупи;
- зчитувачі електронних книг та ін [26].

У рамках даного дослідження розглянемо й порівняємо можливості деяких мобільних застосунків-асистентів для людей з вадами зору, що доступні в плей-маркеті Google Play.

1) Мобільний додаток BeMyEyes.

BeMyEyes – безкоштовний додаток, який з'єднує незрячих та поганобачущих людей зі зрячими добровольцями та представниками компанії для візуальної допомоги за допомогою відеодзвінка в прямому ефірі (відеочат) [3].

Приклади вікон даного додатку представлені на рис. 1.2.

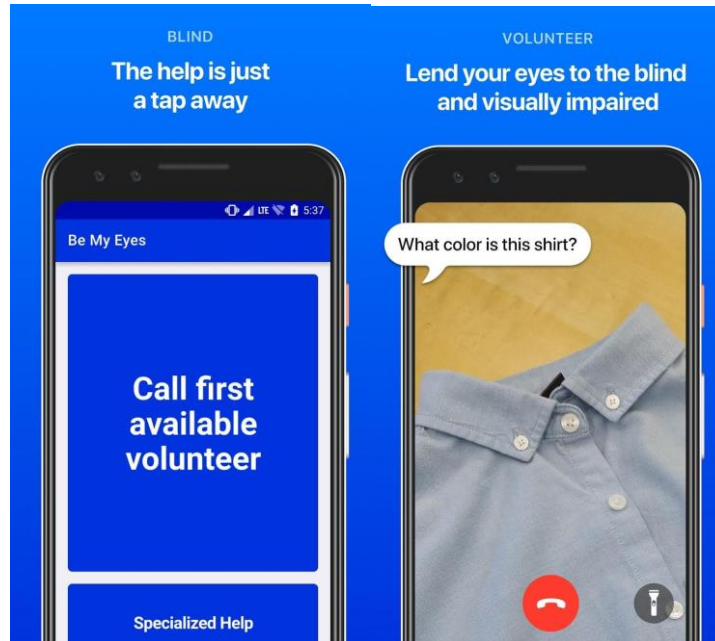


Рисунок 1.2 – Приклади функціоналу мобільного додатку BeMyEyes

Користуючись даним мобільним додатком було виділено наступні переваги:

- простий та зрозумілий мінімалістичний інтерфейс;
- фактично під час відеозв'язку можна запросити будь-яку допомогу (орієнтація у просторі, визначення грошей, підбір кольорів і одягу і таке інше).

Недоліки:

- дзвінок обмежений особистим часом волонтера, який надає візуальну допомогу;
- кожного разу на дзвінок відповідають різні люди, тому, якщо зв'язок перервався, необхідно повторювати запит ще раз;
- питання безпеки – під час оформлення документів чи перерахунку грошей користувач не захищений від зловживань.

Даний мобільний додаток призначений для операційної системи Android від версії 5 і вище.

2) Мобільний додаток Envision AI

Envision AI – додаток для оптичного розпізнавання, який озвучує те, що відбувається в візуальному світі, допомагаючи сліпим і слабкозорячим користувачам вести більш незалежне життя [4].

Даний мобільний додаток призначений для операційної системи Android версій 5 та вище. Додаток доступний в магазині Google Play.

Приклади функціоналу додатку представлені на рис. 1.3.

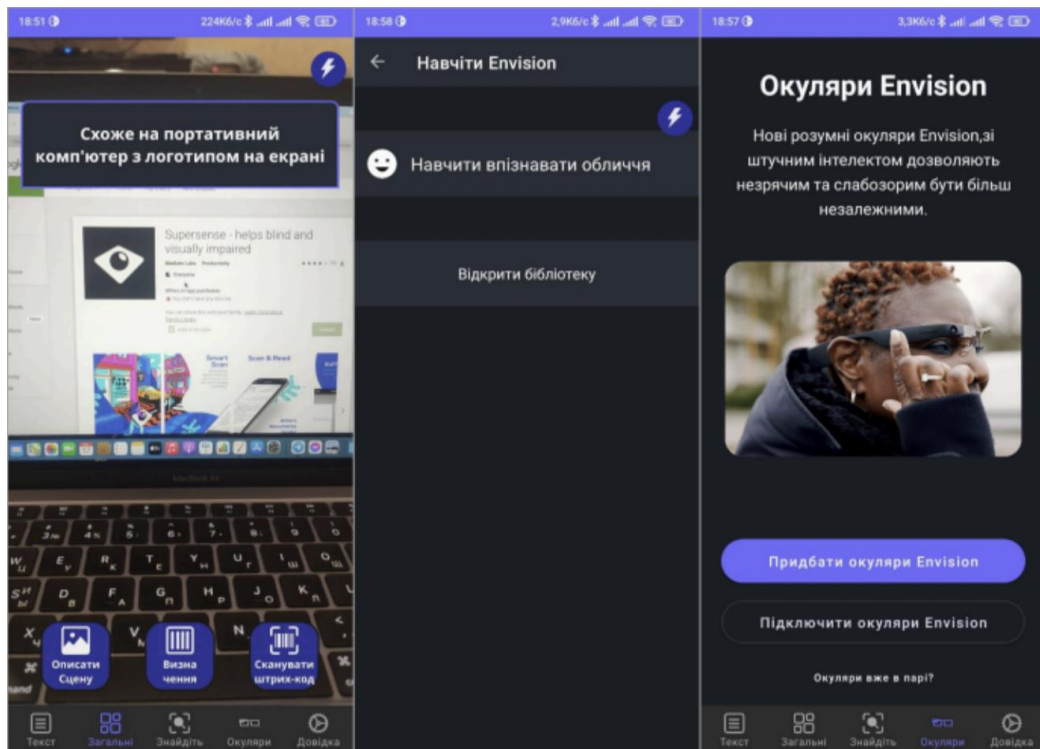


Рисунок 1.3 – Приклади функціоналу додатку Envision AI

Протестувавши мобільний додаток можна виділити переваги:

- різноманітний функціонал, більшість функцій працює коректно;
- функція пошуку предметів – сповіщає звуковим сигналом в якому напрямку рухатися до предмета;
- можливість підключення до додатку додаткового обладнання – окулярів.

У протигагу вищеперерахованим плюсам додатка можна виділити наступні мінуси додатку:

- не розпізнає рукописні тексти як заявлено;
- розпізнає штрих коди лише товарів виготовлених у США;
- функція пошуку людей працює некоректно;

– вартість користування.

3) Мобільний застосунок Sullivan+

Розробник описує Sullivan+, як мобільний додаток для візуальної допомоги, що надається TUAT Corp. для підвищення доступності для людей з вадами зору та незрячих користувачів і інформування користувачів, яким потрібна візуальна допомога про інформацію, яка сприймається за допомогою камери смартфона [5].

Даний мобільний додаток призначений для операційної системи Android версій 5 та вище. Додаток доступний в магазині Google Play.

Приклади функціоналу додатку представлені на рис. 1.4.

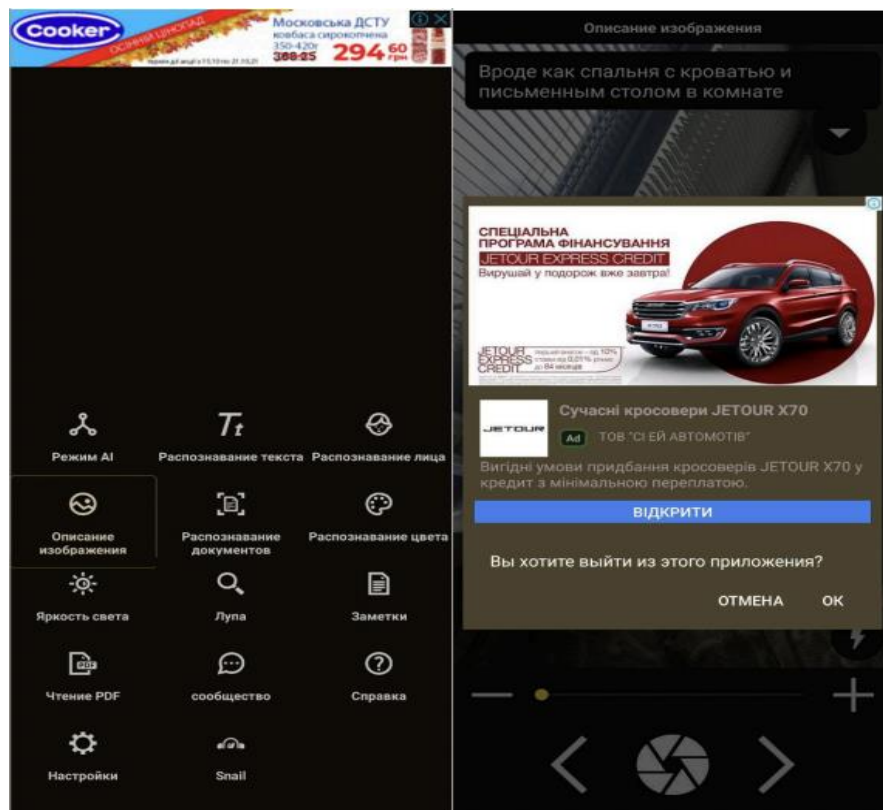


Рисунок 1.4 – Приклади функціоналу додатку Sullivan+

До переваг мобільного додатку можна віднести наступні характеристики:

- процес розпізнавання об'єктів супроводжується звуковим сигналом;
- досить швидко і точно розпізнає об'єкти, додає фрази «здається»;
- керування за допомогою кнопок та жестів;
- додаток в якому працює автофокус.

Після тестування додатку та опрацювання відгуків можна сформулювати такі недоліки:

- банерна реклама на весь екран, важко вийти з додатку;
- мобільний телефон гріється під час роботи з додатком;
- погано розпізнає текст;
- відсутня україномовна версія додатку.

4) Мобільний додаток Aipoly Vision.

Додаток Aipoly Vision позиціонує себе як Vision AI для незрячих та слабоворих [6].

Приклади функціоналу додатку представлені на рис. 1.5.

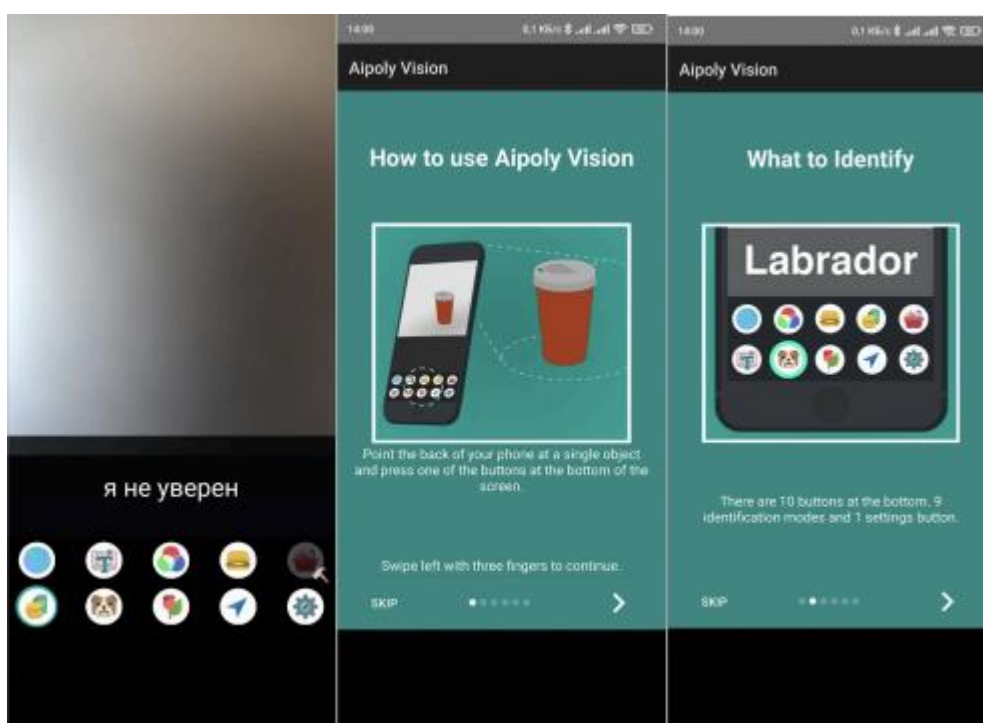


Рисунок 1.5 – Приклади функціоналу додатку Aipoly Vision

Даний мобільний додаток призначений для операційної системи Android версій 5 та вище. Додаток доступний в магазині Google Play.

До особливостей мобільного додатку можна віднести наступні характеристики:

- працює офлайн;

- містить різноманітні функції, серед яких: розпізнавання об'єктів, розпізнавання страв та інгредієнтів, розпізнавання рослин і тварин та ін.

Після тестування додатку та опрацювання відгуків на даний мобільний додаток можна виділити наступні мінуси додатку:

- функція читання текстів не працює, використовували різні тексти та написи на етикетках товарів;
- розпізнати страви не вдалось;
- предмети розпізнає не точно;
- доступна лише англomовна чи російськомовна версія програми;
- висока вартість.

Кожен з розглянутих вище мобільних додатків-аналогів має як певні переваги так і деякі недоліки.

Проаналізувавши вищенаведені додатки встановлено, що всі вони направлені на допомогу незрячим та людям з вадами зору розпізнавати предмети та навколишнє середовище.

Існують рішення, які збігаються із кваліфікаційним бакалаврським проектом за таким функціоналом:

- розпізнавання об'єктів за допомогою камери мобільного телефону;
- розпізнавання текстів;
- розпізнавання грошових коштів.

Серед них можна виділити мобільний додаток Aipoly Vision, проте, він не розпізнає українську гривню.

Частковий збіг, а саме: розпізнавання текстів та розпізнавання об'єктів за допомогою камери мобільного телефону є у таких додатках: Sullivan, Envision+.

Є також рішення, які не обмежені за функціоналом, адже їх робота забезпечується асистентами-людьми.

Фактично, додаток-візуальний комунікатор. В даному дослідженні це – BeMyEyes.

Однак, більшість додатків не містять україномовної версії та мають високу вартість користування.

Отже, можна дійти висновку, що жоден з обраних для аналізу мобільних додатків може бути використаний не всіма українськими користувачами в зв'язку з відсутністю української версії, високою собівартістю та відсутністю реалізації повного переліку заявленого у проекті функціоналу. Найбільш оптимальним рішенням у цьому випадку є розробка власного мобільного додатку, який буде містити наступні функції: розпізнавання об'єктів; розпізнавання текстів; розпізнавання грошових коштів (українська гривня). А також, буде оснащений українським інтерфейсом та розпізнавати тексти українською мовою, буде безкоштовним для використання.

РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Аналіз і специфікація вимог до інформаційної системи

Вимоги до програмного забезпечення – це специфікація того, що повинно бути реалізовано. Це описи поведінки системи, властивості системи або її атрибути. Вони можуть бути обмежені процесом розробки системи.

Карл Вігерс визначає три рівні вимог до програмного забезпечення [7]:

- Бізнес-вимоги – визначають призначення ПЗ, можуть описуватись в документі про бачення (англ. – *vision*) та документі про межі проекту (англ. *scope*).
- Вимоги користувача – визначають набір завдань користувача, які повинна вирішувати програма, а також сценарії їхнього вирішення в системі. Ці вимоги можуть мати вигляд тверджень, варіантів використання, історій користувача, сценаріїв взаємодії.
- Функціональні вимоги – визначають «що» повинен робити програмний продукт. Ці вимоги описуються в документі Специфікація вимог до програмного забезпечення [17].

Види вимог за характером:

1. Функціональний характер – вимоги до поведінки системи:

- Бізнес-вимоги
- Вимоги користувача
- Функціональні вимоги

2. Нефункціональний характер – вимоги до характеру поведінки системи:

- Бізнес-правила – визначають обмеження, що витікають з предметної області.

3. Системні вимоги – вимоги до програмних інтерфейсів, надійності, обладнанню.

4. Атрибути якості.

5. Зовнішні системи та інтерфейси.

6. Обмеження [15].

Розглянемо вимоги до системи мобільного додатку для людей з вадами зору.

Бізнес вимогою мобільного додатку є розробка цифрового інструменту, який забезпечить безбар'єрний доступ людей з вадами зору до дослідження навколишнього світу та визначення об'єктів щоденного вжитку таких як: грошові кошти, побутові предмети, написи та тексти [8].

Бізнес-вимоги до системи продукту зображено на діаграмі, рис. 2.1.

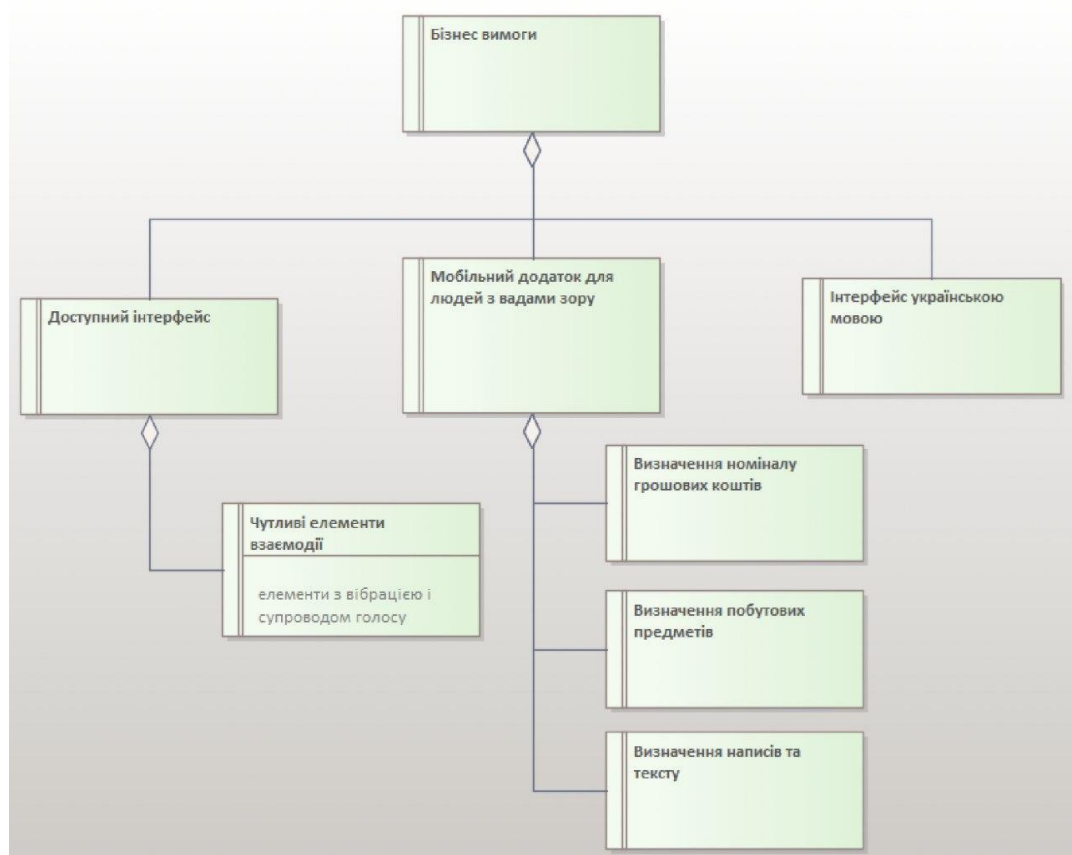


Рисунок 2.1 – Діаграма бізнес-вимог до продукту

Бізнес вимоги до продукту складаються з наступних елементів:

1. Мобільний додаток для людей з вадами зору, що містить вимоги:

1.1. Визначення номіналу грошових коштів;

1.2. Визначення побутових предметів;

1.3. Визначення написів та тексту;

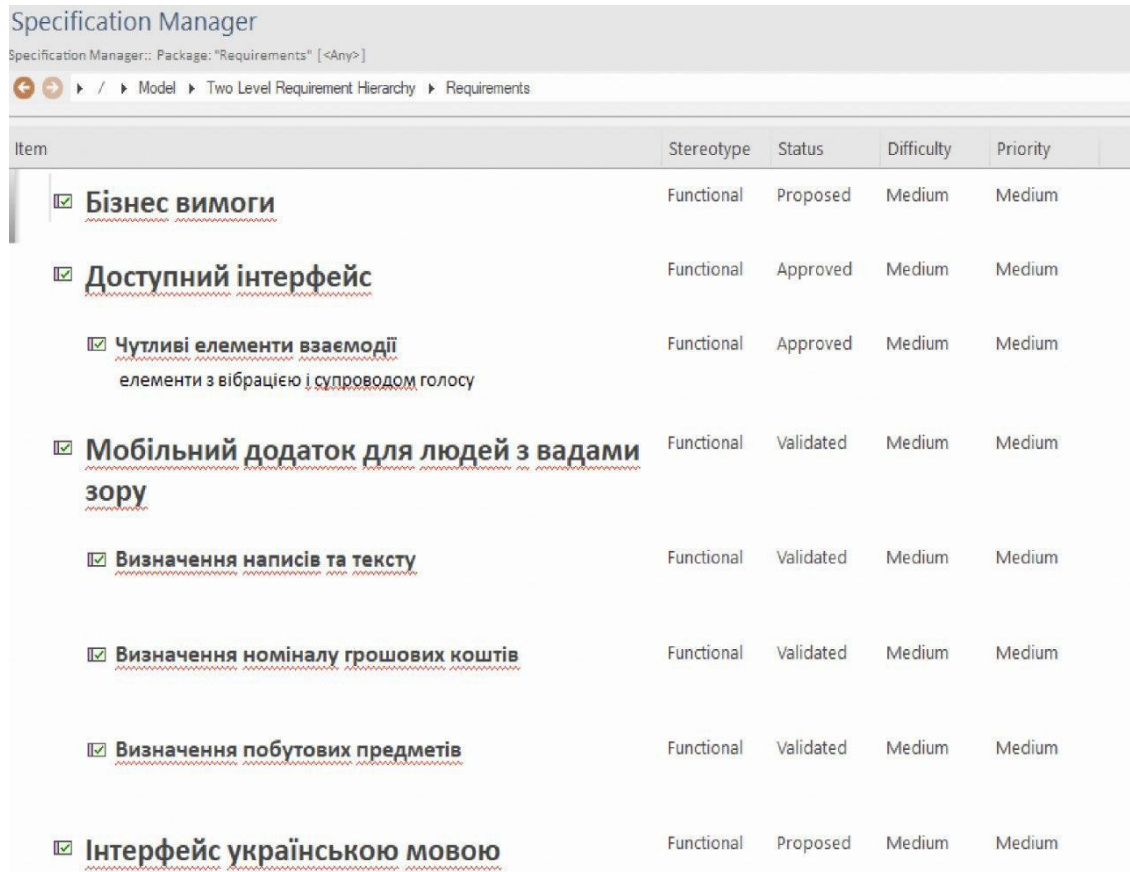
2. Доступний інтерфейс, що містить:

2.1. Чутливі елементи взаємодії – елементи з вібрацією і супроводом голосу;

3. Інтерфейс українською мовою.

Специфікації до бізнес-вимог системи містяться у специфікаціях, рис.

2.2.



Specification Manager
Specification Manager:: Package: "Requirements" [<Any>]

Model ▶ Two Level Requirement Hierarchy ▶ Requirements

Item	Stereotype	Status	Difficulty	Priority
☒ Бізнес вимоги	Functional	Proposed	Medium	Medium
☒ Доступний інтерфейс	Functional	Approved	Medium	Medium
☒ Чутливі елементи взаємодії елементи з вібрацією і супроводом голосу	Functional	Approved	Medium	Medium
☒ Мобільний додаток для людей з вадами зору	Functional	Validated	Medium	Medium
☒ Визначення написів та тексту	Functional	Validated	Medium	Medium
☒ Визначення номіналу грошових коштів	Functional	Validated	Medium	Medium
☒ Визначення побутових предметів	Functional	Validated	Medium	Medium
☒ Інтерфейс українською мовою	Functional	Proposed	Medium	Medium

Рисунок 2.2. – Специфікації до бізнес-вимог проекту

Функціональні вимоги регламентують функціонування або поводження системи. Функціональні вимоги відповідають на запитання, що повинна робити система. Функціональні вимоги визначають основний перелік робіт розробника і встановлюють цілі, задачі, сервіси, що забезпечуються системою для замовника [9].

Функціональні вимоги можуть включати розрахунки, технічні деталі, маніпуляції та обробку даних, а також інші конкретні функції, які визначають, що повинна виконувати система.

Функціональні вимоги до системи продукту зображено на діаграмі, рис. 2.3.

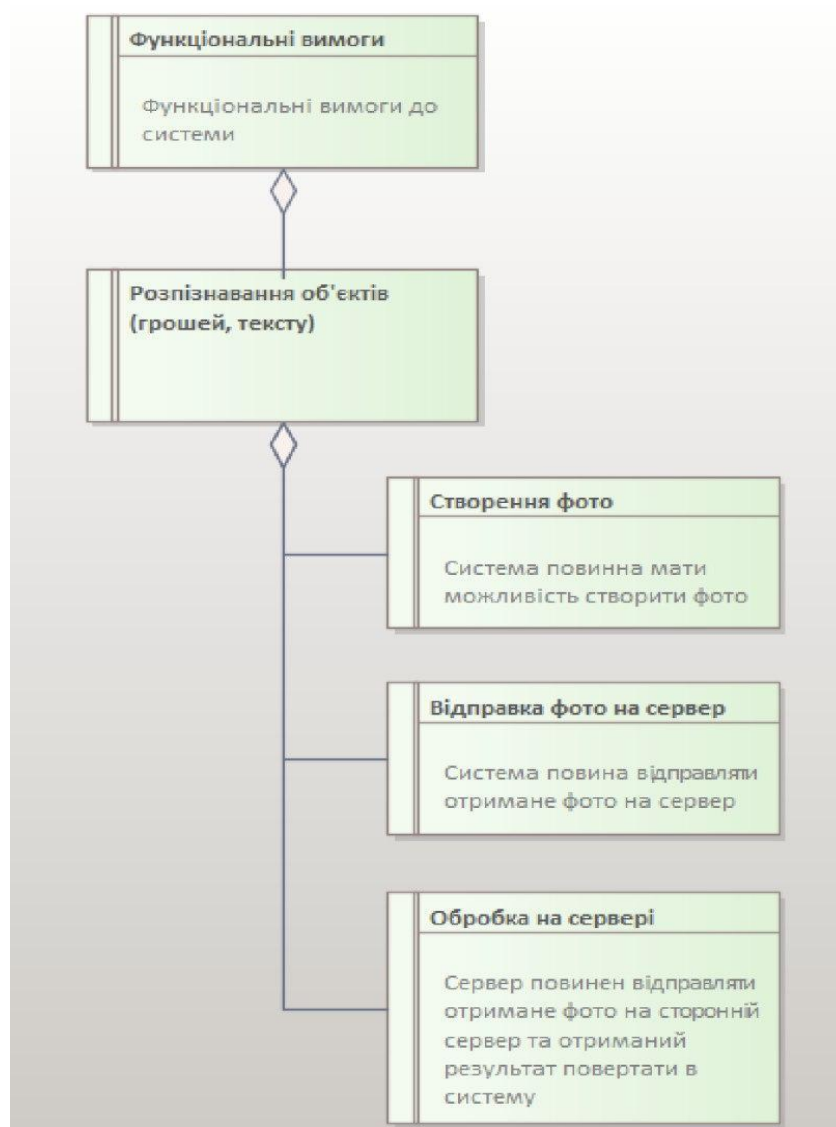


Рисунок 2.3. – Діаграма функціональних вимог до системи

Функціональними вимогами програмного продукту (мобільного додатку) для людей з вадами зору є:

1. Розпізнавання об'єктів – грошей, тексту, предметів;
2. Процес розпізнавання об'єктів відбувається в наступному порядку:
 - 2.1. Створюється фото, – система мобільного додатку повинна мати можливість створення фото;
 - 2.2. Відправка фото на сервер, – система повинна відправляти отримане фото на сервер;

2.3. Обробка на сервері, – сервер повинен відправляти отримане фото на сторонній сервер та отриманий результат повертати в систему.

Специфікацію функціональних вимог містить діаграма на рис. 2.4.

Specification Manager

Specification Manager:: Package: "Requirements" [<Any>]

Model / Two Level Requirement Hierarchy / Requirements

Item	Stereotype	Status	Difficulty	Priority
☒ Функціональні вимоги Функціональні вимоги до системи	Functional	Proposed	Medium	Medium
☒ Розпізнавання об'єктів (грошей, тексту)	Functional	Validated	Medium	Medium
☒ Відправка фото на сервер Система повинна відправляти отримане фото на сервер	Functional	Validated	Medium	Medium
☒ Обробка на сервері Сервер повинен відправляти отримане фото на сторонній сервер та отриманий результат повертати в систему	Functional	Validated	Medium	Medium
☒ Створення фото Система повинна мати можливість створити фото	Functional	Validated	Medium	Medium

Рисунок 2.4. – Специфікація функціональних вимог до системи

Нефункціональні вимоги – це вимоги до програмного забезпечення, які задають критерії для оцінки якості його роботи. На відміну від функціональних вимог, які визначають що система повинна робити, нефункціональні вимоги визначають якою система повинна бути. Нефункціональні вимоги до програмного забезпечення визначаються на першій стадії процесу розробки ПЗ – на етапі аналізу вимог [16].

Нефункціональні вимоги до системи продукту зображено на діаграмі, рис. 2.5.

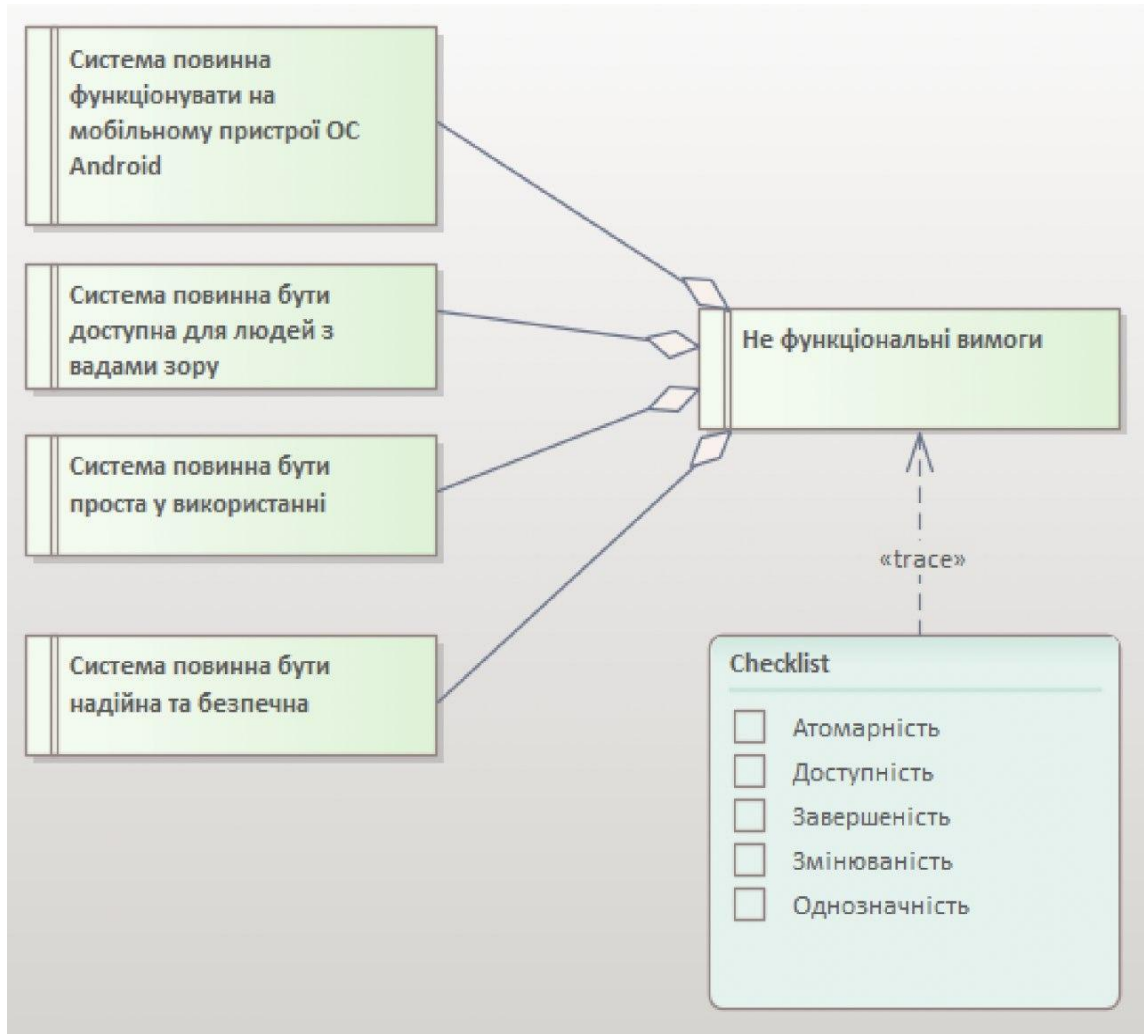


Рисунок 2.5. – Діаграма нефункціональних вимог до системи продукту

Нефункціональними вимогами до мобільного додатку (продукту) для людей з вадами зору є критерії [8]:

1. Система повинна функціонувати на мобільному пристрої з ОС Андроїд;
2. Система повинна бути доступною для людей з вадами зору;
3. Система повинна бути простою у використанні;
4. Система повинна бути надійною та безпечною.

До нефункціональних критеріїв відповідності продукту належить:

- Атомарність;
- Доступність;
- Завершеність;
- Змінюваність;

- Однозначність.

Специфікацію нефункціональних вимог містить діаграма на рис. 2.6.

Specification Manager

Specification Manager:: Package: "Two Level Requirement Hierarchy - Copy" [<Any>]

Model Two Level Requirement Hierarchy - Copy Find Package

Item	Stereotype	Status	Difficulty	Priority
Checklist	Requireme...	Proposed		
☒ Не функціональні вимоги	Nonfunctio...	Proposed	Medium	Medium
☒ Система повинна бути доступна для людей з вадами зору	Implementa...	Proposed	Medium	Medium
☒ Система повинна бути надійна та безпечна	SecurityReq...	Proposed	Medium	Medium
☒ Система повинна бути проста у використанні	Regulatory...	Proposed	Medium	Medium
☒ Система повинна функціонувати на мобільному пристрої ОС Android	Architectur...	Proposed	Medium	Medium

Рисунок 2.6. – Діаграма специфікацій нефункціональних вимог продукту

2.2 Постановка задачі

Після аналізу і специфікації вимог до інформаційної системи для досягнення мети кваліфікаційного бакалаврського проекту необхідно поставити задачі проекту.

Метою кваліфікаційного бакалаврського проекту є практична реалізація проекту, що передбачає розробку мобільного додатку.

Додаток повинен бути націлений на людей, які мають вади зору та потребують допомоги у визначенні об'єктів навколишнього середовища, читання тексту, розпізнавання українських грошей.

Отже, функціонал додатку має дозволяти користувачеві досліджувати світ завдяки простому та ефективному функціоналу:

- розпізнавання предметів та об'єктів, які найчастіше зустрічаються у повсякденному житті;

- розпізнавати українські гривні;
- читати будь-який друкований текст українською мовою.

До задач проекту можна віднести: моделювання інформаційної системи, визначення основних вимог до функціональності, поведінки й інтерфейсів додатку, вибір цільової мобільної платформи та програмного забезпечення, проектування додатку і взаємодії його окремих модулів, а також, безпосередньо реалізація додатку та її результати.

Ключова задача – реалізація алгоритму розпізнавання образу українських національних купюр різного номіналу, фрагментів текстів української мовою

Постановка задачі: Мобільний застосунок повинен розпізнавати образи-команди за допомогою відеокамери. Задача повинна вирішуватись в режимі реального часу.

На вході у систему ми маємо зображення яке отримуємо за допомогою відеокамери мобільного телефону користувача. (табл. 2.1)

Таблиця 2.1

ПЕРЕЛІК І ОПИС ВХІДНИХ ПОВІДОМЛЕНЬ

№	Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
1	фото/зображення	image_id	файл	кожний запит	мобільний додаток

Після отримання вхідного повідомлення шляхом завантаження файлу у мобільний застосунок, формується спеціальний об'єкт-запит, що виконує функцію передачі даних від застосунку у сторонній програмний інтерфейс – Google Vision API.

API Google Cloud Vision використовує машинне навчання для ідентифікації зображень із попередньо навчених моделей на величезних наборах даних зображень. Він класифікує зображення на тисячі категорій, щоб підібрати об'єкти, місця та обличчя, і видає результати з високим вихідним рівнем достовірності значення.

API Google Cloud Vision використовується та інтегрується у мобільний застосунок з метою:

- Виявлення міток і об'єктів, що ідентифікує домінуючий об'єкт на зображенні. Дана властивість використана для створення метаданих у робочому каталозі зображень, що дозволяє здійснювати пошук на основі зображень
- Оптичне розпізнавання символів (OCR) для розуміння тексту в зображенні. Google Cloud Vision також може автоматично ідентифікувати широкий спектр різних мов.

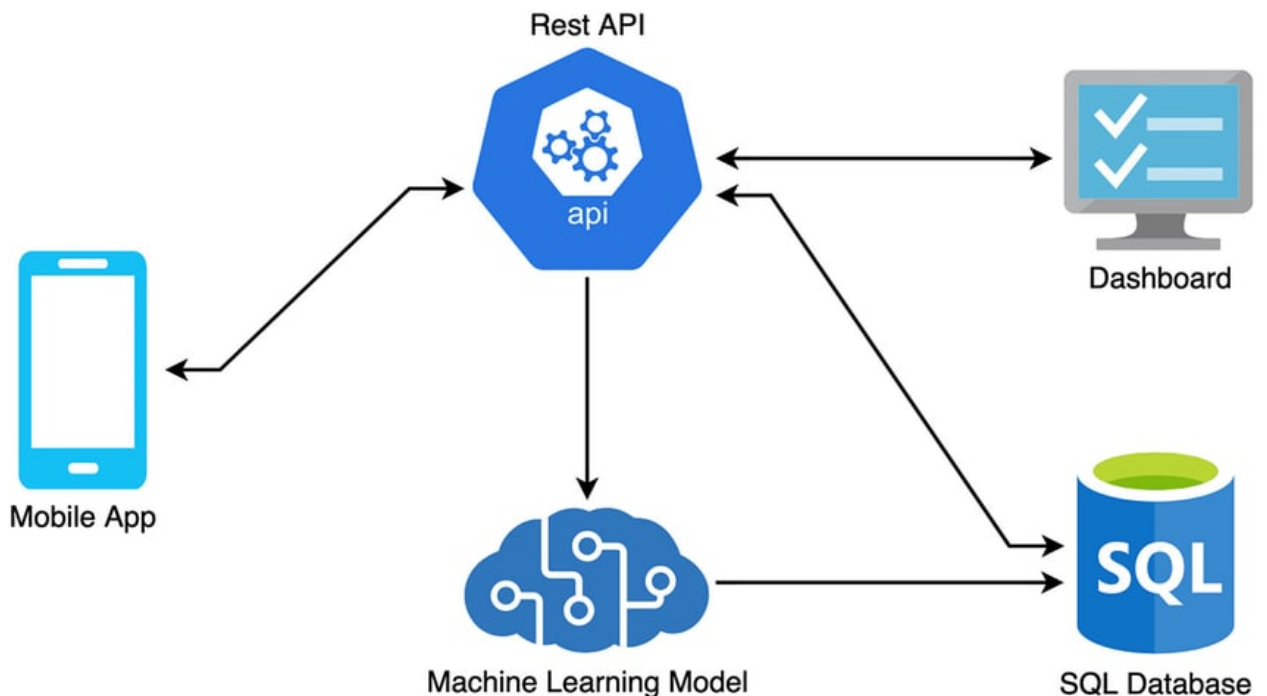


Рисунок 2.7. – REST API архітектура

Як видно з рис. 2.7, сам метод розпізнавання об'єкта виконується на серверній стороні, та не відноситься безпосереднього до розробки розглядаємого у даному кваліфікаційному бакалаврському проекті мобільного застосунку. Передача даних від мобільного застосунку до сервера та назад відбувається через використання REST API. Ключові методи обробки даних (для протоколу HTTP) представлені на рис. 2.8.

GET	/data	Gets all data
POST	/data	Add a new data to the db
PUT	/data	Update an existing data
DELETE	/data	Delete an existing data
users Everything about Users		
POST	/users	Register new user
PUT	/users	Update existing user
DELETE	/users	Delete user
POST	/users/login	Login

Рисунок 2.8. – Приклади REST-запитів

Розглянемо алгоритм формування REST-запитів:

Крок 1 – формується URL, за яким буде здійснюватися запит;

Крок 2 – прописуються всі заголовки (наприклад, Content-type, Authorization);

Крок 3 – виконується вибір HTTP-запиту (GET/POST/PUT/PATCH/DELETE та ін.);

Крок 4 – формується тіло запиту. Тип тіла запиту має бути сформований відповідно до значення заголовку Content-type;

Крок 5 – виконується запит;

Крок 6 – здійснюється обробка отриманих результатів.

Незважаючи на різноманітність параметрів формату повідомлень, більшість API REST використовують JSON (нотацію об'єктів JavaScript) у форматі повідомлень за замовчуванням. Вони використовують JSON, тому що він забезпечує легкий, простий і більш гнучкий формат повідомлень, який збільшує швидкість зв'язку. Облегчена природа JSON також дозволяє виконувати сценарії мобільної розробки та легкий для парсингу в Інтернеті за допомогою JavaScript.

У якості вихідних повідомлень, які ми отримуємо від стороннього програмного інтерфейсу Google, отримується спеціальний JSON-об'єкт, що

містить дані про ідентифікатор вхідного повідомлення та результат його обробки у вигляді тексту (описує, що саме вдалось розпізнати на фото). У випадку коли виникла помилка, то повертається об'єкт з кодом та текстом статусу помилки.

Опис вихідних повідомлень представлено у табл. 2.2.

Таблиця 2.2

ПЕРЕЛІК І ОПИС ВИХІДНИХ ПОВІДОМЛЕНЬ

№	Назва вихідного повідомлення	Ідентифікатор	Форма подання	Періодичність видання	Термін видання і допустимий час затримки	Користувачі інформації
1	JSON об'єкт	result_id	об'єкт	кожен запит	0.4 мс	користувач додатку
2	JSON об'єкт	error_id	об'єкт	у випадку помилки	1.2мс	—

На рис. 2.9 наведено інформаційну модель розв'язання задачі



Рисунок 2.9. – Інформаційна модель розв'язання задачі

Поставлені задачі забезпечать виконання основного функціоналу.

Спроектовані і реалізовані компоненти системи складатимуть повноцінний мобільний додаток.

2.3 Моделювання інформаційної системи

Моделювання мобільного додатку, розробка якого планується, є одним із найважливіших етапів створення якісного програмного продукту. Під час даного процесу розробляються проектні рішення щодо вибору платформи, де буде функціонувати майбутній програмний продукт, визначаються вимоги до майбутнього інтерфейсу продукту, з'ясовуються найбільш доцільні інструменти для досягнення поставленої мети.

2.3.1. Моделювання процесу роботи мобільного додатку

Комплексне визначення функціонального моделювання (IDEF0) – це метод моделювання функцій для бізнес-систем та інженерний підхід для аналізу потреб. Техніка IDEF0 використовує вікно для представлення функцій у процесі та показує супідрядність об'єктів. У IDEF0 розглядаються логічні стосунки між роботами, а не їх тимчасова послідовність (потік робіт) [10].

Можна виділити такі переваги побудови контекстної діаграми IDEF0:

- забезпечує вдосконалений системний аналіз;
- забезпечує вдосконалені методи спілкування;
- допомагає зрозуміти систему та її зв'язки.

Діаграму IDEF0 наведено на рис. 2.10. Вона демонструє процес роботи мобільного додатку. На вході ми отримуємо потребу користувача у розпізнаванні об'єкту (в якості об'єкта маємо на увазі текст, гроші, предмети), а на виході отримуємо опис об'єкта. Управлінням розробки мобільного додатку є нормативна документація, а механізмом в свою чергу: користувач, технічне і програмне забезпечення. Точку зору моделювання процесу роботи

мобільного додатку представляє розробник програмного забезпечення.

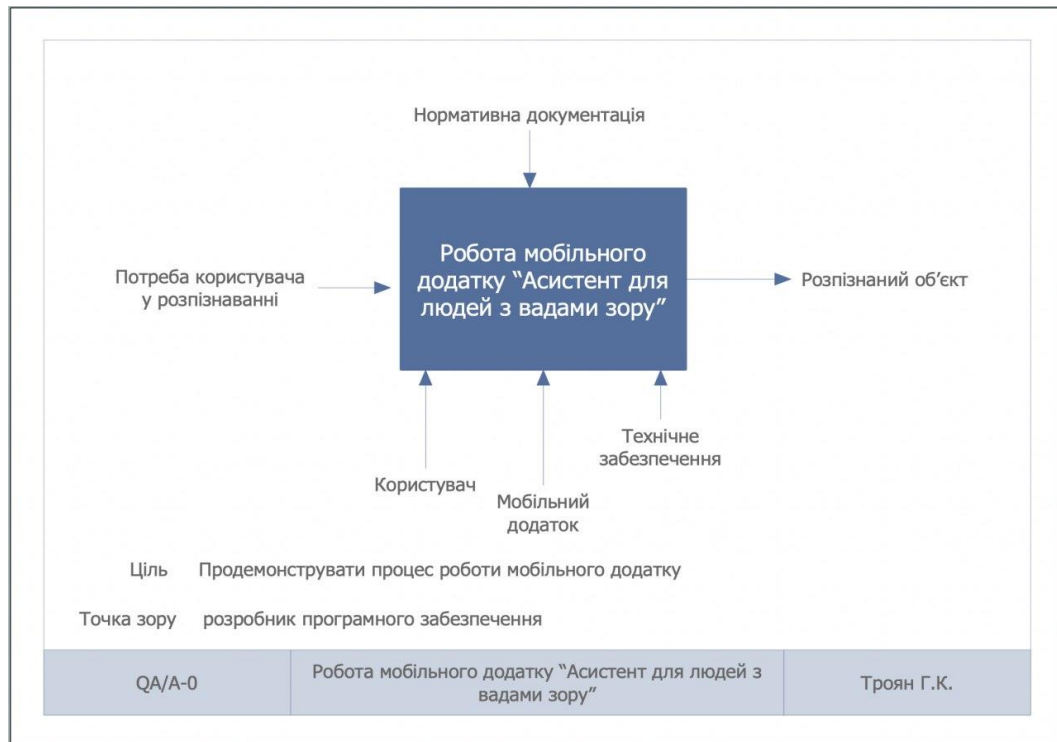


Рисунок 2.10 – Контекстна діаграма IDEF0 для роботи додатку

Процес роботи мобільного додатку поділяється на чотири основних блоки: вибір об'єкта для розпізнавання, фотографія об'єкта, розпізнавання об'єкта, отримання результату розпізнавання. Результатом блоку «вибір об'єкта для розпізнавання» є увімкнення камери телефону з відповідною кнопкою, блоку «фотографія об'єкта» – знімок камерою мобільного телефону, блоку «розпізнавання об'єкту» – передача фотографії на сервер для обробки. На виході процесу роботи додатку отримуємо текстове та звукове повідомлення про результати розпізнавання.

Також розроблено діаграми, які демонструють процес розробки мобільного додатку. На рис. 2.11 представлена діаграма IDEF0. На вході маємо потребу в програмному продукті, а на виході отримуємо готовий програмний продукт. Управліннями розробки мобільного додатку є технічна документація ОС Android, технологій створення мобільних додатків, а механізмами в свою чергу: розробник, замовник та технічне і програмне забезпечення. Точку зору моделювання процесу розробки мобільного додатку представляє розробник ПЗ.

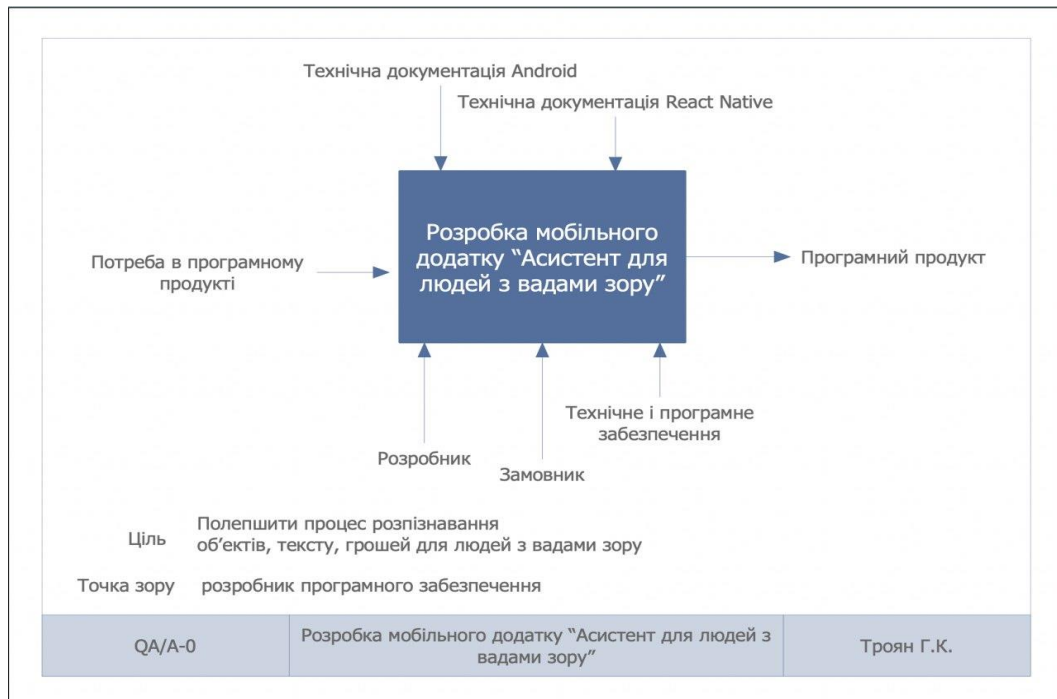


Рисунок 2.11 – Контекстна діаграма IDEF0 для розробки мобільного додатку

Декомпозиція першого рівня у нотації IDEF0 відображає взаємодію процесів розробки мобільного додатку (рис. 2.12).

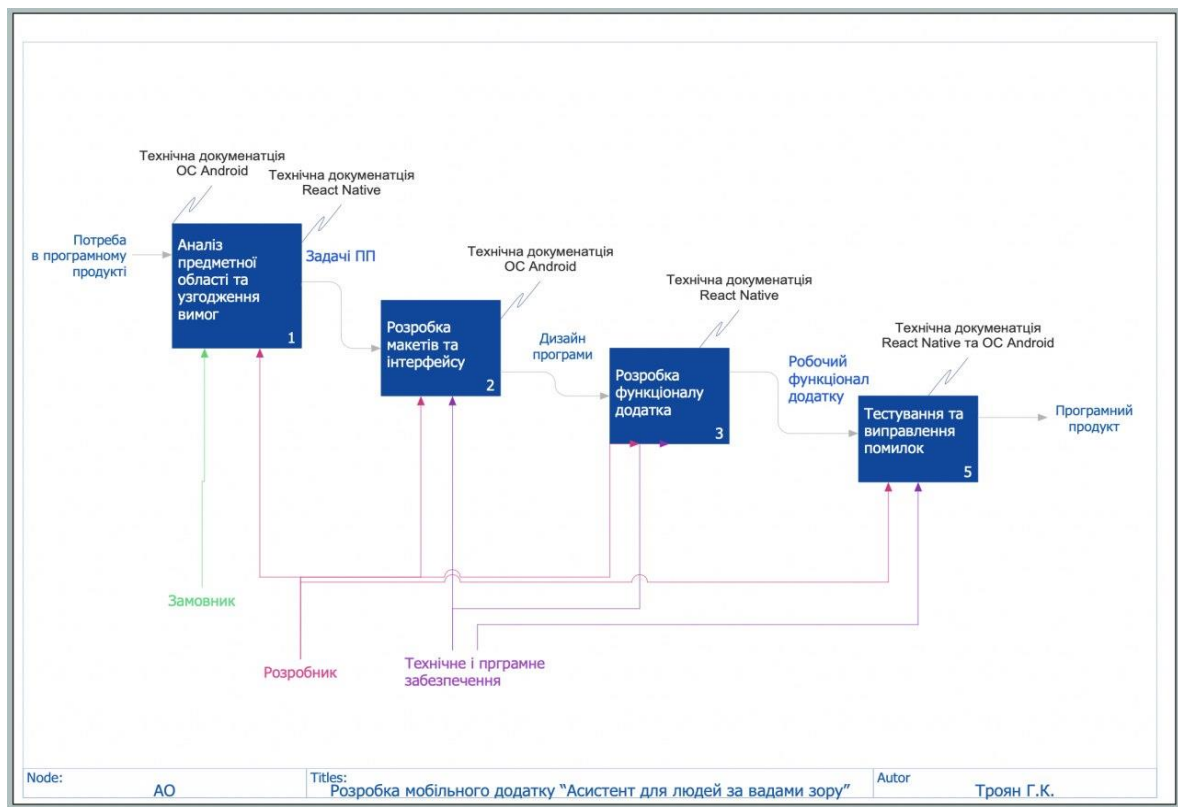


Рисунок 2.12 – Перший рівень декомпозиції IDEF0 розробки мобільного додатку

Процес розробки мобільного додатку поділяється на чотири основних блоки: аналіз предметної області та узгодження вимог, розробка макетів та інтерфейсу, розробка функціоналу додатку та тестування і виправлення помилок. Результатом блоку «аналіз предметної області і узгодження вимог» є задачі ПП, блоку «розробка макетів та інтерфейсу» – готовий дизайн програми, блоку «розробка функціоналу програми» – робочий функціонал додатку. На виході процес розробки отримуємо готовий програмний продукт.

2.3.2. Моделювання функціонування системи

Unified Modeling Language (UML) – уніфікована мова моделювання. Modeling має на увазі створення моделі, що описує об'єкт.

Unified (універсальний, єдиний) – підходить для широкого класу проєктованих програмних систем, різних галузей додатків, типів організацій, рівнів компетентності, розмірів проєктів. UML описує об'єкт в єдиному заданому синтаксисі, тому де б ви не намалювали діаграму, її правила будуть зрозумілі для всіх, хто знайомий з цією графічною мовою – навіть в іншій країні [25].

Одне із завдань UML – служити засобом комунікації всередині команди та спілкування із замовником.

Проектування. UML-діаграми допоможуть при моделюванні архітектури великих проєктів, в якій можна зібрати як великі, так і дрібніші деталі та намалювати каркас (схему) програми. По ньому згодом будуватиметься код.

Реверс-інжиніринг – створення UML-моделі з існуючого програмного коду, зворотна побудова. Може застосовуватися, наприклад, на проєктах підтримки, де є написаний код, але документація є неповною або відсутньою [25].

З моделей можна витягувати текстову інформацію і генерувати відносно тексти, що легко читаються, – документувати. Текст та графіка доповнюватимуть один одного.

Як і будь-яка інша мова, UML має власні правила оформлення моделей та синтаксис. За допомогою графічної нотації UML можна візуалізувати систему, поєднати всі компоненти в єдину структуру, уточнювати та покращувати модель у процесі роботи. На загальному рівні графічна нотація UML містить 4 основні типи елементів [12]:

- фігури;
- лінії;
- значки;
- написи.

UML-нотація є де-факто галузевим стандартом у галузі розробки програмного забезпечення, IT-інфраструктури та бізнес-систем.

Для того, щоб зрозуміти функціонування програмного продукту використовується модель варіантів використання (діаграма прецедентів), яка включає опис акторів, варіанти використання системи, взаємодію між ними у формі діаграми прецедентів.

Діаграма варіантів використання являє собою список дій, які зазвичай визначають взаємодію між роллю (відомої в уніфікованій мові моделювання як актор), і системи для досягнення мети [24]. Актором може бути людина чи інша зовнішня сутність. Діаграма варіантів використання може допомогти представити:

- сценарії, в яких система чи додаток взаємодіють з людьми, організаціями чи зовнішніми системами;
- цілі, які система чи додаток допомагає цим організаціям (відомим як актори);
- обсяг розроблюваної системи.

Можемо виділити два актори, які взаємодіють з додатком:

- користувач – людина, яка має змогу використовувати функціонал мобільного додатку;
- сторонній сервер – система, що надає функціонал обробки запитів, які здійснюються додатком.

Також провівши детальний аналіз було виділено наступні основні варіанти використання даного мобільного додатку:

– ВВ 1 Розпізнавання тексту – ВВ дозволяє користувачеві читати текст українською мовою з інтерфейсу додатку; ініціюється актором користувач; передумова: фотографія тексту створена; процес: користувач запускає мобільний додаток і натискає на кнопку “розпізнати текст”, потрапляє на екран камери та натискає кнопку, яка створює відповідну фотографію тексту, яка надсилається для обробки на сервер;

післяумова: розпізнаний текст відображається на екрані мобільного пристрою користувача та озвучується голосом;

– ВВ 2 Розпізнавання об’єкта – ВВ дозволяє користувачеві розпізнати предмети навколишнього середовища; ініціюється актором користувач; передумова: додаток запущений; процес: користувач у меню натискає кнопку “розпізнати об’єкт”. Відкривається камера мобільного телефону з відповідним функціоналом. Користувач наводить камеру на об’єкт та натискає одноіменну кнопку “розпізнати об’єкт”. Мобільний додаток фіксує об’єкт за допомогою фотографії, надсилає фотографію на сервер для аналізу. Післяумова: результат розпізнавання об’єкта відображається на екрані мобільного пристрою користувача та озвучується голосом;

– ВВ 3 Розпізнавання грошей – ВВ дозволяє користувачеві розпізнати номінал українських гривень; ініціюється актором користувач; передумова: додаток запущений; процес: користувач у меню додатку натискає на натискає на кнопку “розпізнати гроші” Відкривається камера мобільного телефону з відповідним функціоналом. Користувач наводить камеру на грошову купюру та натискає одноіменну кнопку “розпізнати гроші”. Мобільний додаток фіксує об’єкт за допомогою фотографії, надсилає фотографію на сервер для аналізу; післяумова: результат розпізнавання об’єкта відображається на екрані мобільного пристрою користувача та озвучується голосом.

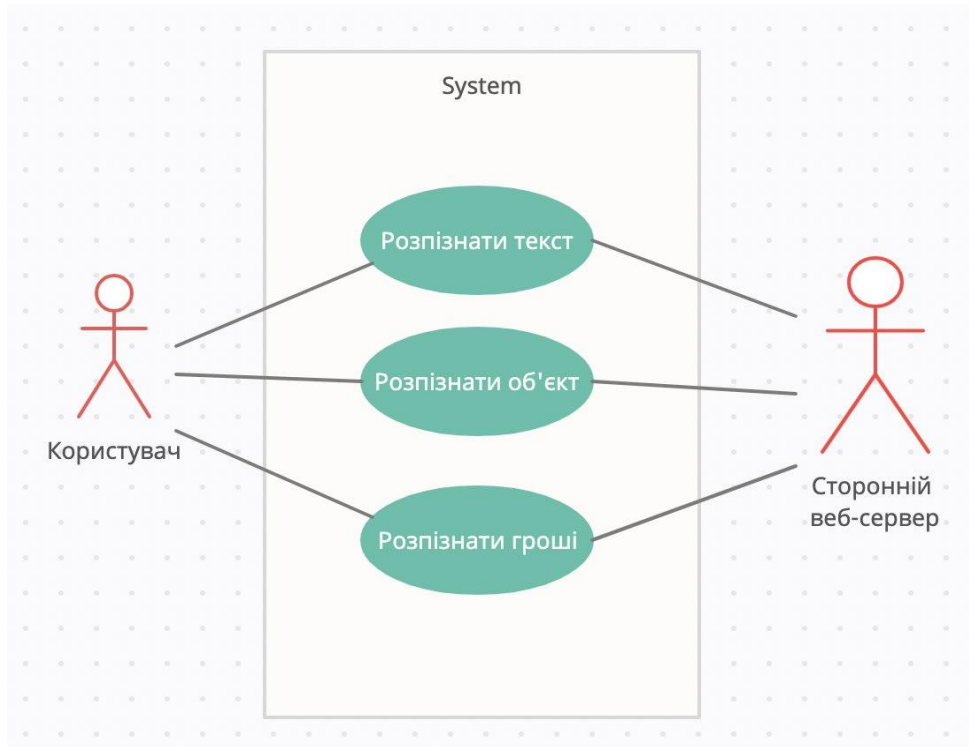


Рисунок 2.13 – Діаграма ВВ користувачем та стороннім сервером
(діаграма прецедентів)

2.3.3. Моделювання структури системи

Для коректного функціонування мобільного додатку і використання всіх можливостей програмного продукту необхідно розробити серверну частину додатку, яку можливо розмістити на хмарному комп'ютерному сервісі Digital Ocean. Розробка серверу здійснюватиметься із використанням технологій: Ruby, Ruby on Rails, TensorFlow та Onnx – для розробки математичної моделі розпізнавання зображень. На сервері буде налагоджено та прописано функціонал взаємодії зі сторонніми «арі» Google Vision, опрацювання запитів мобільного додатку та надсилання відповідей на запити. Також сервер відповідатиме за рендерінг веб-сторінки, яку можна створити для маркетингових цілей і опрацювання запитів зі сторінки. Функцію деплою (процесу викладки нової версії) необхідно налагодити через Cloud66 з метою підвищення рівня безпеки мобільного додатку та серверу. Cloud66 – це DevOps інструмент, який в свою чергу є службою для розгортання, керування та масштабування будь-якої програми в будь-якій хмарі або на серверах.

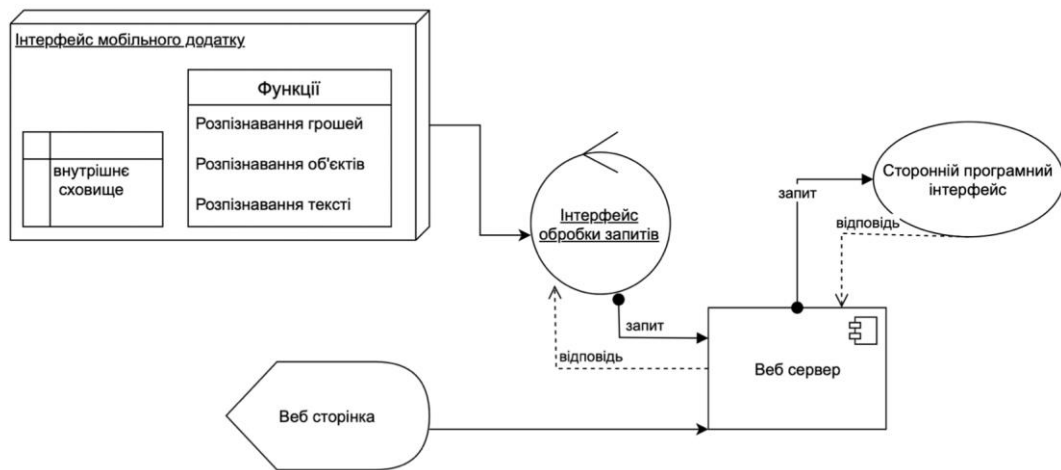


Рисунок 2.14 – Структура системи

2.3.4. Трасування вимог до системи

Трасування базується на специфікаціях усіх зв'язків між елементами вимог або обмежується описами функцій, ситуацій, контексту і можливих рішень.

Трасуванню піддаються:

- вимоги, що змінюються при їхньому формуванні;
- деякі деталі виконання функцій у робочому продукті системи, що не передбачалися, але з'явилися в зв'язку з практичною ситуацією, що виникла;
- зв'язки між різними моделями процесу проектування системи на ЖЦ і прийняті рішення про необхідність зміни вимог через виявлені недоліки в проміжному або робочому продукті;
- інформація про узгодження атрибутів вимог на різних рівнях даної схеми трасування та її матриць;
- системні вимоги, наприклад, до повторного використання готових компонентів;
- результати тестування, за якими можна визначити найбільш ймовірні частини коду, що вимагають зміни для виправлення виявлених дефектів.

На діаграмі зображено трасування вимог, рис. 2.15.

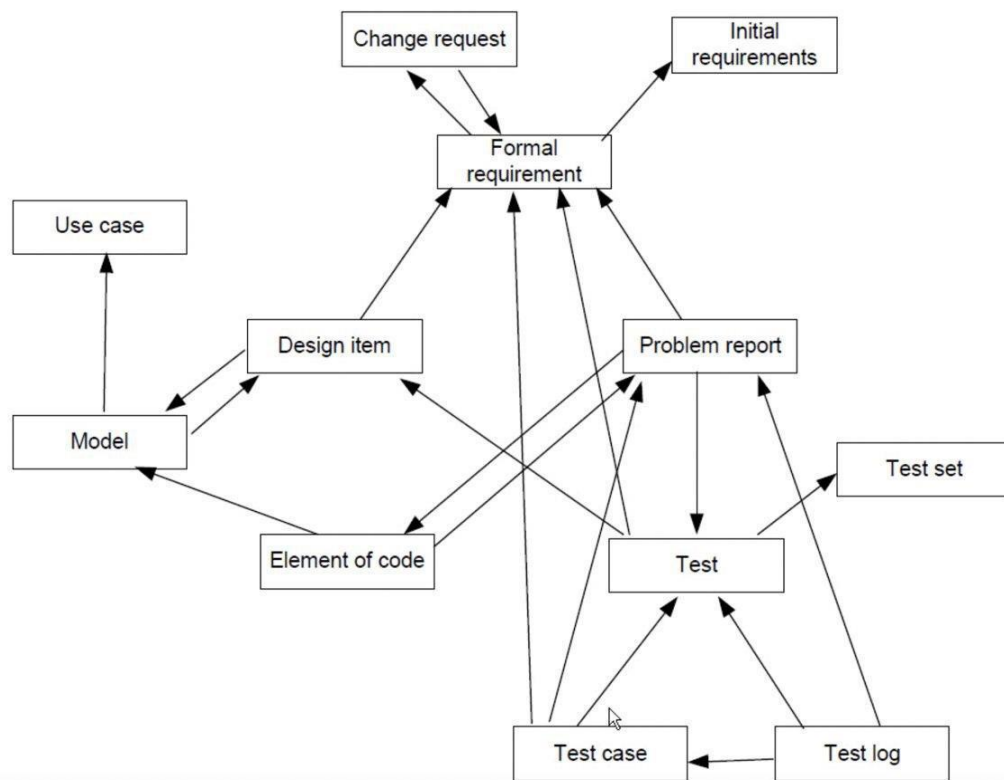


Рисунок 2.15 – Діаграма трасування

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Інформаційне забезпечення

Для подальшого розвитку мобільного додатку у маркетингових цілях необхідно мати можливість збору даних користувача, а саме контактних даних як телефон або пошта. Також ці дані можуть бути корисними у цілях створення аутентифікації користувача у майбутньому. Тож для цього необхідно зрозуміти який саме тип БД може задовольнити вимоги зберігання цих даних та саме спроектувати базу даних.

У світі технологій баз даних існує два основних напрямки: SQL та NoSQL, реляційні та нереляційні бази даних. Відмінності між ними у тому як вони спроектовані, які типи даних підтримують, як зберігають інформацію.

Реляційні БД зберігають структуровані дані, які зазвичай становлять об'єкти реального світу. Скажімо, це можуть бути відомості про людину, або вміст корзини для товарів у магазині, згруповані в таблицях, формат яких заданий на етапі проектування сховища.

Нереляційні БД улаштовані інакше. Наприклад, документоорієнтовані бази зберігають інформацію як ієрархічних структур даних. Йдеться про об'єкти з довільним набором атрибутів. Те, що у реляційної БД буде розбито кілька взаємозалежних таблиць, в нереляційної може зберігатися як цілісної сутності.

Внутрішній пристрій різних систем управління базами даних впливає особливості роботи з ними. Наприклад, нереляційні бази краще піддаються масштабуванню [31].

Нижче показано базу даних, що містить відомості про взаємини людей. Варіант а – це безсхемна структура, побудована у вигляді графа, характерна для NoSQL-рішень. Варіант б показує, як ті самі дані можна зобразити у структурованому вигляді, типовому для SQL.

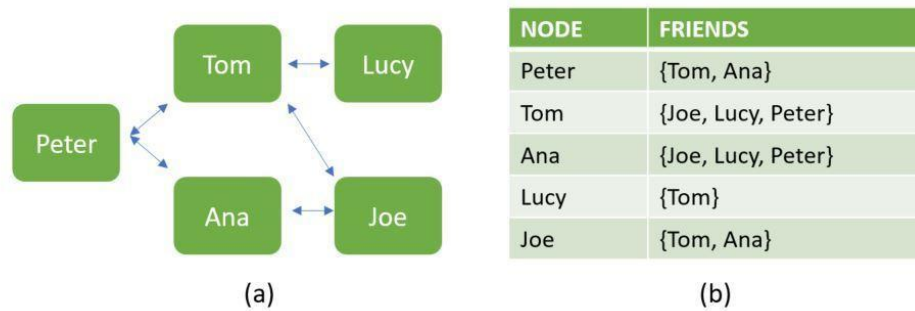


Рисунок 3.1 – Два варіанти представлення даних

Безсхемність означає, що два документи у структурі даних NoSQL не повинні мати однакові поля та можуть зберігати дані різних типів.

При реляційному підході дані треба зберігати в заздалегідь спроектованій структурі, з якої ці дані можна витягувати [31].

Виходячи з цього доцільним є використання саме реляційної бази даних, тому що необхідно зберігати структуровані дані, які є об'єктом відомостей про людину.

Типові представники реляційних баз даних SQL СУБД: MySQL/MariaDB, PostgreSQL, MSSQL та Oracle. Але потрібно розуміти що не всі з них ми можемо використовувати на мобільному пристрої.

Досліджуючи програмне забезпечення яке вже встановлене на пристроях на базі ОС Android можна з'ясувати що там використовується для усіх типів додатків база даних SQLite. SQLite – полегшена реляційна система керування базами даних. Втілена у вигляді бібліотеки, де реалізовано багато зі стандарту SQL-92 [33]. Сирцевий код SQLite поширюється як суспільне надбання (англ. public domain), тобто може використовуватися без обмежень та безоплатно з будь-якою метою [32].

На етапі проектування бази даних визначається назва таблиць, назва полей та тип даних який буде зберігатись:

Назва таблиці: Users (користувачі);

Назви полей: phone_number (номер телефону), email (електронна пошта), first_name (ім'я), last_name (призвіще);

Тип даних: string;

На рис. 3.2 нижче зображено як саме буде виглядати таблиця “Користувачі”.

users	
id	int
first_name	string
last_name	string
phone_number	string
email	string

Рисунок 3.2 – Таблиця бази даних “Користувачі”

Однією з особливостей використання встроєної бази даних мобільного пристрою полягає у тому що ці дані будуть зберігатись лише локально на пристрої користувача. Лише спроектувавши у подальшому “клієнт-сервер” архітектуру [35] з’явиться можливість отримання цих даних за допомогою мережевого спілкування та обміну інформацією у мережі Інтернет, засобами HTTP протоколу. HTTP (англ. HyperText Transfer Protocol - "протокол передачі гіпертексту") - протокол прикладного рівня передачі даних, спочатку - у вигляді гіпертекстових документів у форматі HTML, в даний час використовується для передачі довільних даних [34].

3.2. Технічне забезпечення

Протягом тривалого часу Google регулярно публікувала інформацію про популярність різних версій операційної системи Android, однак у якийсь момент пошуковий гігант перестав це робити. Uptodown [39] опублікував власні дані.

Серед користувачів Uptodown Android 11 займає перше місце з ринковою часткою 29,5%. Далі йде старіша Android 10 з результатом 25,2%.

Третє місце займає Android 9 Pie з результатом 11,5%, слідом йде Android 8.1 Oreo з 11%. Нова Android 12 в даний час встановлена всього на 2,6% користувальницьких пристроїв. Вона посідає лише п'яте місце.

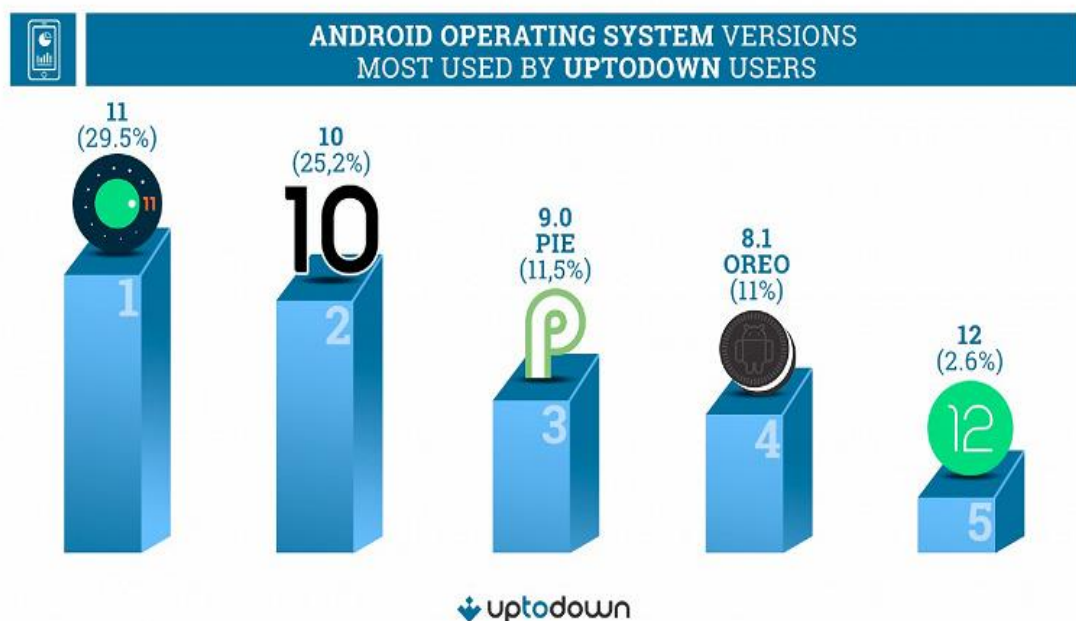


Рисунок 3.3 – Статистика популярності різних версій ОС Android

Для коректної роботи інформаційної системи з функціями розпізнавання пристрій повинен мати камеру та відповідати мінімальним вимогам постачальника програмного забезпечення операційної системи Android. Використовуючи статистику вирішено, що однією з вимог буде версії ОС не нижче 9ої.

Мінімальні вимоги до характеристик пристроїв, послуг операторів та сервісів для управління мобільною інфраструктурою з офіційного сайту Android ОС (Android 9):

- Оперативна пам'ять: 2ГБ;
- Сховище: 16 ГБ;
- Частота процесора: 1,1 ГГц;
- Архітектура: 32-бітна.

Для використання останньої версії Android 12 пристрій повинен мати:

- Оперативна пам'ять: 3ГБ;
- Сховище: 32 ГБ;
- Частота процесора: 1,4 ГГц;
- Архітектура: 64-бітна.

3.3. Програмне забезпечення

За останніми даними порталу StatCounter Global Stats на сьогоднішній день найпопулярнішими мобільними ОС у світі є Android, частка якої складає 69,74%, iOS з часткою 29,49% та інші ОС, частка кожної з яких не перевищує 0,45% [13]. Статистика збиралась з січня 2021 року по січень 2022 року. Кількість пристроїв з системою Android в Україні різнилась і складає 78,38%, а iOS складає 21,32% [14]. Більш докладні дані наведені на рис. 3.4.

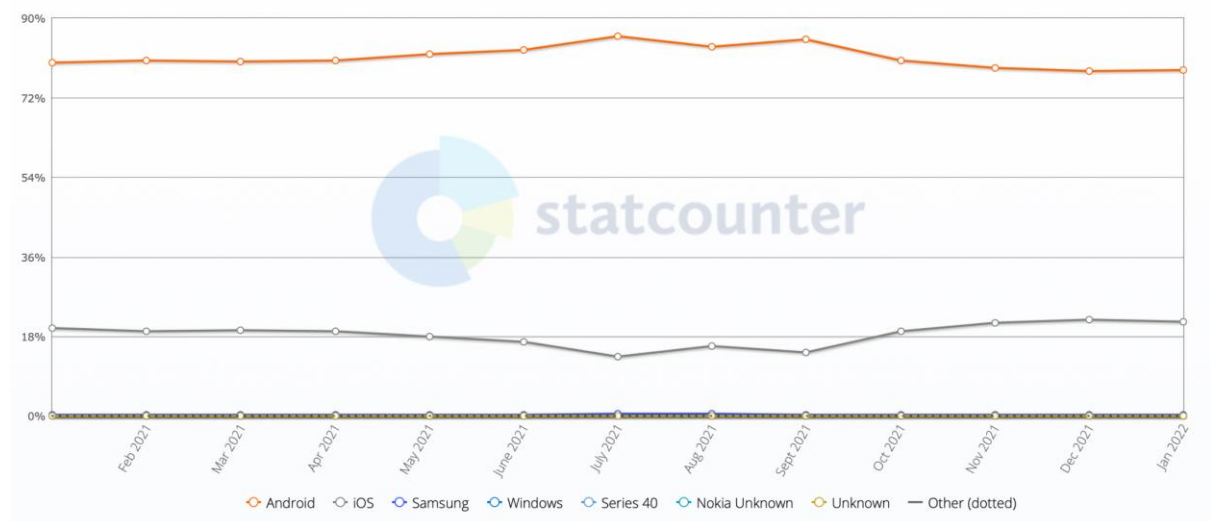


Рисунок 3.4 – Статистика мобільних операційних систем в Україні

Враховуючи вищенаведені статистичні показники можна дійти висновку, що розробка мобільних додатків під ОС Android наразі є актуальною.

Також була проаналізована статистика використання версій Android і найпопулярнішими версіями є версія 8.1 Oreo (16,66%) та версія 6.0 Marshmallow (11,37%). Найнижчою версією Android за останні роки серед найбільш використовуваних є версія 5.1, а отже і розробка буде проводитися починаючи з цієї версії [11]. Детальний рейтинг приведений на рис. 3.2.



Country		OS Version	Share
Ukraine	1	Android 8.1	16.66%
Ukraine	2	Android 6	11.37%
Ukraine	3	Android 7	9.18%
Ukraine	4	Android 7.1	9.18%
Ukraine	5	Android 9	9.05%
Ukraine	6	Android 5.1	8.43%
Ukraine	7	iOS 12.2	7.37%
Ukraine	8	Android 8	6.98%
Ukraine	9	iOS 12.3	4.05%
Ukraine	10	iOS 12.1	3.72%

Рисунок 3.5 – Рейтинг популярності ОС Android за матеріалами сайту DeviceAtlas

Так як для реалізації даного проекту була обрана ОС Android, розглянемо деякі доступні для розробки під цю платформу мови програмування. За останні роки можна виділити дві основні мови програмування: Java та Kotlin, однак для реалізації даного проекту ми розглянемо переваги таких мов програмування як JavaScript, Ruby.

Розробники описують JavaScript як «легку, інтерпретовану, об'єктно-орієнтовану мову з першокласними функціями». JavaScript найбільш відомий як мова сценаріїв для веб-сторінок, але використовується в багатьох

середовищах, які не є браузерами, таких як node.js або Apache CouchDB. Це заснована на прототипах, мультипарадигмальна мова сценаріїв, яка є динамічною та підтримує об'єктно-орієнтований, імперативний та функціональний стилі програмування. З іншого боку, Kotlin описується як «Статично типізована мова програмування, орієнтована на JVM та JavaScript». Kotlin – це статично типізована мова програмування для JVM, Android і браузера, яка на 100% взаємодіє з Java.

JavaScript і Kotlin належать до категорії «Мови» технологічного стека.

«Можна використовувати на фронтенді/бекенді» – це основна причина, чому понад 1556 розробників люблять JavaScript, а понад 28 розробників згадують «Сумісність із Java» як основну причину вибору Kotlin.

Kotlin – це інструмент з відкритим вихідним кодом з 28,3 тис. зірок GitHub і 3,28 тис. GitHub.

За даними спільноти StackShare, JavaScript має ширше схвалення, його згадують у 5080 стеках компаній і 6472 стеках розробників; порівняно з Kotlin, який входить до списку 268 стеків компаній і 208 стеків розробників.

Отже використання JavaScript є доцільним у зв'язку з тим, що це універсальна мова, якою можна написати як мобільний додаток так і веб-сторінку за необхідності. До того ж бібліотека React Native створена для розробки мобільних додатків розроблена саме на мові JavaScript.

Ruby – динамічна, рефлексивна, інтерпретована високорівнева мова програмування. Мова володіє незалежною від операційної системи реалізацією багатопоточності, сильною динамічною типізацією, збирачем сміття та багатьма іншими можливостями

З часу випуску публічної версії в 1995 році Ruby привернув увагу програмістів з усього світу. У 2006 році Ruby здобув масове визнання. У найбільших містах у всьому світі активно діють групи користувачів Ruby, а конференції, присвячені Ruby, заповнені до краю.

Ruby-Talk, основне розсилання для обговорення мови Ruby, містило в середньому 200 повідомлень щодня 2006 року. В останні роки ця кількість

зменшилася, тому що спільнота почала використовувати кілька тематичних розсилок замість однієї спільної.

Індекс ТЮВЕ, який вимірює зростання популярності мов програмування, поміщає Ruby у десятку загальновизнаних мов програмування. Багато в чому зростання популярності мови приписується популярності програмного забезпечення, написаного на Ruby, особливо – Ruby on Rails, фреймворку для розробки веб-додатків.

Ruby також є абсолютно відкритим. Відкритий для безкоштовного використання, змін, копіювання та розповсюдження.

Ruby on Rails – фреймворк, написаний мовою програмування Ruby, реалізує архітектурний шаблон Model-View-Controller для веб-застосунків, а також забезпечує їх інтеграцію з веб-сервером і сервером баз даних. Є відкритим програмним забезпеченням та розповсюджується під ліцензією MIT.

Отже, використання мови Ruby та фреймворку Ruby on Rails є доцільним для розробки веб-серверу за необхідності.

Стосовно вибору IDE для розробки, то було обрано Android Studio, так як це офіційний IDE для Android, програмний пакет побудований Google, у якому є все, що допоможе розробити надійний додаток для Android. Він відомий як третій найвідоміший IDE у світі та здатний прискорити процес розробки, не приносячи шкоди якості. Деякі функції, які роблять Android Studio найкращим інтегрованим середовищем розробки, – це його гнучка система побудови, аналізатор у режимі реального часу, інтуїтивно зрозумілий макет та редактор смарт-коду [18].

Також обрано IDE RubyMine – це інтегроване середовище розробки, яке дозволяє розробникам ефективно працювати з проектами Ruby/Rails – від написання та налагодження коду до тестування та розгортання фінального додатку. Цей програмний продукт створено компанією JetBrains. Саме ця компанія розробила мову програмування Kotlin та у травні 2017 року компанія Google повідомила, що включає підтримку Kotlin у Android Studio 3.0 – офіційний інструмент розробки для ОС Android.

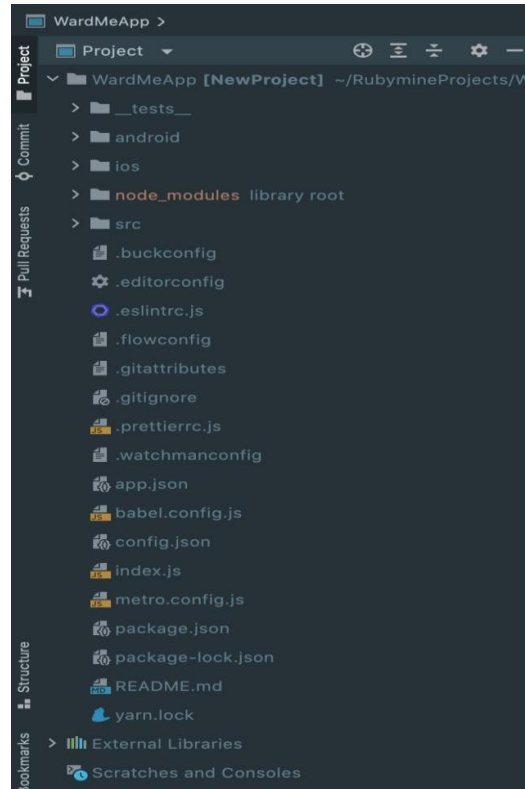


Рисунок 3.7 – Структура проекту

У цьому проекті використовувалися як стандартні бібліотеки так і сторонні бібліотеки, завантажені з веб-ресурсів. Деякі бібліотеки були завантажені з веб сайту GitHub [21], для прикладу деякі з них:

- Vision Camera [22] (бібліотека для використання функцій камери).
- React Navigation [23] (бібліотека для реалізації навігації між екранами).
- React Native Splash Screen [24] (призначена для створення “Splash Screen” (екран-заставка).

Дані бібліотеки використовуються в розробленому мобільному додатку і допомагають у реалізації всіх функціональних можливостей.

Перейдемо до розробки класів мобільного додатку та їх призначення.

Основні розроблені класи та їх призначення приведено в табл. 3.1.

Таблиця 3.1 – Розроблені класи та їх призначення

Ім'я класу	Призначення класу
CameraScreen	призначений для роботи з камерою (створення фотографії, відправка запиту на веб-сервер, отримання відповіді з веб-серверу, відображення відповіді користувачеві)

Таблиця 3.2 – Функції класу CameraScreen та їх призначення

Ім'я функції	Призначення функції
callApi()	формує запит та відправляє на веб-сервер
takePicture()	створює фотографію, перетворює її у формат base64 та викликає функцію callApi
renderTextBlock()	отримує значення та рендерить його відповідно до місця виклику
isEmpty()	перевірка об'єкту на наявність елементів
renderCamera()	метод який відповідає за рендер всього екрану та назначає виконувати функції у контексті екрану

Далі опишемо інші основні функції та їх призначення, які використовувались у різних класах. Функції представлені в табл. 3.3.

Таблиця 3.3 – Основні функції та їх призначення

Ім'я функції	Призначення функції
MainMenu()	рендерить головне меню
App()	рендерить додаток, має налаштування навігації
NavigationContainer()	контейнер для створення навігації
openCamera()	вмикає екран камери
openSettings()	вмикає екран налаштувань
SettingsScreen()	рендерить екран налаштувань

3.4. Результати реалізації інформаційної системи

Результати реалізації інформаційної системи розглянути нижче та деталізовані у Додатках А-Б.

3.4.1. Прототипування інтерфейсу інформаційної системи

Згідно визначення, прототипування – це створення макета, моделі майбутнього мобільного додатка для того, щоб визначити правильність його структури, його функціональності та, в цілому, концепції програми. Макет має чудову властивість усувати непорозуміння між різними фахівцями (керівник, дизайнер, програміст, клієнт), залученими до проекту, структурувати думки і запобігати помилкам і виконання зайвої роботи ще на ранніх стадіях розробки. Можна тестувати майбутній додаток, використовуючи фокус-групу, це допоможе отримати корисну інформацію від майбутніх користувачів [19].

В зв’язку з тим, що цільовою аудиторією даного додатку є люди з вадами зору, при прототипуванні було важливо використовувати певну гаму кольорів, тому перш за все розроблено відповідну палітру. Палітра представлена на рис. 3.8.



Рисунок 3.8 – Кольорові схеми за контрастом

Прототипування інтерфейсу додатку здійснювалось у Figma – онлайн-сервісі для розробки інтерфейсів та прототипування з можливістю організації спільної роботи у режимі реального часу.

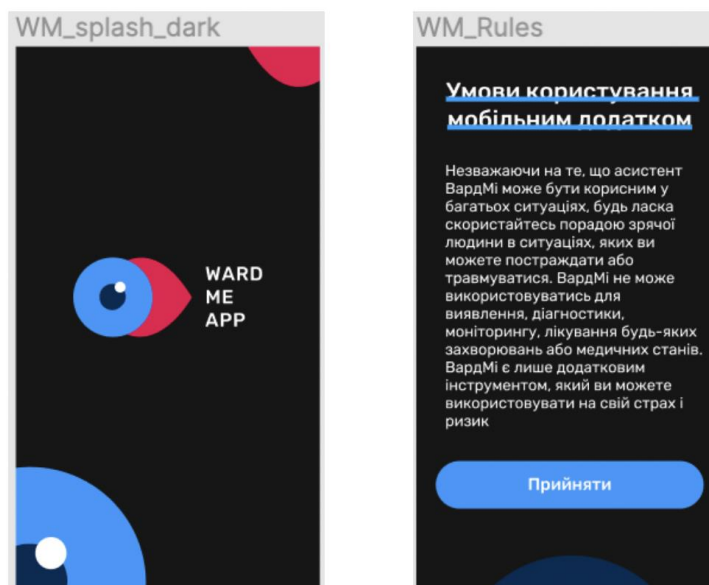


Рисунок 3.9 – Макет сплеш вікна, вікна умов користувача (зліва направо)

На даному рисунку показано сплеш екран, який користувач бачить при відкритті мобільного додатка, вікно умов користувача, які він має прийняти, щоб безперешкодно користуватись додатком.

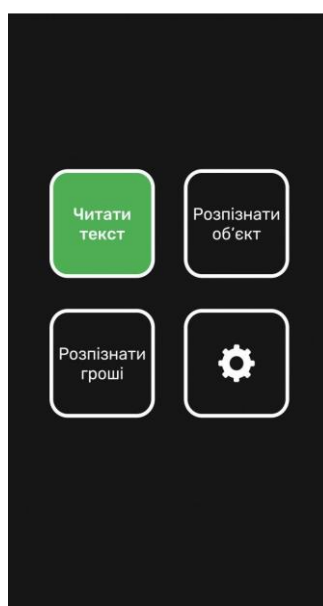


Рисунок 3.10 – Макет екрану головного меню додатку

На рис. 3.10 міститься екран головного меню activity із вибором категорії об'єкту, який може розпізнати додаток та налаштуваннями:

- читати текст;
- розпізнати об'єкт;
- розпізнати гроші;
- налаштування (шестерня).

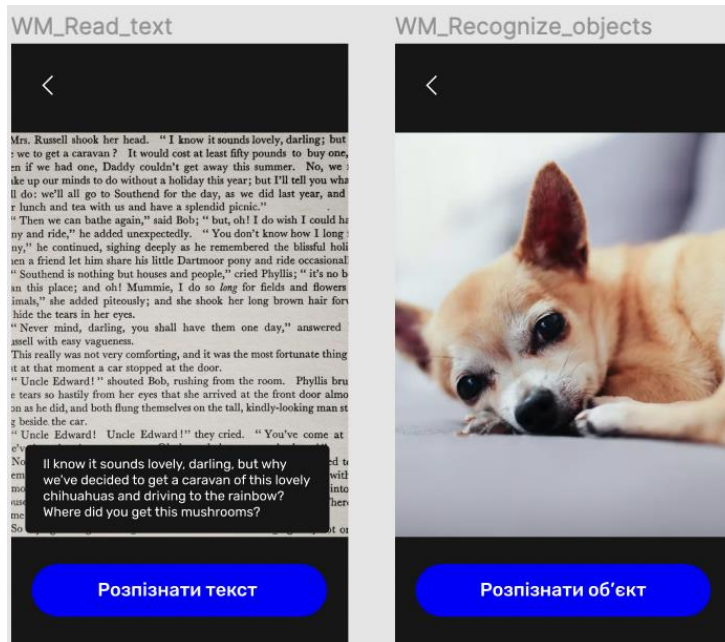


Рисунок 3.11 – Вікно “розпізнати текст” та вікно “розпізнати об’єкт” (зліва направо)

На рис. 3.11, який представлений вище показано activity з функціоналу “розпізнати текст” та “розпізнати об’єкт”. Кнопки екрану “розпізнати текст”:

1. Кнопка “розпізнати текст”.
2. Кнопка повернення до головного меню в лівій верхній частині екрану.

Кнопки екрану “розпізнати об’єкт”:

1. Кнопка “розпізнати об’єкт”.
2. Кнопка повернення до головного меню в лівій верхній частині екрану.

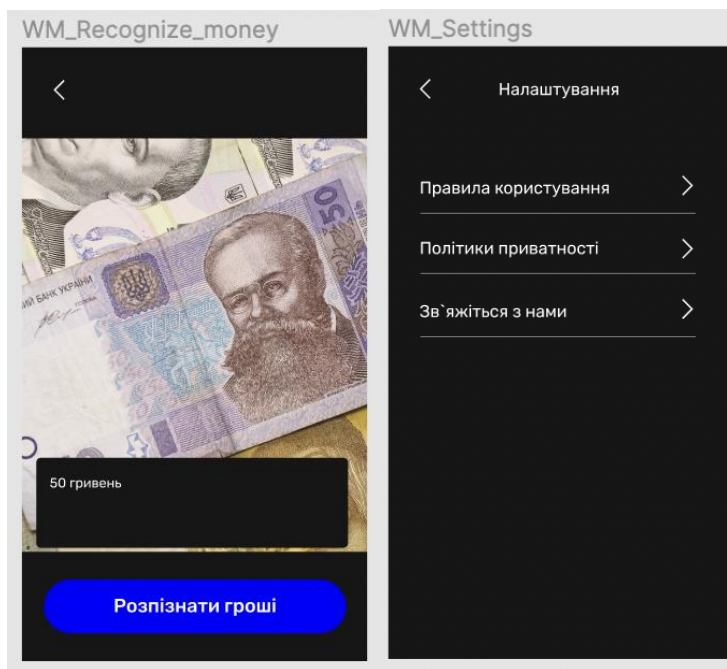


Рисунок 3.12 – Вікно “розпізнати гроші” та вікно “налаштувань” (зліва направо)

На рис. 3.12, який представлений вище показано activity з функціоналу “розпізнати гроші” та “налаштування”. Кнопки екрану “розпізнати гроші”:

1. Кнопка “розпізнати гроші”.
2. Кнопка повернення до головного меню в лівій верхній частині екрану.

Кнопки екрану “налаштування”:

1. Кнопка “правила користування”;
2. Кнопка “політики приватності”;
3. Кнопка “зв’яжіться з нами”.

Для мобільного застосунку було створено логотип, рис. 3.13. Даний логотип відповідає наступним вимогам, які передбачені Google Play:

- підсумковий розмір: 512 x 512 пікселів;
- формат: 32-розрядний PNG;
- кольорова палітра: sRGB;
- максимальний розмір файлу: 1024 КБ;
- форма: повний квадрат;
- тінь: без тіні. [18]

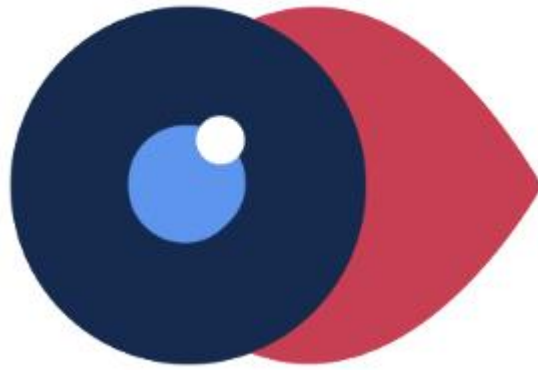


Рисунок 3.13 – Логотип мобільного додатку

За основу логотипу було обрано мінімалістичний стиль з чотирьма основними кольорами: білий, голубий, темносиній та малиновий. Логотип для мобільного додатку був розроблений за допомогою Adobe Illustrator – векторний графічний редактор, розроблений і поширюваний фірмою Adobe Systems [20]. Спочатку було створено ескіз на папері, потім було поетапно відтворено логотип мобільного додатку.

3.4.2. Результати реалізації мобільного застосунку

Результатом реалізації інформаційної системи є мобільний додаток з функціями розпізнавання тексту, об'єктів та грошей.

Після завантаження мобільного додатку на свій телефон користувач проходить наступні кроки:

1. Сплеш екран (стартовий екран);
2. Користувач має прийняти умови користування мобільним додатком з наступним текстом: “Незважаючи на те, що асистент ВардМі може бути корисним у багатьох ситуаціях, будь ласка, скористайтесь порадою зрячої людини у ситуаціях, яких ви можете постраждати або травмуватися ВардМі не може використовуватись для виявлення, діагностики, моніторингу, лікування будь-яких захворювань або медичних станів. ВардМі є лише додатковим інструментом, який ви

можете використовувати беручи відповідальність за результат на себе.

Кнопка “Прийняти”;

- Після того як користувач натиснув кнопку “Прийняти”, відкривається головний екран мобільного додатку з чотирма кнопками: “Розпізнати текст”, “Розпізнати об’єкт”, “Розпізнати гроші”.

Головний екран містить чотири кнопки що активують основних функціонал.

Для розуміння наведено діаграму ієрархії екранів рис. 3.14



Рисунок 3.14 – Діаграма ієрархії екранів

Детальний опис функції “Розпізнавання тексту”, екран “Читати текст”.

Якщо це перша функція, яку користувач використовує, він має надати дозвіл мобільному додатку використовувати камеру мобільного телефону.

1. Якщо користувач бажає розпізнати текст, він натискає на кнопку “Читати текст”. Телефон робить знімок тексту, який розпізнається мобільним додатком та озвучується функцією “Talk Back”. Якщо користувач хоче зробити ще один знімок, він так само натискає кнопку “Читати текст”.

2. Далі користувачу необхідно застосувати інший функціонал додатку, для чого він натискає кнопку “назад” у верхньому лівому куті, яка повертає його на екран з основним функціоналом.

Детальний опис функції “Розпізнавання об’єктів”, екран “Розпізнати об’єкт”.

Якщо це перша функція, яку він використовує, він має надати дозвіл мобільному додатку використовувати камеру мобільного телефону.

1. Користувач бажає розпізнати об’єкт, що знаходиться поруч із ним, для чого він натискає кнопку “Розпізнати об’єкт”, після натискання кнопки вмикається камера мобільного телефону, яку користувач наводить на якийсь об’єкт.

2. Після наведення камери на об’єкт, користувач натискає кнопку “розпізнати об’єкт”, телефон робить знімок і описує, що він сфотографував.

3. Якщо користувачу необхідно розпізнати ще один об’єкт, він повторно натискає кнопку “розпізнати об’єкт” у нижній частині екрану, телефон робить знімок і описує, що він сфотографував.

4. Далі користувачу необхідно застосувати інший функціонал додатку, для чого він натискає кнопку “назад”, яка повертає його на екран з основним функціоналом.

Детальний опис функції “Розпізнавання грошей”, екран “Розпізнати гроші”.

Якщо це перша функція, яку він використовує, він має надати дозвіл мобільному додатку використовувати камеру мобільного телефону.

1. Користувач хоче розпізнати гроші (українські гривні), для чого він натискає кнопку “Розпізнати гроші”, відкривається камера мобільного телефону, яку користувач може навести на об’єкт (купюру).
2. Після наведення камери на об’єкт, користувач натискає кнопку “Розпізнати гроші”, телефон робить знімок об’єкта і озвучує номінал купюри (1-2-5-10 грн).
3. Якщо користувачу необхідно розпізнати ще один об’єкт, він повторно натискає кнопку “розпізнати об’єкт” у нижній частині екрану, телефон робить знімок і описує, що він сфотографував.
4. Далі користувачу необхідно застосувати інший функціонал додатку, для чого він натискає кнопку “назад”, яка повертає його на екран з основним функціоналом.

Екран “Налаштування” або “інформація”.

1. Даний функціонал містить налаштування та є також інформативним.
2. Після натискання на кнопку зі знаком “Шестерня” відкривається екран з наступним наповненням:
 - Зв’яжіться з нами
 - Правила користування (terms of use)
 - Політики приватності (privacy policy)

Детальний опис інформаційного меню “Налаштування”

1. Користувач бажає зв’язатися зі службою підтримки, для чого він обрав пункт “зв’яжіться з нами”. При натисканні автоматично одразу, при натисканні на кнопку данного меню, відкривається вікно для написання і відправки електронного повідомлення на поштову адресу, в якому користувач може написати текстове повідомлення і направити його на вказану поштову скриньку.
2. Умови користування (правила) – при натисканні відкривається сторінка веб сайту з умовами користування.
3. Політики приватності – при натисканні відкривається сторінка веб сайту з політиками.

Під час розробки мобільного додатку були виконані поставлені у проекті задачі та реалізовано відповідний функціонал та спроектований інтерфейс. В подальшому можливо доповнювати як саму структуру додатку іншими або оновленими вікнами, так і додатковими функціями для більшої зручності користувачів.

ВИСНОВКИ

Під час виконання завдань кваліфікаційного бакалаврського проекту було вирішено актуальну проблему – розроблення інформаційної системи (ІС) з функціями розпізнавання об'єктів, грошей, тексту для людей з вадами зору для можливості її подальшого впровадження. ІС забезпечить безбар'єрний доступ людей з вадами зору до дослідження навколишнього світу та полегшить визначення ними об'єктів щоденного вжитку. У ході роботи проведено моделювання предметної області для вирішення поставлених задач та огляд літератури, змодельовано верхній та нижній рівні ІС.

У ході роботи проведено аналіз сучасного стану можливостей операційної системи (ОС) Android для вирішення поставленої задачі та проводиться огляд літератури, здійснено огляд існуючих мобільних додатків, а також проведено їх порівняння та виявлення недоліків, які використано під час розробки вимог та моделювання мобільного додатку. Зокрема, визначено бізнес-вимоги, функціональні вимоги, нефункціональні вимоги, їх специфікації та пріоритетність, які відображені у відповідних діаграмах.

Визначивши мету та задачі системи, було сформовано вимоги до системи та здійснено її проектування за допомогою уніфікованої мови моделювання – UML. Після цього змодельовано поведінку системи, її структуру та користувацький інтерфейс.

Закінчення етапу розробки вимог і моделювання системи послідовно наблизило до її прототипування та реалізації за допомогою програмного забезпечення доцільного для виконання поставлених задач. Розроблено макети дизайну додатку та логотипу.

Після прототипування інтерфейсу здійснено практичну реалізацію мобільного додатку за допомогою обраного програмного забезпечення у відповідності з визначеними вимогами, макетом дизайну.

За час розробки інформаційної системи у великому обсязі використовувалися матеріали та інструменти необхідні для моделювання та

прототипування, що допомогло закріпленню напрацьованих навичок і нових умінь у даних областях знань.

З результатів реалізації інформаційної системи вбачається, що вона є лаконічною, зрозумілою і простою у користуванні. Для вдосконалення та коректної роботи системи необхідно проводити відповідні тестування, після яких можливо доповнювати структуру мобільного додатку (ІС) як іншими або оновленими вікнами, так і додатковими функціями для більшої зручності користувачів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Операційна система Android: веб-сайт.
<https://uk.wikipedia.org/wiki/Android>
2. Публікація додатку Google Play Console: веб-сайт.
<https://support.google.com/googleplay/android-developer/answer/9859751?hl=ru> (дата звернення 23.12.2021).
3. Be My Eyes. Google Play: веб-сайт.
<https://play.google.com/store/apps/details?id=com.bemyeyes.bemyeyes&hl=ru&gl=US> (дата звернення: 05.01.2022).
4. Envision AI. Google Play: веб-сайт.
<https://play.google.com/store/apps/details?id=com.letsenvision.envisionai&hl=ru&gl=US> (дата звернення: 05.01.2022).
5. Sullivan+. Google Play: веб-сайт.
<https://play.google.com/store/apps/details?id=tuat.kr.sullivan&hl=ru&gl=US> (дата звернення: 05.01.2022).
6. Aipoly. Google Play: веб-сайт.
https://play.google.com/store/apps/details?id=com.aipoly.vision&hl=en_US (дата звернення: 28.12.2021).
7. Вигерс Карл Разработка требований к программному обеспечению/Пер, с англ. – М.: Издательство-торговый дом «Русская Редакция», 2004. – 576 с.
8. Конспект лекцій з дисципліни «Аналіз вимог до програмного забезпечення» для студентів напрямку підготовки 6.050103 «Програмна інженерія». Укладач: канд. фіз.-мат. наук, доцент Божуха Л.М.– Дніпродзержинськ, ДДТУ: 2015. – 94 с.
9. Опорний конспект з курсу “Аналіз вимог до програмного забезпечення” для студентів напрямку підготовки “Програмна інженерія” / О.Л. Козак – Тернопіль, 2011. – 56 с.

10. Методологія IDEF0: веб-сайт.
https://stud.com.ua/87184/ekonomika/metodologiya_idef0
11. Mobile OS versions by country DiviceAtlas: веб-сайт. URL:
<https://deviceatlas.com/blog/mobile-os-versions-by-country#ua> (дата звернення: 10.01.2022).
12. UML Use Case Diagram LucidChart: веб-сайт.
<https://www.lucidchart.com/pages/videos/uml-use-case-diagram-tutorial>
(дата звернення: 03.01.2022).
13. Mobile Operating System Market Share Worldwide Jan 2021 – Jan 2022
StatCounter Global Stats: веб-сайт. URL: <https://gs.statcounter.com/os-market-share/mobile/worldwide> (дата звкернення: 10.01.2022)
14. Mobile Operating System Market Share Ukraine Jan 2021 – Jan 2022
StatCounter Global Stats: веб-сайт. URL: <https://gs.statcounter.com/os-market-share/mobile/ukraine/#monthly-202101-202201> (дата звернення: 10.01.2022).
15. Вимоги до програмного забезпечення: веб-сайт. URL:
https://uk.wikipedia.org/wiki/Вимоги_до_програмного_забезпечення#cite_note-1 (дата звернення: 30.01.2022).
16. Нефункціональні вимоги: веб-сайт. URL:
https://uk.wikipedia.org/wiki/Нефункціональні_вимоги (дата звернення: 19.02.2022).
17. Специфікація вимог до програмного забезпечення: веб-сайт. URL:
https://uk.wikipedia.org/wiki/Специфікація_вимог_до_програмного_забезпечення (дата звернення: 19.02.2022).
18. Специфікації дизайну іконок для Google Play: веб-сайт. URL:
<https://developer.android.com/distribute/google-play/resources/icon-design-specifications> (дата звернення: 03.01.2022).
19. Test Case Software testing fundamentals:
<http://softwaretestingfundamentals.com/test-case/> (дата звернення: 21.01.2022).

20. Adobe Illustrator Wikipedia: веб-сайт. URL: https://en.wikipedia.org/wiki/Adobe_Illustrator#cite_ref-_6feaca787ce222de_1-0 (дата звернення: 20.12.2021).
21. Github: веб-сайт. URL: <https://github.com/>
22. Vision Camera: веб-сайт. URL: <https://github.com/mrousavy/react-native-vision-camera>
23. React Native Splash Screen: веб-сайт. URL: <https://reactnavigation.org>
24. Mobile application testing Wikipedia: веб-сайт. URL: https://en.wikipedia.org/wiki/Mobile_application_testing (дата звернення: 05.12.2021).
25. Ларман Крэг. Применение UML 2.0 и шаблонов проектирования, 3-е издание. – М.: Вильямс, 2013. с. 209-222.
26. Akkara JD, Kuriakose A. Smartphone apps for visually impaired persons (Online). Kerala J Ophthalmol 2019;31:242-8 – URL: <https://www.kjophthal.com/article.asp?issn=0976-6677;year=2019;volume=31;issue=3;spage=242;epage=248;aulast=Akkara>
27. Азін В. О., Байда Л. Ю., Грибальський Я. В., Красюкова-Еннс О. В. Доступність та універсальний дизайн : навч.-метод. посіб./ за заг. ред. Байди Л. Ю., КрасюковоїЕннс О. В. — К., 2013. —128с.
28. Носенко Ю. Електронна інклюзія як ефективна стратегія забезпечення доступності та відкритості освіти / Ю. Носенко // Педагогічні інновації: ідеї, реалії, перспективи. – 2016. – Вип. 2. – С. 116-123. – Режим доступу: http://nbuv.gov.ua/UJRN/ped_in_2016_2_22.
29. Створення безбар'єрного середовища та соціальна інклюзія: світовий досвід для України : аналіт. доп. / [Зубченко С. О., Каплан Ю. Б., Тищенко Ю. А.]. – Київ : НІСД, 2020. – 24с/
30. Carole Martinez. 13 Must-Have Apps for Blind or Visually Impaired People in 2022 (Online). – URL: <https://www.inclusivecitymaker.com/apps-blind-visually-impaired-people/>
31. SQL та NoSQL. URL: <https://habr.com/ru/company/ruvds/blog/324936/>
32. SQLite. URL: <https://uk.wikipedia.org/wiki/SQLite>

- 33.Information Technology -Database Language SQL. URL:
<http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt>
- 34.HTTP. URL: <https://uk.wikipedia.org/wiki/HTTP>
- 35.Клієнт-серверна архітектура. URL: https://uk.wikipedia.org/wiki/Клієнт-серверна_архітектура
- 36.Статистика людей з вадами зору УНІАН. URL:
<https://www.unian.ua/health/country/76863-sogodni-mijnarodniy-den-nezryachih-lyudey.html>
- 37.Сліпота. URL: <https://uk.wikipedia.org/wiki/Сліпота>
- 38.Object recognition. URL: <https://www.mathworks.com/solutions/image-video-processing/object-recognition.html>
- 39.Uptown. URL: <https://en.uptodown.com/android>

ДОДАТКИ

Додаток А

Фрагменти програмного коду

1. Функція виклику стороннього програмного інтерфейсу.

```
callApi = async (base64, action) => {
  const formData = new FormData();
  formData.append('image', base64);
  formData.append('features', action);

  try {
    let response = await fetch(config.server.host + config.server.route, {
      method: 'POST',
      headers: {
        "Accept": "application/json"
      },
      body: formData
    });

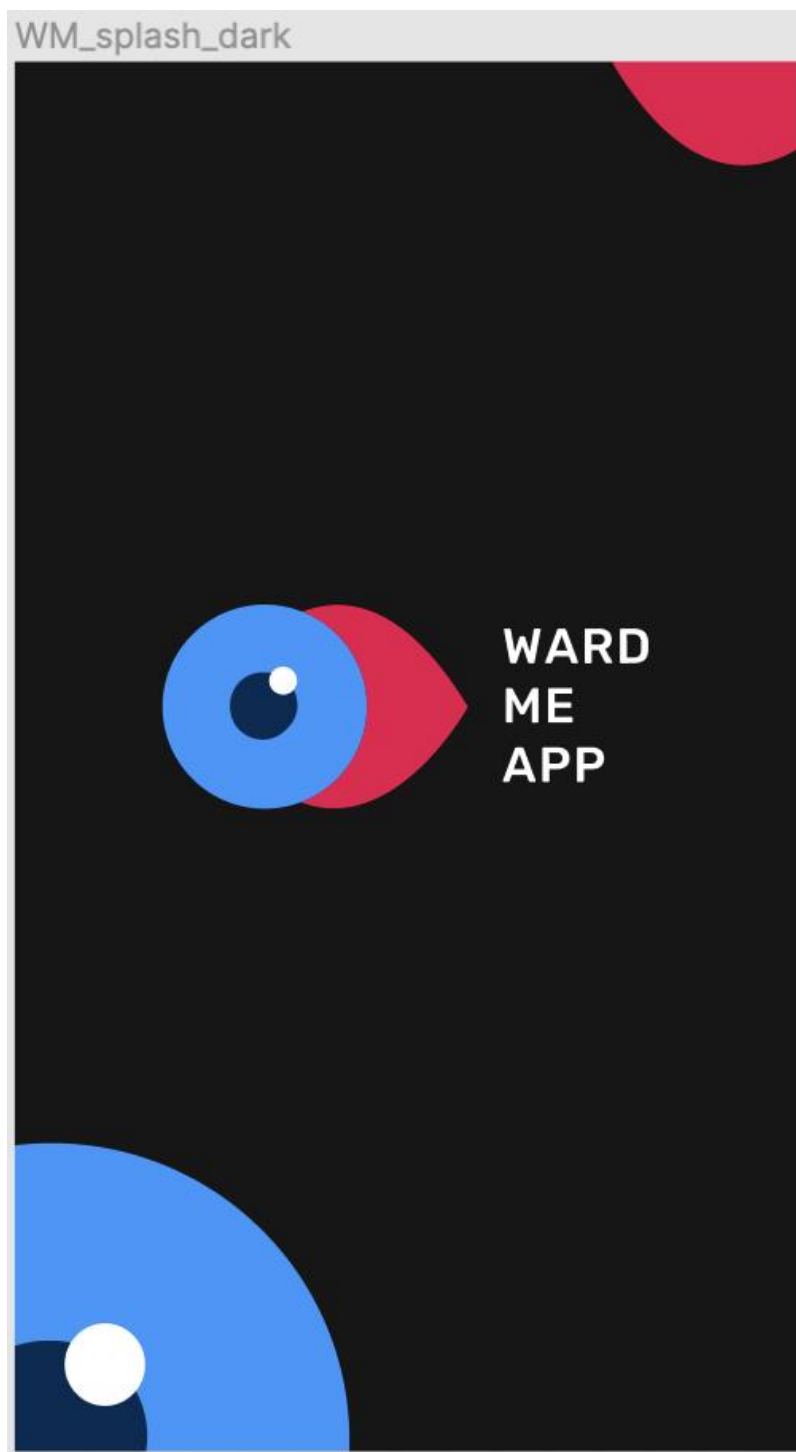
    await response.text().then(res => {
      if (response.ok) {
        this.setState(prevState => ({
          canDetectText: !prevState[this.state.canDetectText],
          loading: false,
          textDetection: res,
        }))
      } else {
        this.setState( {
          loading: false
        })
        alert("Помилка, спробуйте ще");
      }
    })
  } catch (e) {
    console.log(e.message)
  }
}
```

2. Функція відкриття екрану камери.

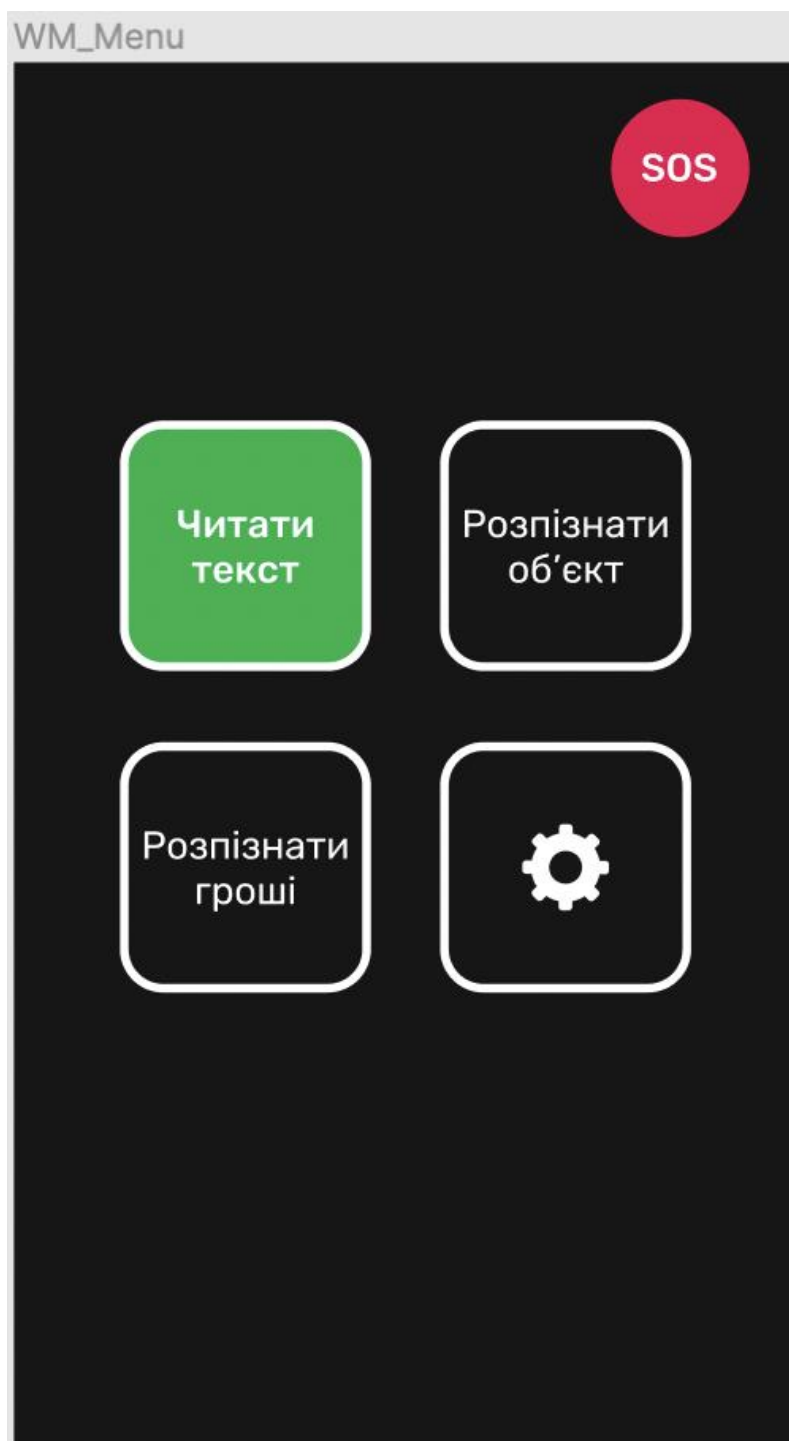
```
const openCamera = (props) => useCallback(() => {  
  Vibration.vibrate();  
  
  requestCameraPermission().then((res) =>{  
    if (res) {  
      navigation.navigate('CameraScreen', props);  
    } else {  
      Alert.alert('Немає доступу',  
        'Цей функціонал заблокований, поки ви не надасте дозвіл використання камери.',  
        [  
          {  
            text: "Дати доступ",  
            onPress: () => Linking.openSettings(),  
            style: "ok",  
          },  
          {  
            text: "Скасувати",  
            style: "cancel",  
          },  
        ],  
      )  
    }  
  })  
}, [])
```

Прототип екранів мобільного додатку

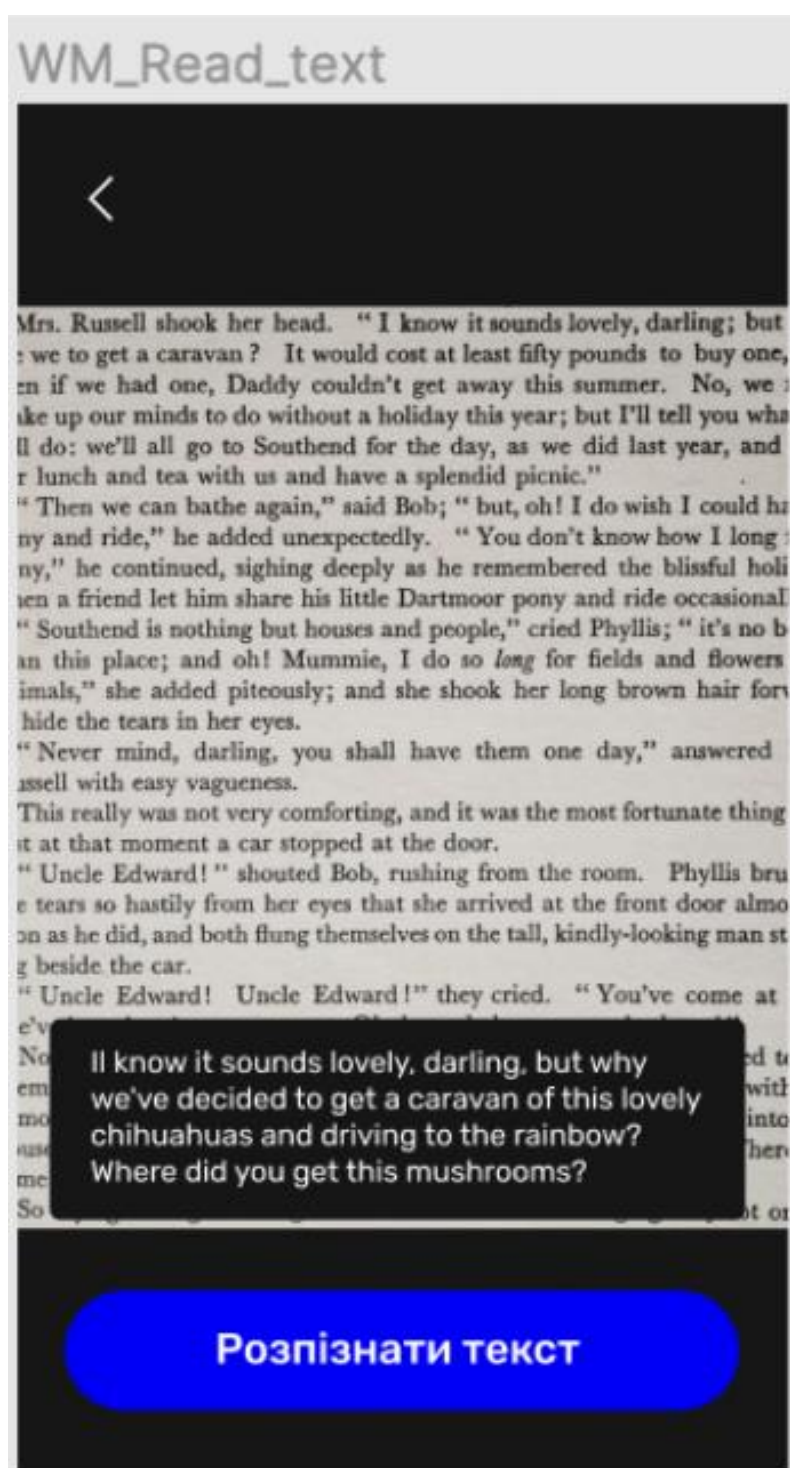
1. Сплеш екран (splash screen)



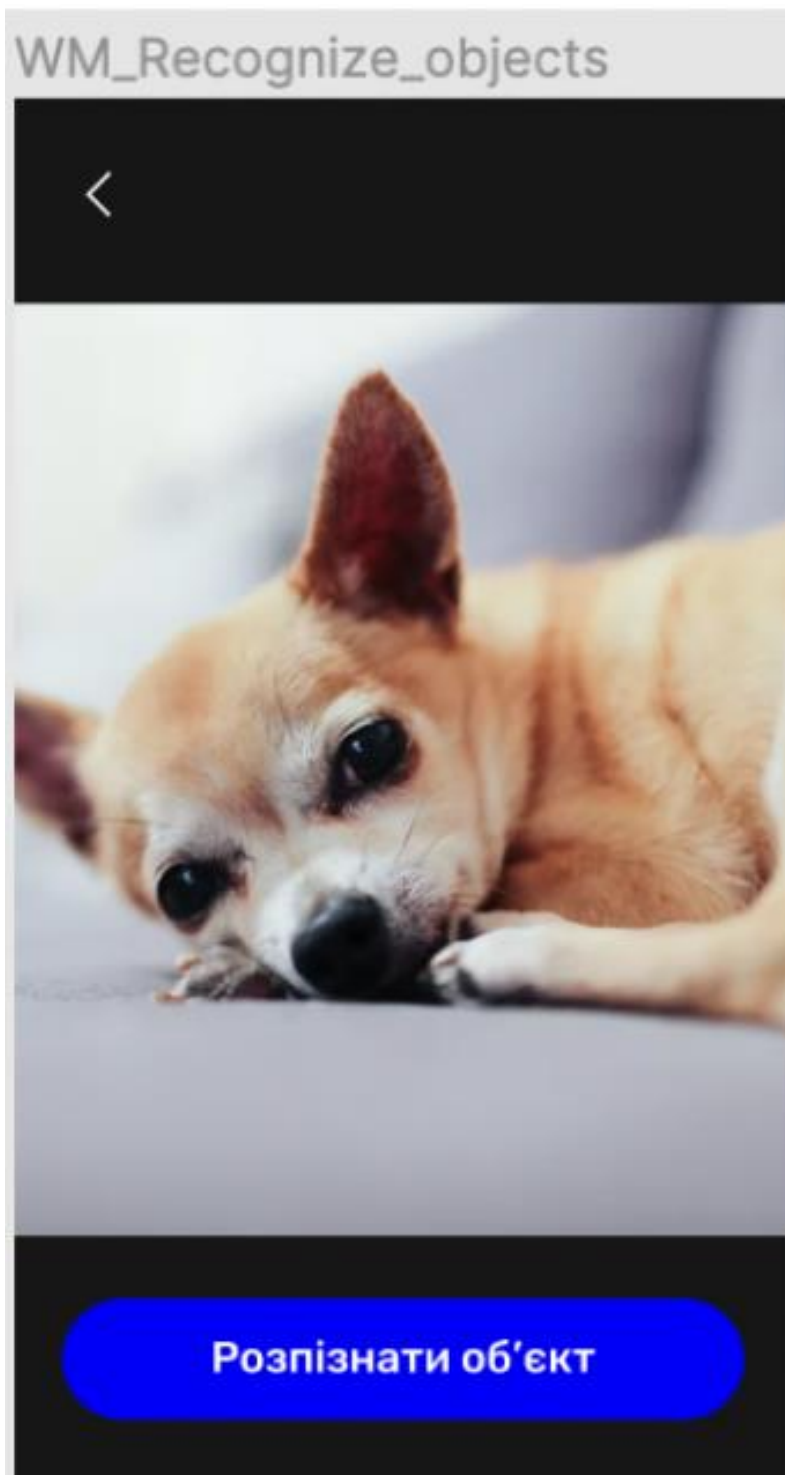
2. Екран “Меню”, головний екран



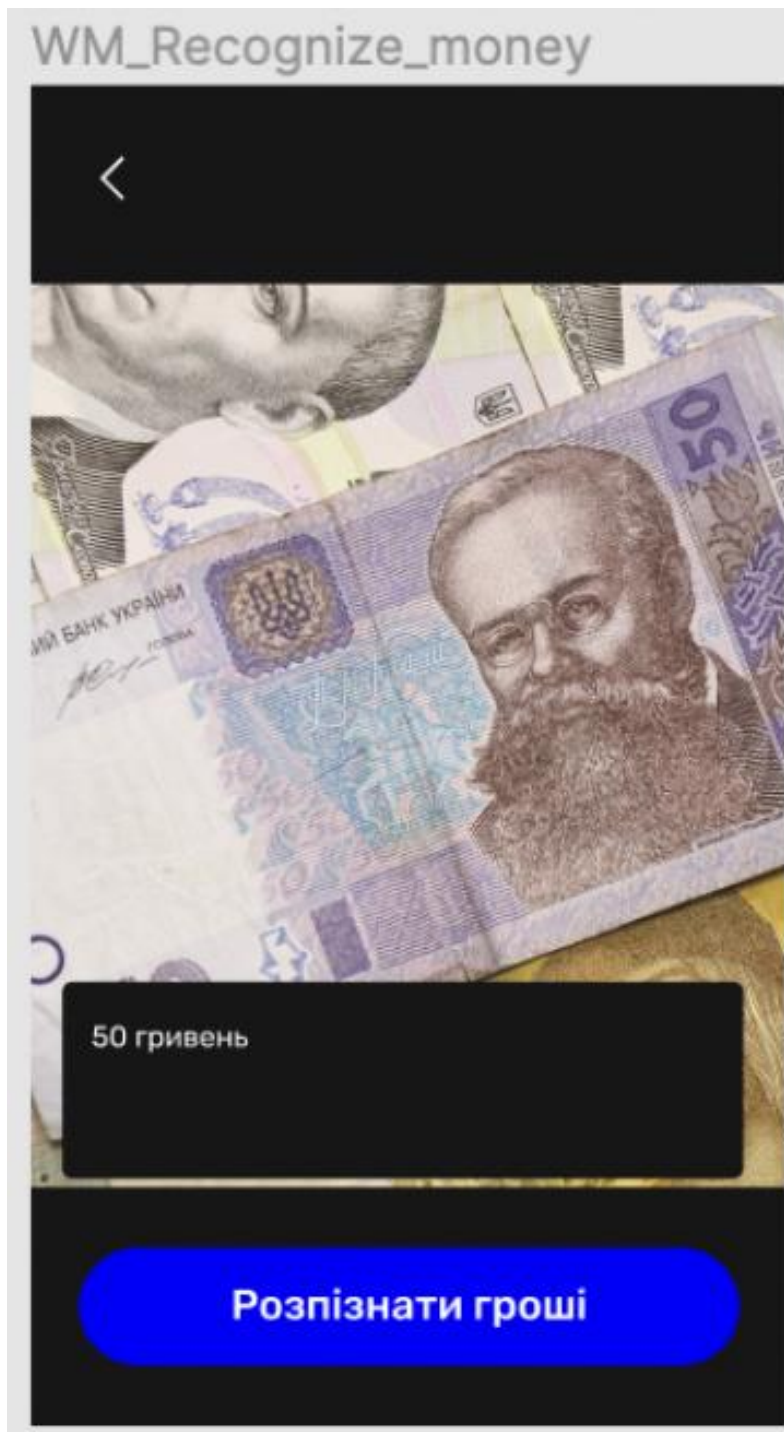
3. Екран “Розпізнати текст”



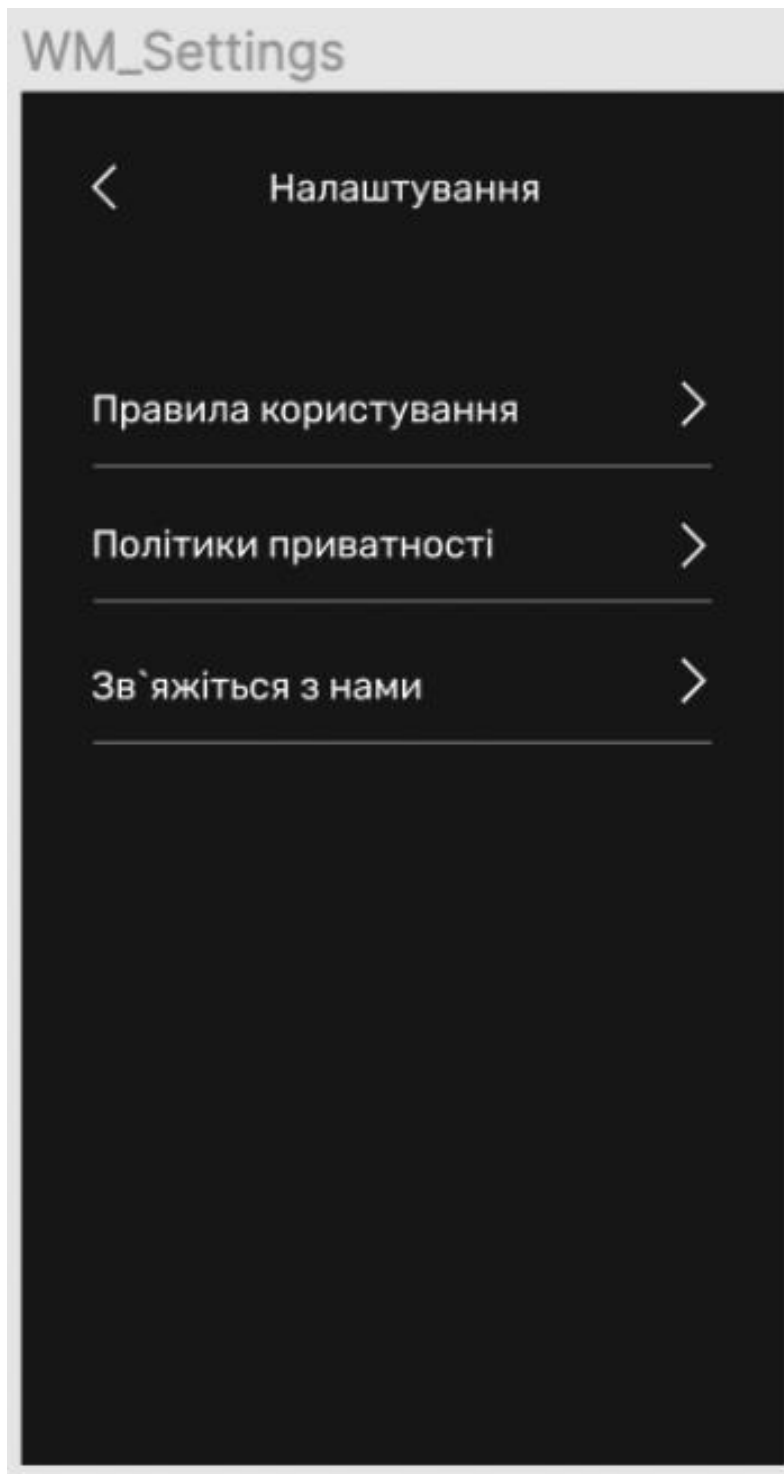
4. Екран “Розпізнати об’єкт”



5. Екран “Розпізнати гроші”



6. Екран “Налаштування”



Підтвердження факту апробації результатів досліджень

Збірник праць учасників Всеукраїнської науково-практичної конференції
«Інформаційні технології та моделювання систем»

Троян Г. К., студентка
Київський національний економічний університет
імені Вадима Гетьмана»
Науковий керівник – к. е. н, доцент Зінов'єва І. С.

**МОБІЛЬНИЙ ЗАСТОСУНОК РОЗПІЗНАВАННЯ ТЕКСТУ, ОБ'ЄКТІВ
НАВКОЛИШНЬОГО СЕРЕДОВИЩА ТА ГРОШЕЙ ДЛЯ ЛЮДЕЙ С ВАДАМИ ЗОРУ**

Сучасні інформаційні технології вирішують багато задач та проблем у повсякденному житті людей, серед іншого – виступають у ролі допоміжних інструментів – асистентів. На цей час ведуться багато досліджень у галузі ІТ задля підвищення доступності та усунення бар'єрів для людей з вадами зору, в тому числі, проводиться розробка відповідних мобільних додатків. Одним зі способів розв'язання окресленої проблеми є розробка мобільного застосунку «асистента», який допоможе людям із вадами зору розпізнавати предмети, які її оточують, зокрема, номінал грошей, читати та розпізнати текст.

Як свідчить практика, люди з вадами зору часто неохоче користуються спеціалізованими (медицини) пристосуваннями для покращення слабкого зору, зокрема: окуляри, ручні пристрої збільшення зображень тощо. Серед багатьох причин виділяються фактори соціальної стигматизації та заперечення [1]. Тому, в сучасному світі на поруч людям із вадами зору може прийти смартфон, вміння користуватися яким може істотно розширити можливості таких людей.

На даному етапі розвитку ринку програмних рішень для осіб із вадами зору існує значна кількість різноманітних програм, інформаційних систем та мобільних додатків як з відкритою / публічною ліцензією, так і на платній основі. Вони працюють під різними ОС, як то iOS, Android, Windows та ін. Розпізнавання предметів здійснюється за допомогою камери мобільного телефону, шляхом фотографування або відеозапису навколишнього середовища користувача. У спеціалізованій літературі виділяють такі типи програмних продуктів для людей з вадами зору: програми для голосового управління; ігри; ідентифікатори та зчитувачі на основі камери; навігатори; електронні лупи; зчитувачі електронних книг та ін. [1].

На українському ринку програмного забезпечення, на жаль, немає доступних програмних продуктів, що б враховували національні особливості та були б направлені на покращення соціалізації людей з вадами зору. Більшість застосунків представлені розробками закордонних продуктових компаній, вони погано розуміють та розпізнають українську мову, не завжди «розуміють» номінал української валюти тощо.

Розробка мобільного застосунку – асистента для людей з вадами зору, який отримав назву «WardMe App», відбувалась згідно з розробленими макетами і створеними діаграмами та функціональними вимогами. В якості середовища розробки було обрано Node.js та Java Development Kit для компіляції. У проєкті використано як стандартні бібліотеки, так і сторонні бібліотеки, завантажені з веб-ресурсів. Деякі бібліотеки були завантажені з веб-ресурсу GitHub [2], для прикладу деякі з них: Vision Camera [3] (бібліотека для використання функцій камери); React Navigation [4] (бібліотека для реалізації навігації між екранами); React Native Splash Screen [5] (призначена для створення «Splash Screen» (екран-заставка).

Прототипування інтерфейсу додатку здійснювалось у Figma – онлайн-сервісі для розробки інтерфейсів та прототипування з можливістю організації спільної роботи у режимі реального часу.

Результатом робіт з проєктування став мобільний додаток з функціями розпізнавання тексту, об'єктів та грошей. Після завантаження мобільного додатку на свій телефон користувач проходить наступні кроки:

1. Сплеш екран (стартовий екран);
2. Користувач має прийняти умови користування мобільним додатком з наступним текстом: «Незважаючи на те, що асистент WardMe може бути корисним у багатьох ситуаціях, будь ласка, скористайтесь порадою зрячої людини у ситуаціях, яких ви можете постраждати

або травмуватися WardMe не може використовуватись для виявлення, діагностики, моніторингу, лікування будь-яких захворювань або медичних станів. WardMe є лише додатковим інструментом, який ви можете використовувати беручи відповідальність за результат на себе. Кнопка “Прийняти”;

3. Після того як користувач натиснув кнопку “Прийняти”, відкривається головний екран мобільного додатку з чотирма кнопками: “Читати текст”, “Розпізнати об’єкт”, “Розпізнати гроші”, “Налаштування”.

На рис. 1 представлено дизайн макету головного екрану застосунку із вибором категорії об’єкту, який може бути розпізнаним та налаштуваннями.

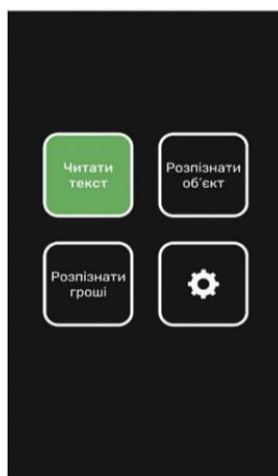


Рис. 1. Макет екрану головного меню мобільного застосунку

Якщо користувач бажає розпізнати текст, він натискає на кнопку “Читати текст”. Телефон робить знімок тексту, який розпізнається мобільним додатком та озвучується функцією “Talk Back”. Якщо користувач хоче зробити ще один знімок, він так само натискає кнопку “Читати текст” (див. рис. 2).

Далі користувачу необхідно застосувати інший функціонал додатку, для чого він натискає кнопку “назад” у верхньому лівому куті, яка повертає його на екран з основним функціоналом.

Користувач може розпізнати об’єкт, що знаходиться поруч із ним, для чого потрібно натиснути кнопку “Розпізнати об’єкт”, після натискання кнопки вмикається камера мобільного телефону, яку користувач наводить на якийсь об’єкт. Після наведення камери на об’єкт, користувач натискає кнопку “розпізнати об’єкт”, телефон робить знімок і описує, що він сфотографував. Якщо користувачу необхідно розпізнати ще один об’єкт, він повторно натискає кнопку “розпізнати об’єкт” у нижній частині екрану, телефон робить знімок і описує, що він сфотографував.

Якщо користувач хоче розпізнати гроші (українські гривні), для чого він натискає кнопку “Розпізнати гроші” (див. рис. 2), відкривається камера мобільного телефону, яку користувач може навести на об’єкт (купюру). Після наведення камери на об’єкт, користувач натискає кнопку “Розпізнати гроші”, телефон робить знімок об’єкта й озвучує номінал купюри (1-2-5-10 грн). Далі користувачу необхідно застосувати інший функціонал додатку, для чого він натискає кнопку “назад”, яка повертає його на екран з основним функціоналом.

Якщо користувачу необхідно розпізнати ще один об’єкт, він повторно натискає кнопку “розпізнати об’єкт” у нижній частині екрану, телефон робить знімок і описує, що він сфотографував.

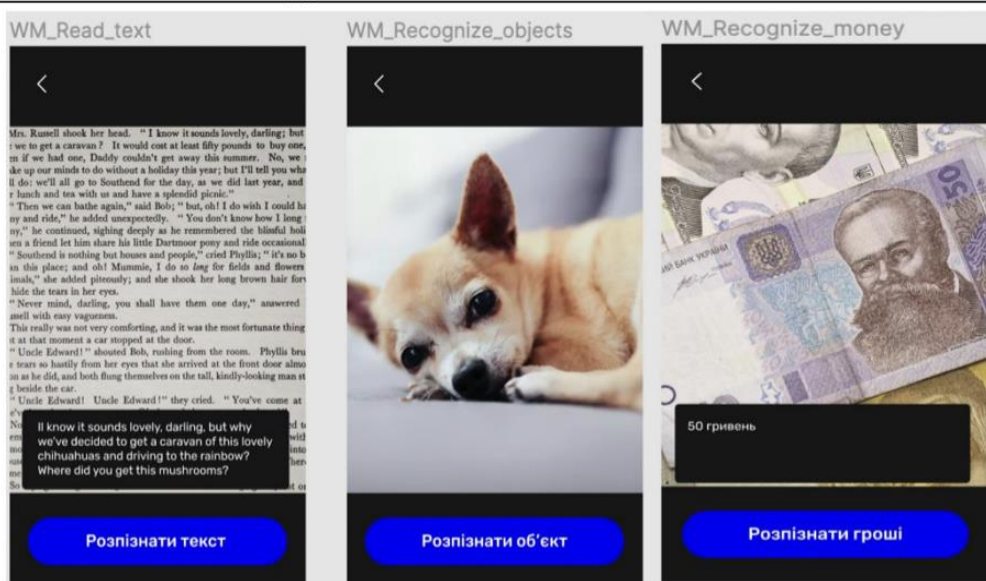


Рис. 2. Вікна: “Розпізнати текст”, “Розпізнати об’єкт”, “Розпізнати гроші” (зліва направо)

Під час розробки мобільного застосунку з функціями розпізнавання об’єктів, грошей, тексту для людей з вадами зору були виконані поставлені задачі, реалізовано відповідний функціонал, спроектовано інтерфейс. Надалі доцільним є доопрацювання структури самого застосунку – збільшення функціональних можливостей та включення додаткових функцій для більшої зручності користувачів.

Список використаних джерел

1. Akkara JD, Kuriakose A. Smartphone apps for visually impaired persons (Online). Kerala J Ophthalmol 2019;31:242-8. URL: <https://www.kjophthal.com/article.asp?issn=0976-6677;year=2019;volume=31;issue=3;spage=242;epage=248;aulast=Akkara>
2. Github: веб-сайт (Online). URL: <https://github.com/>
3. Vision Camera: веб-сайт (Online). URL: <https://github.com/mrousavy/react-native-vision-camera>
4. React Native Splash Screen: веб-сайт (Online). URL: <https://reactnavigation.org>
5. Mobile application testing Wikipedia: веб-сайт (Online). URL: https://en.wikipedia.org/wiki/Mobile_application_testing