

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інженерії програмного забезпечення**

**ДОПУЩЕНО ДО ЗАХИСТУ**

Завідувач кафедри, канд. техн. наук, доцент

\_\_\_\_\_ Є. О. Давиденко

«\_» \_\_\_\_\_ 2024 р

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**МОБІЛЬНИЙ ЗАСТОСУНОК ІНТЕРНЕТ-МАГАЗИНУ З**  
**ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ**  
**ЗОБРАЖЕНЬ**

Спеціальність «Інженерія програмного забезпечення»

121 — КРБ — 408.22011016

*Здобувач: 4 курсу, групи 408*

\_\_\_\_\_ О. В. Муха

«\_» \_\_\_\_\_ 2024 р

*Керівник: PhD, ст. викладач кафедри ІІЗ*

\_\_\_\_\_ К. О. Антіпова

«\_» \_\_\_\_\_ 2024 р

*Консультант: канд. техн. наук., доцент*

\_\_\_\_\_ А. О. Алексєєва

«\_» \_\_\_\_\_ 2024 р

**Миколаїв – 2024**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інженерії програмного забезпечення**

ЗАТВЕРДЖУЮ

Завідувач кафедри, канд. техн. наук, доцент

\_\_\_\_\_ Є. О. Давиденко

«\_\_» \_\_\_\_\_ 2024 р

**ЗАВДАННЯ**  
**на виконання кваліфікаційної роботи бакалавра**

Видано здобувачу групи 408 факультету комп'ютерних наук

Мухі Олегу Вікторовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи

Мобільний застосунок інтернет-магазину з використанням технології  
розпізнавання зображень

Затверджена наказом по ЧНУ від «22» грудня 2023 р. № 269

2. Строк представлення кваліфікаційної роботи «\_\_» червня 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом є мобільний застосунок інтернет-магазину з  
впровадженою технологією розпізнавання зображень

4. Перелік питань, що підлягають розробці:

- дослідження предметної області та аналіз існуючих аналогів;
- формування специфікації вимог до мобільного застосунку, враховуючи функціональність розпізнавання зображень товарів;
- визначення архітектури мобільного застосунку;
- моделювання і проектування користувацького інтерфейсу для програмного забезпечення;

- розробка програмного забезпечення;
- тестування мобільного застосунку, зокрема перевірка коректності розпізнавання зображень.

## 5. Перелік графічних матеріалів

Презентація

---

## 6. Завдання до спеціальної частини

---

## 7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алексєєва А. О.	Кафедра екології	Спеціальна частина з охорони праці

Керівник роботи PhD, ст. викладач кафедри ІІЗ Антіпова Катерина

Олександрівна

---

*(посада, прізвище, ім'я, по батькові)*

*(підпис)*

Завдання прийнято до виконання

Муха Олег Вікторович

*(прізвище, ім'я, по батькові здобувача)*

*(підпис)*

Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 2024 р.

**КАЛЕНДАРНИЙ ПЛАН**  
**виконання кваліфікаційної роботи**

Тема: Мобільний застосунок інтернет-магазину з використанням технології розпізнавання зображень

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	15.12.2023	22.12.2023	Виконано
2.	Літературний огляд згідно теми КРБ	20.01.2024	30.01.2024	Виконано
3.	Складання календарного плану згідно плану КРБ	01.02.2024	02.02.2024	Виконано
4.	Аналіз предметної області	04.02.2024	06.02.2024	Виконано
5.	Розробка специфікації до КРБ	07.02.2024	10.02.2024	Виконано
6.	Проектування інтерфейсу користувача	12.02.2024	15.03.2024	Виконано
7.	Розробка функціональності мобільного застосунку	16.03.2024	01.04.2024	Виконано
8.	Інтеграція технології розпізнавання зображень	02.04.2024	12.04.2024	Виконано
9.	Тестування та налагодження створеного застосунку	15.04.2024	16.04.2024	Виконано
10.	Розробка спеціальної частини з охорони праці	17.04.2024	30.04.2024	Виконано
11.	Відгук керівника КРБ	01.05.2024	02.05.2024	Виконано
12.	Оформлення КРБ та презентації для захисту	03.05.2024	10.05.2024	Виконано
13.	Попередній захист КРБ	03.06.2024	03.06.2024	Виконано
14.	Подання КРБ рецензенту	04.06.2024	11.06.2024	
14.	Відгук рецензента КРБ	12.06.2024	18.06.2024	
16.	Захист КРБ	27.06.2024	28.06.2024	

Розробив здобувач Муха Олег Вікторович  
(прізвище, ім'я, по батькові)

  
(підпис)

«\_\_» \_\_\_\_\_ 0 2024 р.

Керівник роботи PhD, ст. викладач кафедри ІПЗ Антіпова Катерина  
Олександрівна

(посада, прізвище, ім'я, по батькові)

(підпис)

«\_\_» \_\_\_\_\_ 2024 р.

## АНОТАЦІЯ

до кваліфікаційної роботи бакалавра  
«Мобільний застосунок інтернет-магазину з використанням  
технології розпізнавання зображень»  
Здобувач 408 гр.: Муха Олег Вікторович  
Керівник: PhD, ст. викладач кафедри ІІЗ Антіпова К. О.

Актуальність теми даної кваліфікаційної роботи визначається швидким розвитком сучасних технологій та постійною зміною уподобань споживачів у сфері електронної комерції. Зростання популярності мобільних пристроїв та їх роль у повсякденному житті ставлять перед розробниками вимогу до мобільних застосунків інтернет-магазинів – надати користувачам зручний та швидкий інструмент для пошуку та придбання товарів.

Ця робота присвячена розробці мобільного застосунку інтернет-магазину з використанням технології розпізнавання зображень, щоб полегшити процес пошуку та покупки товарів користувачами.

Об'єкт роботи: процес розробки мобільних застосунків інтернет-магазинів.

Предмет роботи: технології розпізнавання зображень для оптимізації процесу пошуку та покупки товарів у мобільному застосунку.

Мета роботи: розробка мобільного застосунку інтернет-магазину із впровадженою технологією розпізнавання зображень для оптимізації процесу здійснення покупок користувачами.

Для досягнення поставленої мети необхідно виконати наступний перелік завдань:

- 1) дослідити предметну область згідно отриманої теми КРБ;
- 2) розробити інтерфейс мобільного застосунку, який буде інтуїтивно зрозумілим та зручним для користувачів;
- 3) інтегрувати технологію розпізнавання зображень для можливості пошуку товарів за їх зображенням;
- 4) забезпечити безпеку та конфіденційність даних користувачів, які використовують застосунок;

5) провести тестування та вдосконалення застосунку згідно з отриманими результатами.

Структура кваліфікаційної роботи бакалавра включає вступ, чотири розділи, висновки та перелік джерел посилань.

Вступ окреслює важливість теми, формулює мету, дає короткий огляд задачі, а також описує предмет, об'єкт та методи дослідження.

Перший розділ присвячений аналізу існуючих підходів у сфері розробки мобільних застосунків інтернет-магазинів. На основі цього аналізу сформульовано функціональні вимоги до мобільного застосунку, які визначають його основні функціональні можливості та вимоги до взаємодії з користувачем.

Другий розділ розглядає процес моделювання функціональних та інформаційних моделей мобільного застосунку з використанням технології розпізнавання зображень. Спочатку визначаються основні функції застосунку, які повинні бути реалізовані для забезпечення зручного та ефективного користування.

Третій розділ присвячений визначенню архітектури мобільного застосунку, вибору необхідних технологій та мов програмування для реалізації функціоналу. Описуються ключові компоненти застосунку, їх взаємодія та структура. Також розробляються діаграми, які відображають логіку роботи програмного забезпечення.

Четвертий розділ описує реалізацію програмного забезпечення згідно з розробленою архітектурою та моделями. Розробляються окремі модулі та компоненти застосунку, які потім інтегруються в цілісну систему. Після цього проводиться тестування застосунку для перевірки його працездатності та відповідності вимогам. У розділі також розглядається інтеграція TensorFlow Lite.

Кваліфікаційна робота бакалавра викладена на 64 сторінок, вона містить 4 розділи, 30 ілюстрацій, 8 таблиць, 12 джерел в переліку посилань.

Ключові слова: *мобільний застосунок, інтернет-магазин, розробка мобільного застосунку, програмування на Kotlin, розробка під Android, розпізнавання зображень, штучний інтелект, машинне навчання.*

## **ABSTRACT**

of the Bachelor's Thesis

"Mobile application of an online store using image recognition technology"

Student of group 408: Mukha Oleh Viktorovich

Supervisor: Ph.D., Senior Lecturer Antipova K. O.

The relevance of the topic of this bachelor's qualification work is determined by the rapid development of modern technologies and the constant change in consumer preferences in the field of e-commerce. The increasing popularity of mobile devices and their role in daily life impose a requirement on developers of mobile applications for online stores to provide users with a convenient and quick tool for searching and purchasing products.

This work is dedicated to the development of a mobile application for an online store using image recognition technology to facilitate the process of searching and purchasing products for users.

Object of the work: the process of developing mobile sites for online stores.

Subject of the work: recognition technologies to optimize the process of searching and purchasing goods from a mobile store.

Aim of the work: to develop a mobile application for an online store with integrated image recognition technology to optimize the shopping process for users.

To achieve the set goal, the following tasks need to be completed:

- 1) Investigate the subject area according to the given topic;
- 2) Develop a user interface for the mobile application that will be intuitive and convenient for users;
- 3) Integrate image recognition technology to enable searching for products by their images;
- 4) Ensure the security and confidentiality of user data using the application;
- 5) Conduct testing and improvement of the application based on the obtained results.

The structure of the bachelor's qualification work includes an introduction, four chapters, conclusions, and a list of references.

The introduction outlines the importance of the topic, formulates the aim, provides a brief overview of the tasks, and describes the subject, object, and research methods.

The first chapter is dedicated to analyzing existing approaches in the field of developing mobile applications for online stores. Based on this analysis, functional requirements for the mobile application have been formulated, defining its main functional capabilities and user interaction requirements.

The second chapter examines the process of modeling functional and informational models of the mobile application using image recognition technology. First, the main functions of the application that need to be implemented to ensure convenient and effective use are determined. Then, an informational model is developed that reflects the structure and relationships between different components of the application.

The third chapter focuses on defining the architecture of the mobile application, selecting the necessary technologies and programming languages for implementing the functionality. The key components of the application, their interaction, and structure are described. Diagrams that depict architectural solutions and the logic of the software operation are also developed. The conclusions provide an evaluation of the work done and the results obtained.

The fourth chapter describes the implementation of the software according to the developed architecture and models. Individual modules and components of the application are developed and then integrated into a cohesive system. Following this, the application is tested to verify its operability and compliance with the requirements. The chapter also considers the integration of TensorFlow Lite.

The bachelor's qualification work is presented on 64 pages, it includes 4 chapters, 30 illustrations, 8 tables, and 12 sources in the list of references.



## ЗМІСТ

<b>ПЕРЕЛІК СКОРОЧЕНЬ</b> .....	3
<b>ВСТУП</b> .....	4
<b>1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ</b> .....	5
1.1 Огляд мобільних застосунків-аналогів інтернет-магазинів.....	5
1.2 Аналіз системи, що розробляється.....	10
1.3 Специфікація вимог до мобільного застосунку.....	11
Висновки до розділу 1 .....	14
<b>2 МОДЕЛЮВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ</b> .....	15
2.1 Етапи реалізації проєкту .....	15
2.2 Створення Use Case.....	16
2.3 Алгоритм роботи програмного забезпечення.....	20
2.4 Розробка діаграми класів .....	23
Висновки до розділу 2 .....	25
<b>3 ПРОЄКТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ ТА ОГЛЯД СТЕКУ ВИКОРИСТАНИХ ТЕХНОЛОГІЙ</b> .....	26
3.1 Створення UML-діаграм.....	26
3.1.1 Діаграми взаємодії .....	27
3.1.2 Діаграма станів та переходів .....	30
3.1.3 Діаграми компонентів та розгортання .....	33
3.1.4 Діаграма пакетів.....	35
3.1.5 Модель бази даних.....	36
3.2 Огляд стеку технологій.....	38
3.2.1 Мова програмування.....	39
3.2.2 Технології розробки інтерфейсу користувача.....	40
3.2.3 Технології зберігання та управління даними .....	42
3.2.4 Технологія розпізнавання зображень .....	43
Висновки до розділу 3 .....	45
<b>4 ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ЗАСТОСУНКУ ТА ПРОВЕДЕННЯ ТЕСТУВАННЯ</b> .....	46
4.1 Огляд дизайну мобільного застосунку .....	46

Кафедра інженерії програмного забезпечення	
Мобільний застосунок інтернет-магазину з використанням технології розпізнавання зображень	2
4.2 Реалізація програмного компоненту розпізнавання зображень .....	53
4.3 Тестування мобільного застосунку .....	58
Висновки до розділу 4 .....	60
<b>ВИСНОВКИ</b> .....	62
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ</b> .....	63

## ПЕРЕЛІК СКОРОЧЕНЬ

БД	–	база даних
КРБ	–	кваліфікаційна робота бакалавра
ОС	–	операційна система
ШІ	–	штучний інтелект
ПЗ	–	програмне забезпечення
МЗ	–	мобільний застосунок
ТРЗ	–	технологія розпізнавання зображень
ТЗ	–	технічне завдання
API	–	application programming interface
UI	–	user interface
MVVM	–	model-view-view model
DI	–	dependency injection
UML	–	unified modeling language
TM	–	teachable machine
TFL	–	tensorflow lite

## ВСТУП

**Актуальність теми.** У сучасному світі зростає популярність онлайн-шопінгу, особливо через мобільні пристрої. Технологія розпізнавання зображень стає все більш важливою, оскільки вона дозволяє користувачам швидко знаходити та купувати товари, подивившись на них через камеру свого смартфона. Це спрощує процес пошуку товарів та забезпечує зручність користувачів. Розробка мобільного застосунку інтернет-магазину з використанням технології розпізнавання зображень може сприяти підвищенню конкурентоспроможності підприємств у сфері електронної торгівлі. Крім того, вона може покращити користувацький досвід і збільшити обсяги продажів, що є важливими практичними перевагами для бізнесу.

**Об'єктом роботи** є мобільні застосунки інтернет-магазинів.

**Предметом роботи** є впровадження технології розпізнавання зображень для полегшення процесу пошуку та покупки товарів у мобільному застосунку.

**Метою роботи** є розробка мобільного застосунку інтернет-магазину із впровадженою технологією розпізнавання зображень для полегшення процесу здійснення покупок користувачами.

Для досягнення поставленої мети необхідно виконати наступний перелік завдань:

- 1) дослідити предметну галузь згідно отриманої теми КРБ;
- 2) розробити інтерфейс мобільного застосунку, який буде інтуїтивно зрозумілим та зручним для користувачів;
- 3) інтегрувати технологію розпізнавання зображень для можливості пошуку товарів за їх зображенням;
- 4) забезпечити безпеку та конфіденційність даних користувачів, які використовують застосунок;
- 5) провести тестування та вдосконалення застосунку згідно з отриманими результатами.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Огляд мобільних застосунків-аналогів інтернет-магазинів

Огляд застосунків-аналогів для мобільних інтернет-магазинів є ключовим кроком у визначенні необхідного функціоналу і виявленні можливих недоліків конкуруючих рішень. Це дозволяє уникнути їх під час розробки власного продукту, підвищуючи його якість та конкурентоспроможність. Успішний мобільний застосунок для інтернет-магазину повинен не тільки відповідати стандартам функціоналу аналогічних продуктів, але й пропонувати інноваційні можливості для залучення більшої кількості користувачів та покращення їх досвіду.

Для аналізу було відібрано три відомі мобільні застосунки інтернет-магазинів, які займають лідируючі позиції на ринку: Rozetka.ua (табл. 1.1), Prom.ua (табл. 1.2) та Shafa.ua (табл. 1.3).

#### ***Rozetka.ua***

Rozetka.ua — це провідний онлайн-магазин в Україні, який пропонує широкий асортимент товарів, від електроніки до побутових товарів. Мобільний застосунок Rozetka.ua робить покупки зручнішими завдяки інтуїтивно зрозумілому інтерфейсу і швидкому пошуку (рис. 1.1).

Таблиця 1.1 – Опис аналогу «Rozetka.ua»

<b>Назва</b>	Rozetka.ua
<b>Архітектура</b>	N-tier
<b>Виробник</b>	Rozetka
<b>Мова реалізації</b>	Kotlin
<b>Функції</b>	1. Пошук товарів 2. Перегляд каталогу товарів 3. Порівняння товарів 4. Додавання товарів у кошик

Кінець таблиці 1.1

<b>Функції</b>	<ol style="list-style-type: none"> <li>Оформлення замовлень</li> <li>Оплата замовлень</li> <li>Можливість залишення відгуків</li> <li>Система рейтингу товару</li> </ol>
<b>Переваги</b>	<ol style="list-style-type: none"> <li>Можливість використання застосунку без створення облікового запису</li> <li>Підтримка різноманітних способів оплати</li> <li>Швидке оновлення асортименту</li> </ol>
<b>Недоліки</b>	<ol style="list-style-type: none"> <li>Інтуїтивно незрозумілий інтерфейс</li> <li>Часом велика кількість рекламних матеріалів</li> <li>Висока навантаженість серверів під час розпродажів, що може сповільнити роботу застосунку</li> </ol>
<b>Джерело інформації</b>	<a href="https://rozetka.com.ua/ua/">https://rozetka.com.ua/ua/</a>



Рисунок 1.1 – Інтерфейс застосунку «Rozetka.ua»

### ***Prom.ua***

Prom.ua — це українська торгова платформа, яка об'єднує велику кількість малого та середнього бізнесу. Мобільний застосунок дозволяє користувачам легко купувати товари від різних продавців і порівнювати пропозиції (рис. 1.2).

Таблиця 1.2 – Опис аналогу «Prom.ua»

<b>Назва</b>	Prom.ua
<b>Архітектура</b>	Microservices
<b>Виробник</b>	TIU Company
<b>Мова реалізації</b>	Kotlin
<b>Функції</b>	<ol style="list-style-type: none"><li>1. Пошук товарів</li><li>2. Перегляд каталогу товарів</li><li>3. Порівняння товарів</li><li>4. Додавання товарів у кошик</li><li>5. Оформлення замовлень</li><li>6. Оплата замовлень</li><li>7. Можливість залишення відгуків</li><li>8. Система рейтингу товара</li></ol>
<b>Переваги</b>	<ol style="list-style-type: none"><li>1. Зручність пошуку за допомогою фільтрів</li><li>2. Можливість використання застосунку без створення облікового запису</li><li>3. Прямий зв'язок з продавцями</li></ol>
<b>Недоліки</b>	<ol style="list-style-type: none"><li>1. Часто зустрічаються непідтверджені продавці, що може призвести до ризиків для покупців</li></ol>
<b>Джерело інформації</b>	<a href="https://prom.ua/">https://prom.ua/</a>

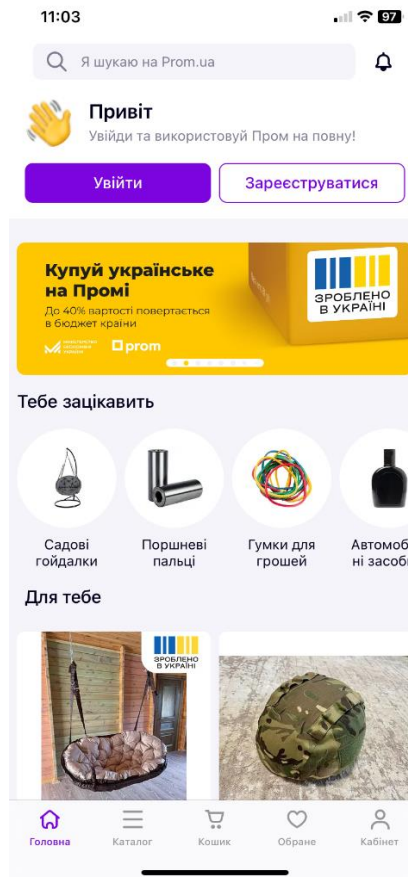


Рисунок 1.2 – Інтерфейс застосунку «Prom.ua»

***Shafa.ua***

Shafa.ua — це платформа для купівлі та продажу одягу, що в основному спеціалізується на вторинному ринку. Застосунок допомагає користувачам легко знаходити, продавати та купувати одяг, аксесуари та взуття (рис. 1.3).

Таблиця 1.3 – Опис аналогу «Shafa.ua»

<b>Назва</b>	Shafa.ua
<b>Архітектура</b>	Microservices
<b>Виробник</b>	Shafa
<b>Мова реалізації</b>	Kotlin
<b>Функції</b>	<ol style="list-style-type: none"> <li>1. Пошук товарів</li> <li>2. Перегляд каталогу товарів</li> <li>3. Порівняння товарів</li> </ol>



Кінець таблиці 1.3

<b>Функції</b>	<ol style="list-style-type: none"><li>4. Додавання товарів у кошик</li><li>5. Оформлення замовлень</li><li>6. Оплата замовлень</li><li>7. Можливість залишення відгуків</li><li>8. Система рейтингу товару</li></ol>
<b>Переваги</b>	<ol style="list-style-type: none"><li>1. Можливість використання застосунку без створення облікового запису</li><li>2. Швидке оновлення асортименту</li></ol>
<b>Недоліки</b>	<ol style="list-style-type: none"><li>1. Обмежена інформація про стан товару, що може ввести в оману покупців</li></ol>
<b>Джерело інформації</b>	<a href="https://shafa.ua/">https://shafa.ua/</a>

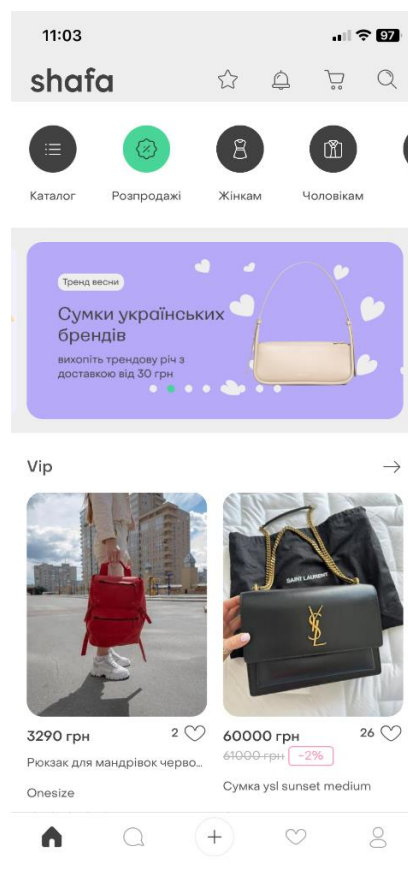


Рисунок 1.3 – Інтерфейс застосунку «Shafa.ua»

## 1.2 Аналіз системи, що розробляється

Розробка мобільного застосунку інтернет-магазину з використанням технології розпізнавання зображень (рис. 1.4) має на меті полегшити процес покупок та покращити користувацький досвід. Користувач може легко знаходити бажані товари шляхом завантаження зображення або фотографування предмета, а система автоматично ідентифікує продукт і пропонує найбільш релевантні результати. Це дозволяє клієнтам швидше знаходити товари, підвищуючи шанси на продаж.

Система повинна підтримувати багатокористувацький режим і забезпечувати стабільну роботу на платформі Android, а також бути доступною 24/7.

Таблиця 1.4 – Опис застосунку, що розробляється

<b>Функції</b>	<ol style="list-style-type: none"> <li>1. Реєстрація користувачів</li> <li>2. Авторизація користувачів</li> <li>3. Використання застосунку без облікового запису</li> <li>4. Рейтинг товарів</li> <li>5. Повідомлення про акції</li> <li>6. Пошук товарів за їх зображенням</li> <li>7. Пошук товарів за назвою</li> <li>8. Пошук товарів за категорією</li> <li>9. Додавання товарів у кошик</li> <li>10. Оформлення замовлення</li> <li>11. Додавання товарів до списку улюблених</li> </ol>
<b>Користувачі</b>	<ol style="list-style-type: none"> <li>1. Незареєстрований користувач</li> <li>2. Авторизований користувач</li> <li>3. Адміністратор</li> </ol>

Кінець таблиці 1.4

<b>Сценарії роботи системи</b>	<ol style="list-style-type: none"><li>1. Користувач встановлює застосунок і реєструється, щоб отримати доступ до персоналізованих функцій.</li><li>2. Користувач може скористатися пошуком за зображенням або за назвою продукту для швидкого знаходження товару.</li><li>3. Користувач може додати товар до списку бажань або кошика для подальшого замовлення.</li><li>4. Система пропонує схожі товари.</li><li>5. Користувач може видалити товар з кошика або оформити замовлення.</li><li>6. Адміністратор відповідає за управління застосунком та модерацію вмісту.</li><li>7. Система повідомляє користувача про акції та знижки.</li></ol>
--------------------------------	--

### 1.3 Специфікація вимог до мобільного застосунку

#### ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

##### Призначення застосунку, для якого розробляється програмне забезпечення

Призначенням мобільного застосунку є полегшення процесу покупок у інтернет-магазині за допомогою технології розпізнавання зображень. Користувач може сфотографувати або завантажити зображення товару, а система знайде відповідні продукти в каталозі магазину.

##### Межі проєкту ПЗ

Крайня дата завершення роботи над ПЗ – 22.06.2024р.

#### ЗАГАЛЬНИЙ ОПИС

##### Сфера застосування

Цей застосунок призначений для полегшення процесу пошуку та купівлі товарів в інтернет-магазині за допомогою розпізнавання зображень. Його можна

використовувати для швидкого пошуку продуктів у таких категоріях, як електроніка, одяг, побутові товари тощо.

### **Характеристики користувачів**

Основні характеристики користувачів: наявність смартфона або планшета з доступом до Інтернету та бажання здійснювати покупки онлайн.

### **Загальна структура і склад системи**

Основні компоненти системи включають сервер, локальну базу даних (Room), клієнт (мобільний застосунок), який взаємодіє з сервером через API.

### **Загальні обмеження**

Для коректної роботи застосунку потрібен стабільний доступ до Інтернету та смартфон з ОС Android версії 8.0 або вище.

## **ФУНКЦІЇ СИСТЕМИ МОБІЛЬНОГО ЗАСТОСУНКУ**

### *Функція пошуку товарів за зображенням*

#### **Опис функції**

Користувач може сфотографувати або завантажити зображення товару, і система знайде відповідні продукти в каталозі магазину.

#### **Вхідна і вихідна інформація**

Вхідна інформація: зображення товару, яке надав користувач.

Вихідна інформація: перелік продуктів, які відповідають критеріям пошуку.

#### **Функціональні вимоги**

Система машинного навчання для розпізнавання зображень, база даних товарів магазину та доступ до Інтернету.

### *Функція додавання товару у кошик*

#### **Опис функції**

Функція дозволяє користувачам додавати декілька товарів у кошик.

#### **Вхідна і вихідна інформація**

Вхідна інформація: обрані товари користувачем для оформлення замовлення.

Вихідна інформація: список обраних товарів у кошику.

#### **Функціональні вимоги**

База даних товарів та можливість додавання товару у кошик в інтерфейсі застосунку.

## **ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ**

### **Джерела та зміст вхідної інформації (даних)**

Вхідна інформація надходить від користувачів, які використовують функції застосунку, такі як пошук та купівля товарів.

### **Вимоги до способів організації, збереження та ведення інформації**

Обмін даними здійснюється через RESTful API, базою даних буде Room (SQLite).

## **ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ**

Для застосунку не передбачено жорстких обмежень. Він повинен працювати на будь-якому сучасному смартфоні з підтримкою ОС Android 8.0 та вище, та об'ємом ОЗП 1 ГБ та більше.

## **ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

### **Архітектура програмної системи**

База даних: Реляційна БД Room (SQLite).

Мобільний клієнт: Застосунок для Android, розроблений на Kotlin.

### **Системне програмне забезпечення**

Мобільна платформа: Android 8.0 (Oreo) або вище.

Фреймворк: Android SDK для розробки нативних застосунків.

Бібліотеки: Retrofit для взаємодії з API, Glide/Picasso для обробки зображень, Room для роботи з локальною базою даних.

### **Мережне програмне забезпечення**

Протоколи: Використання HTTPS для захищеного обміну даними між сервером і клієнтом.

Обмін даними: RESTful API для взаємодії між мобільним застосунком і сервером.

### **Програмне забезпечення для обробки зображень**

Використання спеціалізованої бібліотки для розпізнавання зображень TensorFlow Lite.

## **ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ**

### **Інтерфейс користувача**

Інтерфейс повинен бути інтуїтивно зрозумілим і зручним, забезпечувати легке використання функцій.

### **Апаратний інтерфейс**

Відповідність принципам Material Design для забезпечення узгодженого користувацького досвіду.

## **ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

### **Доступність**

Підтримка широкого спектра пристроїв Android (8.0 та вище).

### **Продуктивність**

Швидкий час відгуку для взаємодії з користувачем.

### **Надійність**

Надійне збереження даних користувачів та інформації про замовлення.

### **Безпека**

Шифрування конфіденційних даних, використання токенів для авторизації.

## **Висновки до розділу 1**

У першому розділі КРБ проведено огляд та аналіз існуючих аналогів. Аналіз дав можливість виявити їх переваги та недоліки, розвинути навички визначення вимог та покращення функціоналу для розробки власного застосунку.

Також проведено аналіз системи, в якому визначено типи користувачів, основний функціонал та сценарії використання МЗ. Сформовано вимоги до ПЗ, описано призначення та функціонал, визначено архітектуру та технології.

Розділ містить специфікацію вимог до програмного забезпечення, включаючи детальний опис як функціональних, так і нефункціональних вимог, для визначення взаємодії користувачів з програмним забезпеченням.

## 2 МОДЕЛЮВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ

### 2.1 Етапи реалізації проєкту

Розробка мобільного застосунку для інтернет-магазину з використанням технології розпізнавання зображень включає розробку дизайну, створення інтерфейсу та реалізацію функціоналу серверної частини, що може виконуватися як одним розробником, так і командою фахівців. Тривалість розробки залежить від складності проєкту, розміру команди та досвіду фахівців.

Етапи розробки включають:

1. збір інформації про продукт та формування переліку вимог і задач (product backlog). В цьому етапі важливо відповісти на такі питання:

- 1) які проблеми має вирішувати застосунок?
- 2) який результат очікується від розробки?
- 3) який відведено час на розробку?
- 4) який бюджет на розробку?
- 5) який функціонал планується?
- 6) чи існують які-небудь обмеження у використанні?
- 7) які технології використовуватимуться?
- 8) хто є цільовою аудиторією застосунку?

2. Проєктування архітектури системи з розробкою UML-діаграм, таких як діаграми класів, варіантів використання, послідовності та розгортання. Цей етап визначає структуру системи, що дає чітке уявлення про компоненти та їх взаємодію.

3. Створення дизайну інтерфейсу відповідно до вимог. Розробка макетів включає розробку логотипу, вибір шрифтів та колірної гами. Макети допомагають візуалізувати користувацький досвід та інтерфейс застосунку.

4. Розробка front-end та back-end частин включає реалізацію інтерфейсу застосунку для Android-платформи та серверної частини з функціоналом. Важливо

забезпечити оптимальну роботу застосунку та інтеграцію з технологіями розпізнавання зображень.

5. Проведення тестування для виявлення помилок та покращення стабільності застосунку. Після успішного тестування продукт готовий до запуску на ринок.

6. Після запуску необхідно регулярно оновлювати застосунок для підтримки актуального функціоналу, виправлення помилок та покращення продуктивності.

## 2.2 Створення Use Case

Use Case – це опис послідовності дій, відповідно до якого користувач взаємодіє з системою для виконання певного завдання залежно від функціоналу застосунку. Use Case описує, що система повинна робити, а не як вона це робить.

### **Короткий Use Case: Використання функції розпізнавання зображень**

Користувач запускає мобільний застосунок та виконує вхід в обліковий запис. Система перевіряє дані користувача та автентифікує його. Після успішного входу користувач переходить до головного екрану застосунку. Користувач обирає опцію "Пошук за зображенням" у головному меню. Система активує камеру мобільного пристрою із запитом на зйомку або вибір фотографії з галереї. Після отримання зображення система застосовує технологію розпізнавання зображень для ідентифікації товарів. Користувач може вибрати товар з результатів розпізнавання або виправити запит за допомогою редагування або повторної зйомки. Після вибору товару користувач переходить до сторінки з деталями товару, де може переглянути більше інформації, ціну та здійснити покупку.



## **Поверхневий Use Case: Здійснення платежу**

### **Головний сценарій (успішний):**

1. Користувач відкриває мобільний застосунок та авторизується в обліковому записі.
2. Користувач обирає опцію "Пошук за зображенням" та фотографує або вибирає зображення товару.
3. Система розпізнає товар на зображенні та виводить результати пошуку.
4. Користувач обирає потрібний товар, переглядає його деталі та додає до кошика.
5. Користувач переходить до оформлення замовлення, вказує адресу доставки та обирає метод оплати.
6. Після успішного оформлення замовлення система підтверджує його та відображає номер замовлення.

### **Альтернативні сценарії:**

1. Користувач вибирає товар, якого немає у наявності. Система повідомляє про відсутність товару та пропонує обрати альтернативний товар.
2. Під час оплати відбувається помилка з банківським переказом. Система відображає повідомлення про помилку та пропонує спробувати знову або вибрати інший метод оплати.
3. Товар, який користувач додав до кошика, вийшов із продажу внаслідок одночасної покупки іншим користувачем. Система повідомляє користувача про це та пропонує видалити товар з кошика або замінити його на інший.
4. Система недоступна через технічні проблеми. Користувач отримує повідомлення про недоступність системи та рекомендується спробувати пізніше.
5. Користувач здійснює покупку, але в результаті виникає помилка сервера. Система автоматично записує цю помилку та повідомляє адміністратора про необхідність вирішення проблеми.

## Повний Use Case: Додавання товару у кошик

Таблиця 2.1 – Повний Use Case

Usecase section	Comment
Use Case Name	Додавання товару у кошик
Scope	System
Level	User-goal
Primary Actor	Зареєстрований користувач
Stakeholders and interests	Користувач - придбання товару; Магазин - продаж товару
Preconditions	Користувач повинен бути авторизований у мобільному додатку
Success guarantee	Обраний товар додано до кошика успішно
Main Success Scenario	<ol style="list-style-type: none"> <li>1. Користувач відкриває мобільний додаток.</li> <li>2. Користувач авторизується у системі.</li> <li>3. Користувач переходить до каталогу товарів.</li> <li>4. Користувач шукає товар за допомогою розпізнавання зображень.</li> <li>5. Система відображає результати розпізнавання.</li> <li>6. Користувач обирає потрібний товар.</li> <li>7. Користувач натискає кнопку "Додати до кошика".</li> <li>8. Система додає обраний товар до кошика користувача.</li> <li>9. Система підтверджує додавання товару у кошик.</li> <li>10. Користувач може продовжити покупки або переглянути кошик.</li> </ol>
Extensions	<ol style="list-style-type: none"> <li>1. Користувач не авторизований і не може додати товар у кошик. <ol style="list-style-type: none"> <li>- 1.1 Система перенаправляє користувача на сторінку авторизації.</li> <li>- 1.2 Після авторизації користувач продовжує додавати товар у кошик.</li> </ol> </li> <li>2. Система не може розпізнати товар за зображенням. <ol style="list-style-type: none"> <li>- 2.1 Користувач обирає товар з каталогу вручну.</li> <li>- 2.2 Користувач повідомляє про помилку розпізнавання.</li> </ol> </li> <li>3. Помилка додавання товару у кошик (наприклад, недостатньо товару). <ol style="list-style-type: none"> <li>- 3.1 Система відображає повідомлення про помилку.</li> <li>- 3.2 Користувач може відредагувати кількість товару або скасувати додавання.</li> </ol> </li> <li>4. Користувач вирішує скасувати додавання товару. <ol style="list-style-type: none"> <li>- 4.1 Користувач вибирає опцію "Скасувати" у кошику.</li> <li>- 4.2 Система видаляє товар з кошика користувача.</li> </ol> </li> <li>5. Користувач виходить з додатку без додавання товару у кошик. <ol style="list-style-type: none"> <li>- 5.1 Користувач закриває додаток без виконання покупки.</li> </ol> </li> </ol>

### Кінець таблиці 2.1

Special Requirements	Доступ до камери для використання технології розпізнавання
Technology and Data Variations List	Використання API для розпізнавання зображень
Frequency of Occurrence	Дуже висока

Діаграма варіантів використання (рис. 2.1) надає наочне уявлення про взаємодію користувачів із системою, демонструючи основні сценарії використання мобільного застосунку інтернет-магазину. Ця діаграма показує ключові варіанти використання для різних типів користувачів і відображає взаємозв'язки між ними.

На діаграмі представлені три основні актори:

- Гість. Нерегулярний користувач, який не має зареєстрованого облікового запису, але може переглядати каталог товарів, шукати їх за категоріями та реєструвати новий акаунт.
- Зареєстрований користувач. Має доступ до розширених функцій, таких як перегляд детальної інформації про товар, додавання товарів до списку бажань, перегляд замовлень, оформлення замовлень, пошук товарів за допомогою розпізнавання зображень та інше.
- Адміністратор. Відповідає за адміністрування системи, включаючи управління списком товарів, замовленнями та іншими адміністративними функціями.

Основні варіанти використання, що відображені на діаграмі:

- Гість має можливість зареєструвати новий обліковий запис для отримання розширеного доступу.
- Усі користувачі можуть переглядати товари, використовуючи функцію перегляду каталогу.
- Зареєстровані користувачі можуть скористатися інноваційною функцією пошуку товарів за допомогою зображень.
- Зареєстровані користувачі можуть додавати товари до кошика і здійснювати покупки.

- Користувачі можуть відстежувати статус своїх замовлень.
- Адміністратор має можливість додавати, видаляти та редагувати товари, а також керувати замовленнями.

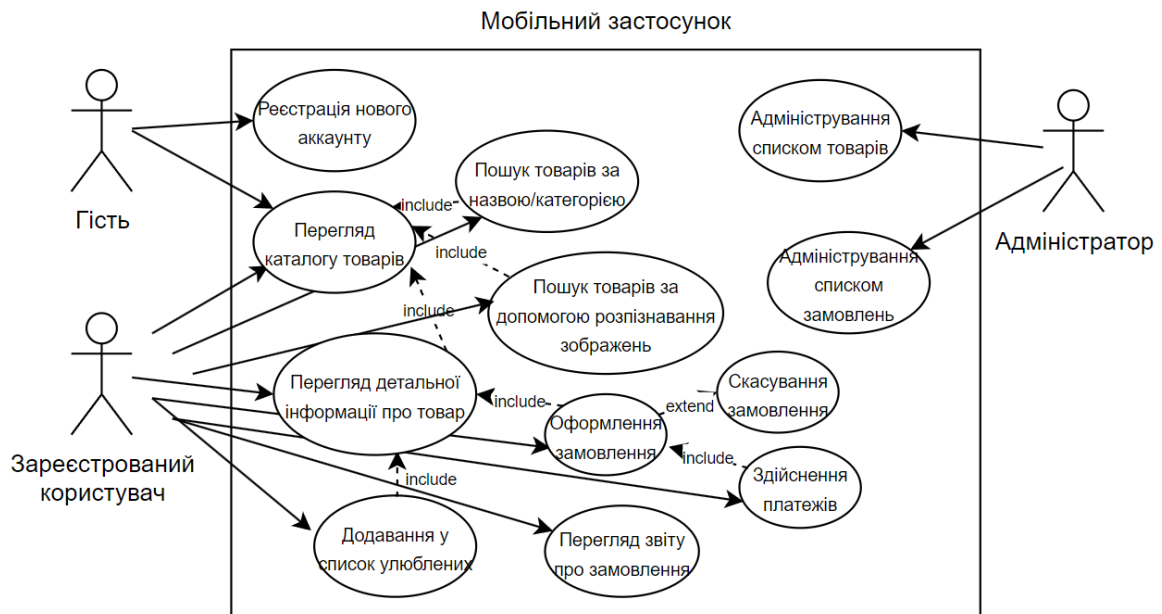


Рисунок 2.1 – Діаграма варіантів використання ПЗ

Діаграма варіантів використання програмного забезпечення необхідна для того, щоб наочно відобразити функціональні вимоги системи та взаємодію між користувачами (акторами) і системою.

### 2.3 Алгоритм роботи програмного забезпечення

Мобільний застосунок має бути зручним та зрозумілим для широкого кола користувачів. Користувач повинен мати змогу легко скористатися застосунком на смартфоні з ОС Android. Програмне забезпечення покликане спростити процес пошуку та купівлі товарів, для чого застосунок міститиме відповідний функціонал.

Функціонал повинен залишатися актуальним і корисним для всіх користувачів: покупців, продавців та адміністраторів. Для наочного відображення алгоритму роботи програмного забезпечення створено три діаграми діяльності, які демонструє послідовність дій, що виконуються під час взаємодії користувача із застосунком:

1) *Пошук товару за допомогою розпізнавання зображення* (рис. 2.2). Діаграма показує основні етапи процесу пошуку товару за допомогою розпізнавання зображення.

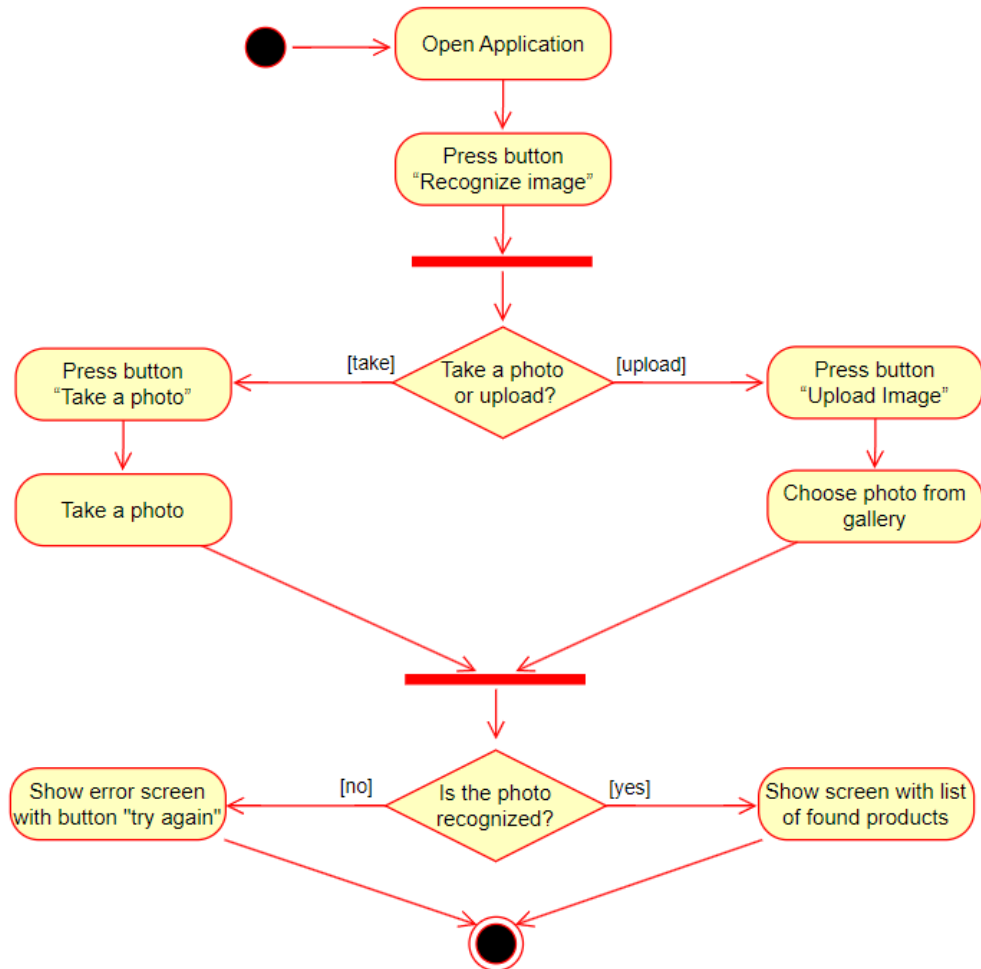


Рисунок 2.2 – Діаграма діяльності для функції «Пошук товару за допомогою розпізнавання зображення»

2) *Перегляд каталогу товарів* (рис. 2.3). Діаграма деталізує процес перегляду каталогу товарів у застосунку.

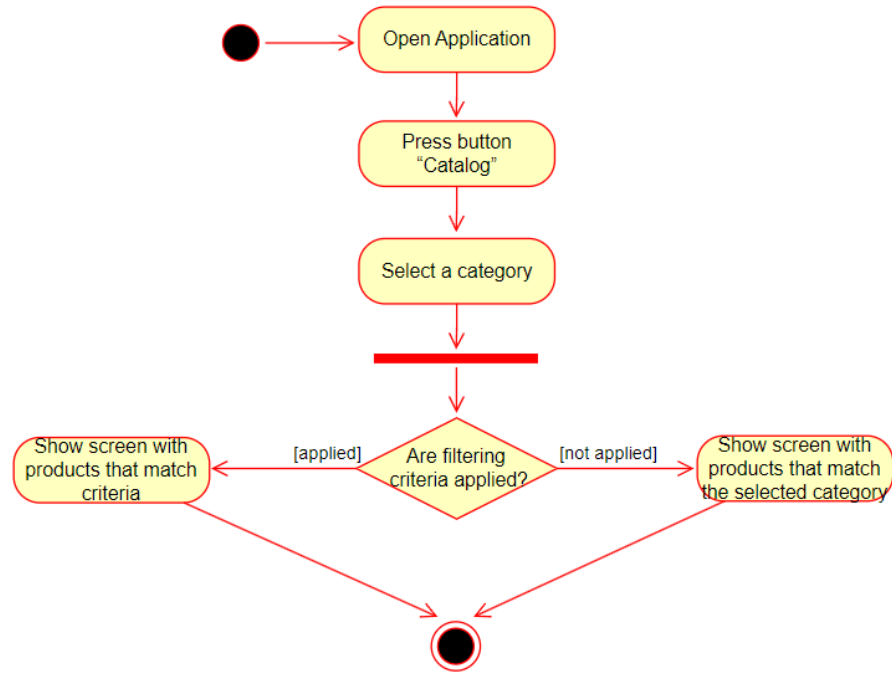


Рисунок 2.3 – Діаграма діяльності для функції «Перегляд каталогу товарів»

3) *Оплата замовлення* (рис. 2.4). Діаграма описує етапи процесу оплати замовлення.

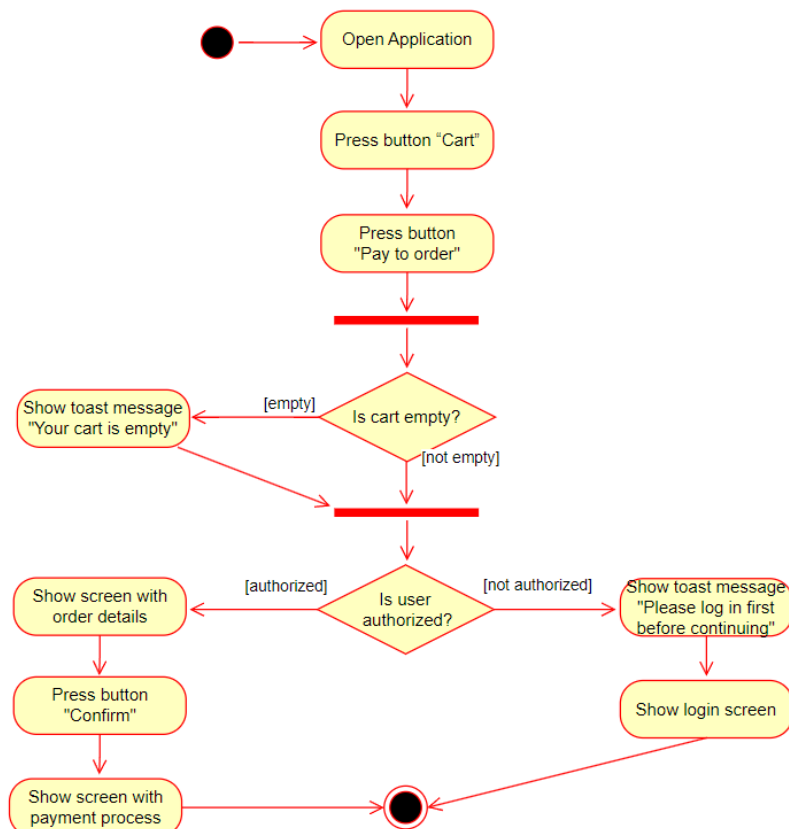


Рисунок 2.4 – Діаграма діяльності для функції «Оплата замовлення»

Діаграми діяльності використовується для моделювання та візуалізації процесів або робочих потоків в системі. Вони допомагають зрозуміти послідовність виконання дій, взаємодію між різними елементами системи, а також виявити можливі оптимізації та вузькі місця.

## 2.4 Розробка діаграми класів

Розробка діаграми класів (рис. 2.5) є ключовим етапом у моделюванні архітектури програмного забезпечення. Вона надає графічне представлення структури системи, описуючи класи, їхні атрибути, методи та взаємозв'язки між ними. Це дозволяє визначити основні компоненти програмного забезпечення та їхні взаємодії ще на етапі проектування [4].

Основні класи, представлені на діаграмі:

1. **Cart (Кошик).** Клас, що зберігає товари, які користувач планує купити, разом з їх загальною ціною. Має методи для додавання, редагування та видалення товарів.
2. **Order (Замовлення).** Представляє замовлення, містить інформацію про товари, користувача, статус та ціну. Методи дозволяють перевіряти статус замовлення, здійснювати оплату та редагувати його.
3. **Product (Товар).** Містить інформацію про товари, їх назву, опис, рейтинг та зображення. Також включає методи для пошуку товарів за категорією, назвою або зображенням.
4. **User (Користувач).** Описує користувачів, їх персональні дані та надає методи для реєстрації, входу, скидання паролю та інших операцій з профілем.
5. **Wishlist (Список бажань).** Містить список товарів, які користувач додав до свого списку бажань. Методи дозволяють додавати та видаляти товари зі списку, переглядати його вміст.
6. **Review (Відгук).** Представляє відгуки користувачів про товари. Включає методи для додавання, редагування та видалення відгуків.

7. Advertisement (Реклама). Містить інформацію про рекламні оголошення, такі як назва, опис, зображення. Методи дозволяють показувати рекламу та отримувати детальну інформацію про неї.

8. Category (Категорія). Представляє категорії товарів, зберігає їх назви та описи. Метод допомагає знайти всі товари, які входять до конкретної категорії.

9. ImageRecognitionService (Сервіс розпізнавання зображень). Забезпечує функції розпізнавання об'єктів на зображеннях. Містить методи для завантаження, зйомки та ідентифікації об'єктів на фото.

10. SearchHistory (Історія пошуку). Відстежує історію пошукових запитів користувача. Методи дозволяють зберігати, показувати або очищати історію.

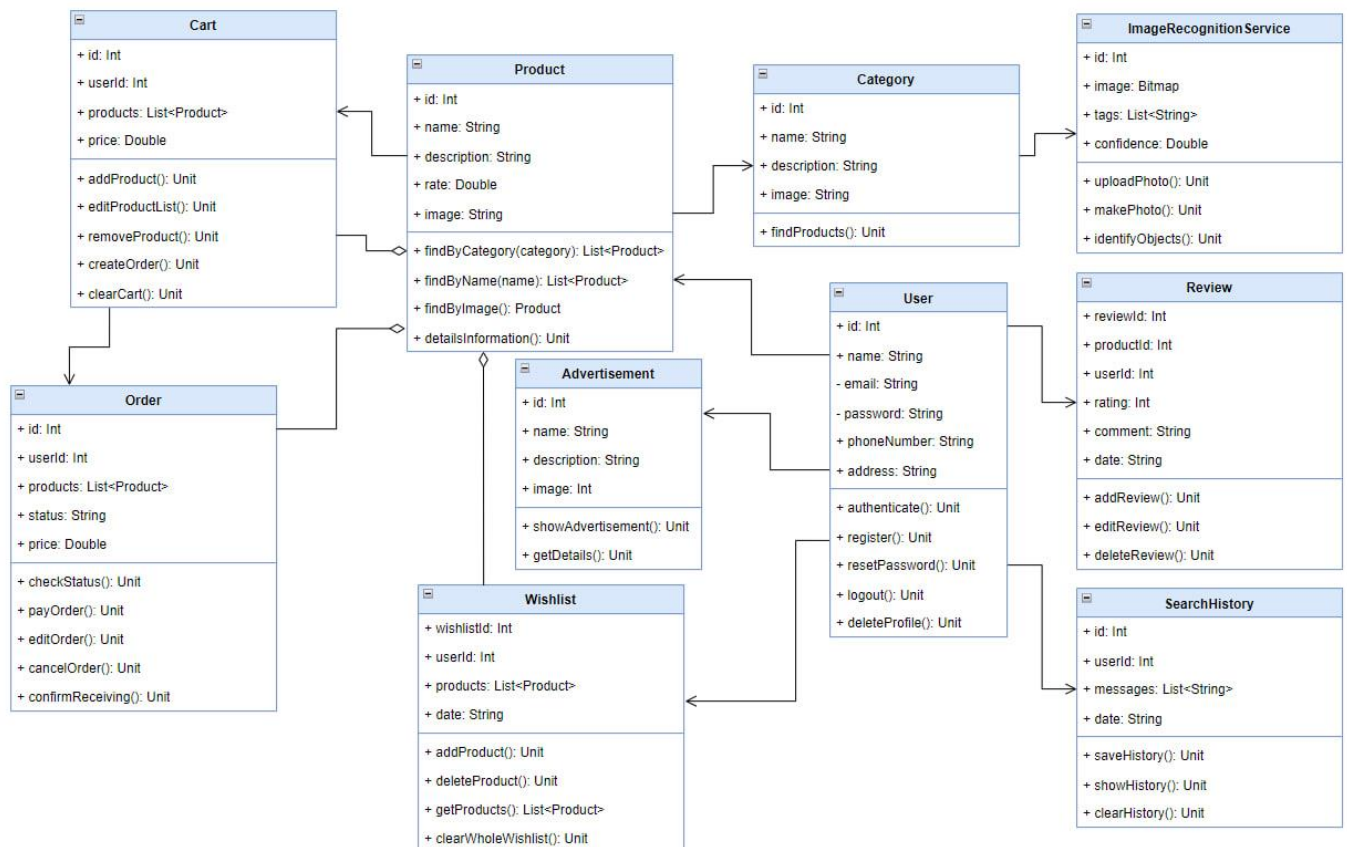


Рисунок 2.5 – Діаграма класів

Діаграма класів демонструє, як різні компоненти системи співпрацюють між собою, щоб забезпечити основні функції застосунку: пошук товарів, управління замовленнями, зберігання відгуків та списків бажань.



## **Висновки до розділу 2**

У другому розділі було розглянуто процес моделювання мобільного застосунку інтернет-магазину, описано етапи реалізації проєкту, алгоритми роботи програмного забезпечення та розроблено діаграму класів. Це дозволило зміцнити навички в моделюванні, плануванні та проєктуванні архітектури програмного забезпечення для мобільних платформ.

Розробка Use Case допомогла визначити основні сценарії використання застосунку, включаючи варіанти взаємодії користувача з функціоналом. На основі коротких і детальних сценаріїв було розроблено повні Use Case, які описують усі деталі використання програмного забезпечення.

Алгоритм роботи програмного забезпечення описав основні кроки взаємодії користувача із застосунком, включаючи пошук товарів, їх додавання в кошик та оформлення замовлення. Було створено діаграму класів, яка дозволяє наочно побачити структуру застосунку, зрозуміти, як різні компоненти взаємодіють між собою та які функції вони виконують.

Використані UML-діаграми в цьому розділі чітко демонструють структуру застосунку, вимоги до нього, а також архітектуру програмного комплексу.

## **3 ПРОЄКТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ ТА ОГЛЯД СТЕКУ ВИКОРИСТАНИХ ТЕХНОЛОГІЙ**

### **3.1 Створення UML-діаграм**

У процесі моделювання та проєктування програмного забезпечення часто використовують UML-діаграми. Вони є одними з найпоширеніших і найбільш зручних інструментів, оскільки дозволяють описувати бізнес-процеси, системне проєктування, а також різні організаційні структури.

У рамках розробки мобільного застосунку інтернет-магазину з використанням технології розпізнавання зображень, було використано наступні UML-діаграми для моделювання різних аспектів системи:

- **Діаграма варіантів використання (Use Case Diagram).** Визначення функціональних вимог до застосунку, включаючи взаємодію користувача з функцією розпізнавання зображень для пошуку товарів.
- **Діаграма класів (Class Diagram).** Моделювання структури класів та об'єктів застосунку, включаючи класи, відповідальні за розпізнавання зображень, обробку запитів та взаємодію з базою даних товарів.
- **Діаграма послідовностей (Sequence Diagram).** Відображення послідовності взаємодій між користувачем, застосунком та зовнішніми системами (наприклад, сервером розпізнавання зображень) під час виконання ключових сценаріїв використання, таких як пошук товару за зображенням.
- **Діаграма діяльності (Activity Diagram).** Моделювання процесів, пов'язаних з розпізнаванням зображень, таких як завантаження зображення, його обробка, отримання результатів та відображення відповідних товарів.
- **Діаграма компонентів (Component Diagram).** Відображення архітектури застосунку, включаючи компоненти, відповідальні за інтерфейс користувача, розпізнавання зображень, обробку даних та зв'язок з сервером.

– Діаграма розгортання (Deployment Diagram). Моделювання фізичного розгортання компонентів застосунку на мобільних пристроях та серверах, включаючи взаємодію з сервісами розпізнавання зображень.

На етапі моделювання мобільного застосунку для інтернет-магазину з використанням технології розпізнавання зображень використано діаграму варіантів використання системи, діаграми діяльності та діаграму класів. Ці діаграми чітко відображають структуру та вимоги до застосунку. На етапі проєктування застосовано станів та переходів, компонентів та розгортання, пакетів, які відображають поведінку, архітектуру ПЗ та його апаратну частину.

### 3.1.1 Діаграми взаємодії

Діаграми взаємодії використовуються для моделювання динамічних аспектів системи, показуючи, як об'єкти взаємодіють один з одним у певній послідовності дій.

Для мобільного застосунку інтернет-магазину з використанням технології розпізнавання зображень було розроблено три діаграми взаємодії.

Перша діаграма (рис. 3.1) демонструє процес відображення списку товарів:

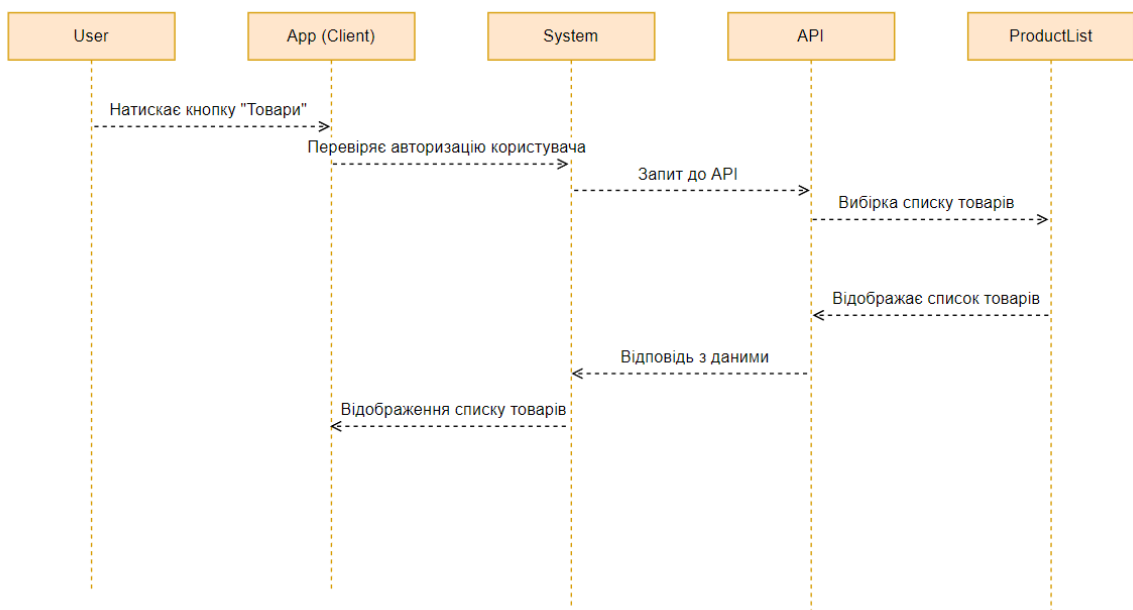


Рисунок 3.1 – Діаграма взаємодії до Use Case «Відображення списку товарів»

Послідовність дій виглядає наступним чином:

1. User натискає кнопку «Товари».
2. App (Client) перевіряє авторизацію користувача.
3. System надсилає запит до API для отримання списку товарів.
4. API здійснює вибірку списку товарів з бази даних.
5. API повертає відповідь з даними до System.
6. System відображає список товарів у App (Client).

Друга діаграма (рис. 3.2) описує процес відображення детальної інформації про товар:

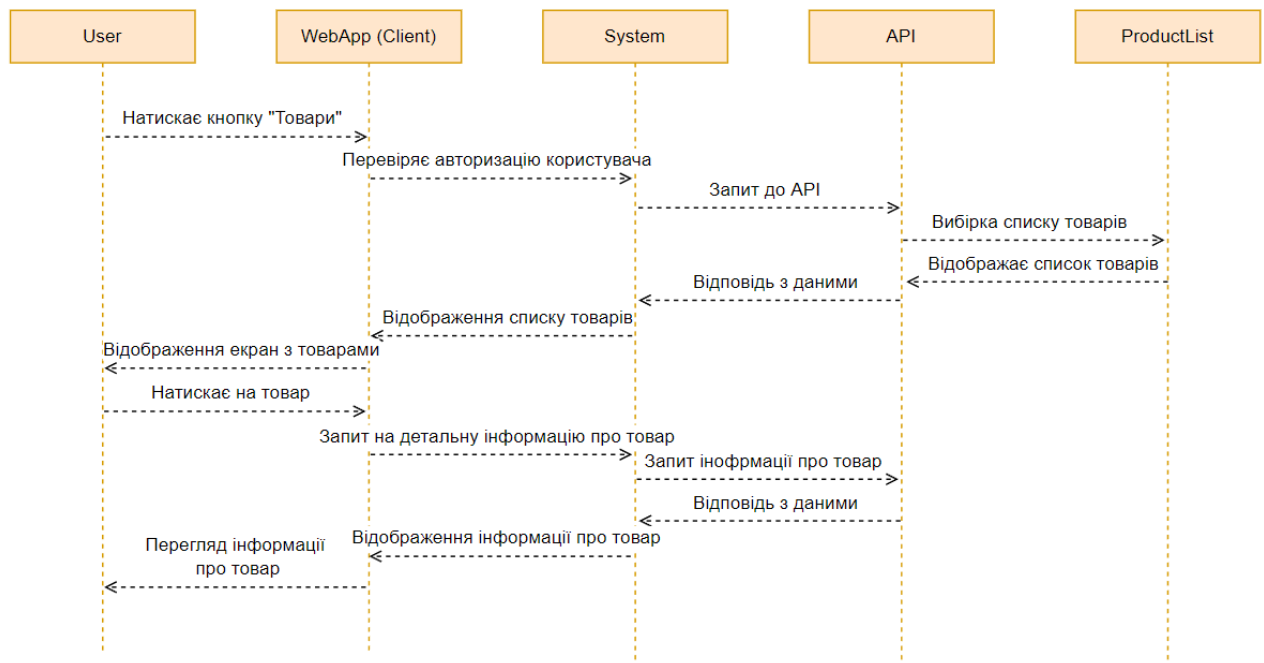


Рисунок 3.2 – Діаграма взаємодії до Use Case «Відображення списку товарів»

Послідовність дій виглядає наступним чином:

1. User натискає кнопку «Товари».
2. WebApp (Client) перевіряє авторизацію користувача.
3. System надсилає запит до API для отримання списку товарів.
4. API здійснює вибірку списку товарів з бази даних.
5. API повертає відповідь з даними до System.
6. System відображає список товарів у WebApp (Client).
7. User натискає на конкретний товар.

8. WebApp (Client) надсилає запит до System для отримання детальної інформації про товар.

9. System надсилає запит до API для отримання детальної інформації про товар.

10. API повертає відповідь з даними до System.

11. System відображає детальну інформацію про товар у WebApp (Client).

Третя діаграма (рис. 3.3) ілюструє процес додавання товару до кошика:

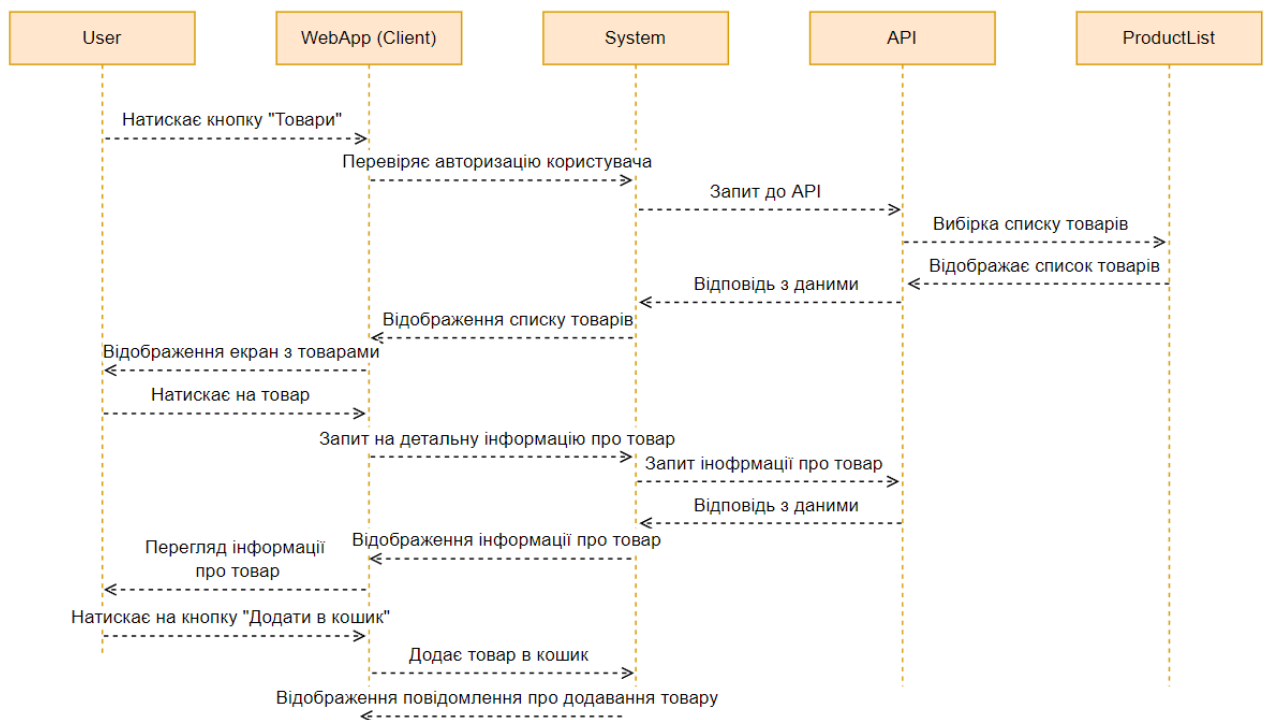


Рисунок 3.3 – Діаграма взаємодії до Use Case «Додавання товару до кошика»

Послідовність дій виглядає наступним чином:

1. User натискає кнопку "Товари".
2. WebApp (Client) перевіряє авторизацію користувача.
3. System надсилає запит до API для отримання списку товарів.
4. API здійснює вибірку списку товарів з бази даних.
5. API повертає відповідь з даними до System.
6. System відображає список товарів у WebApp (Client).
7. User натискає на конкретний товар.

8. WebApp (Client) надсилає запит до System для отримання детальної інформації про товар.
9. System надсилає запит до API для отримання детальної інформації про товар.
10. API повертає відповідь з даними до System.
11. System відображає детальну інформацію про товар у WebApp (Client).
12. User натискає кнопку "Додати в кошик".
13. WebApp (Client) надсилає запит до System для додавання товару до кошика.
14. System додає товар до кошика та відправляє відповідь WebApp (Client).
15. WebApp (Client) відображає повідомлення про успішне додавання товару до кошика.

### 3.1.2 Діаграма станів та переходів

Діаграми станів та переходів (Statechart Diagrams) є важливим інструментом для моделювання поведінки мобільного застосунку інтернет-магазину з використанням технології розпізнавання зображень. Вони дозволяють візуалізувати динаміку системи, показуючи, як застосунок реагує на різні події та дії користувача.

У рамках проєкту кваліфікаційної роботи бакалавра було розроблено 3 ключові діаграми станів та переходів:

- 1) загальна діаграма станів та переходів (рис. 3.4);
- 2) діаграма станів для usecase «Створення та оплата замовлення» (рис. 3.5);
- 3) діаграма станів для usecase «Розпізнавання об'єкту за допомогою камери» (рис. 3.6).

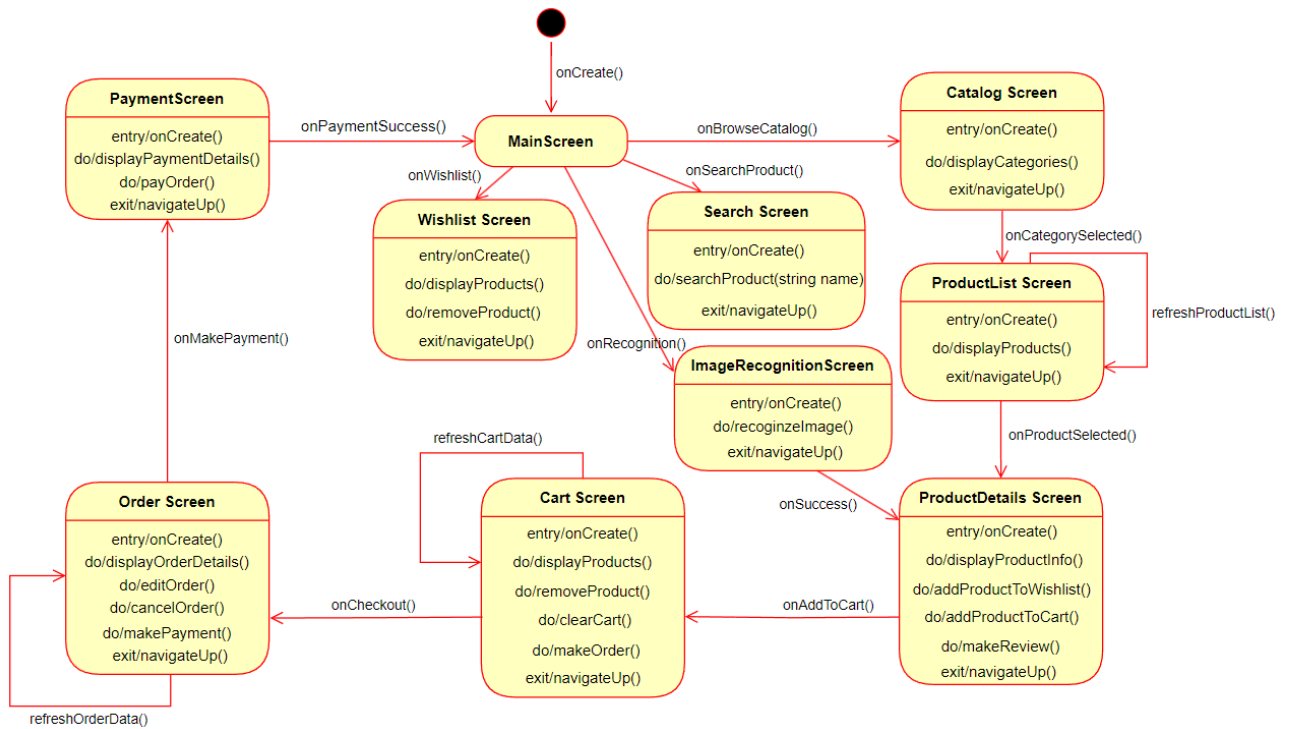


Рисунок 3.4 – Загальна діаграма станів

Ця діаграма відображає загальний життєвий цикл застосунку та можливі взаємодії користувача з ним. Вона демонструє основні стани застосунку (перегляд товарів, кошик, оформлення замовлення тощо) та переходи між ними, ініційовані діями користувача (додати до кошика, перейти до оплати тощо).

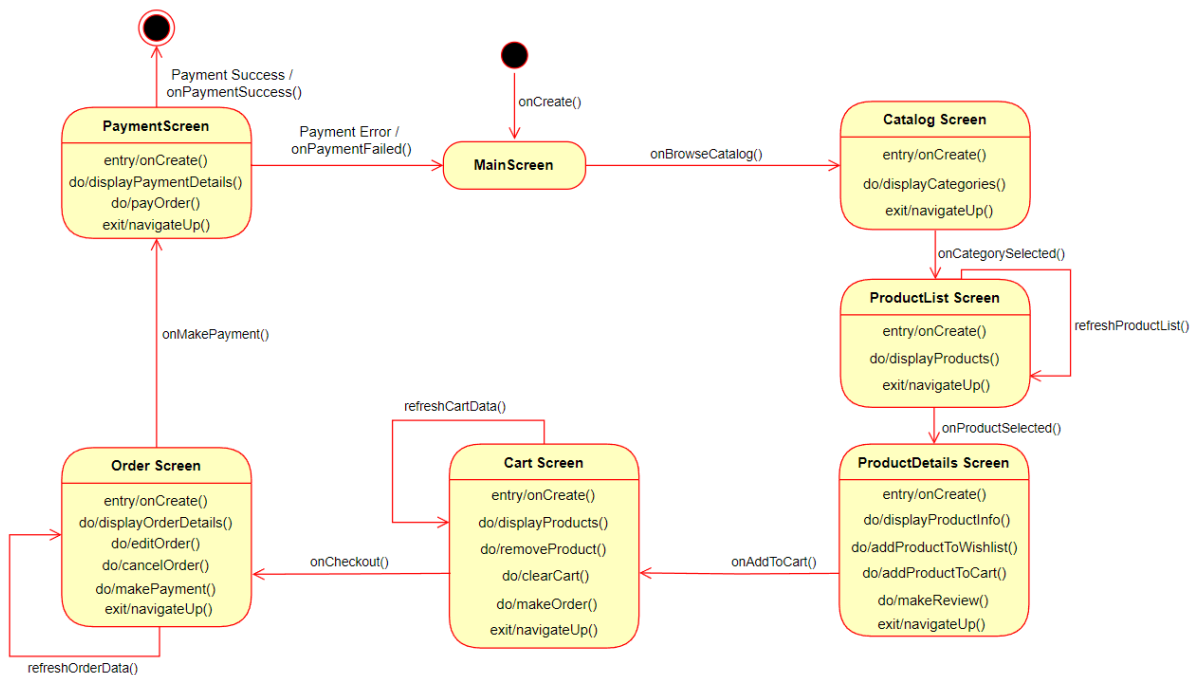


Рисунок 3.5 – Діаграма станів для usecase «Створення та оплата замовлення»

Ця діаграма деталізує процес оформлення замовлення, починаючи з додавання товарів у кошик і закінчуючи підтвердженням оплати. Вона показує можливі стани замовлення та переходи між ними, пов'язані з діями користувача та системою.

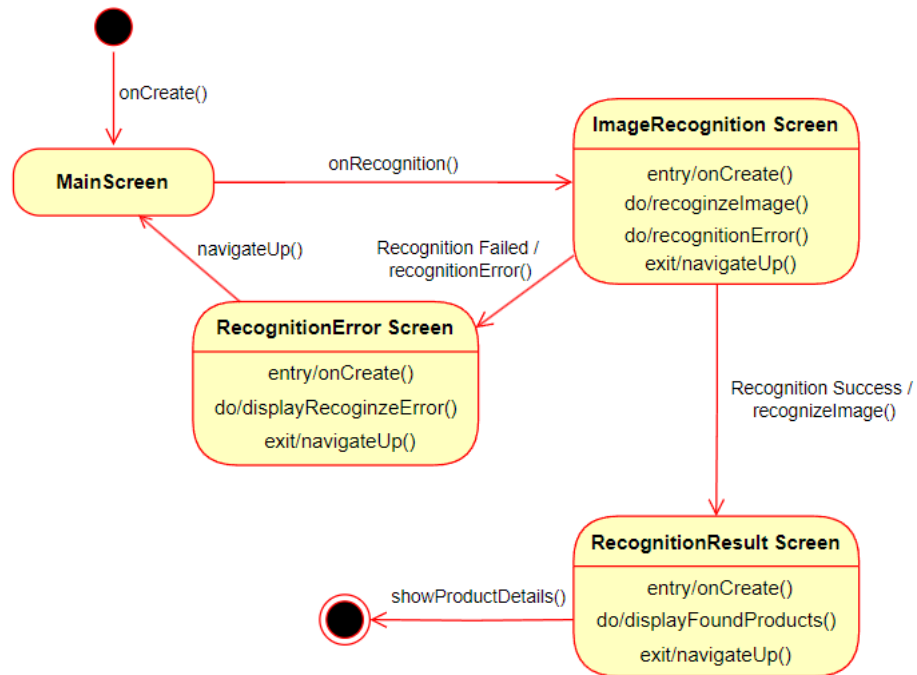


Рисунок 3.6 – Діаграма станів для usecase «Розпізнавання об'єкту за допомогою камери»

Ця діаграма ілюструє процес розпізнавання об'єктів за допомогою камери смартфона для виконання дій, таких як додавання товарів до кошика або вибір персонального консультанта. Вона відображає стани процесу розпізнавання та переходи між ними, зумовлені діями користувача та результатами роботи алгоритму розпізнавання.

Ці діаграми станів та переходів служать важливим інструментом для розуміння та аналізу поведінки мобільного застосунку, допомагаючи виявити потенційні проблеми та оптимізувати взаємодію з користувачем.



### 3.1.3 Діаграми компонентів та розгортання

Діаграми компонентів і розгортання використовуються для моделювання фізичних аспектів програмного забезпечення. Вони допомагають визначити структуру системи, взаємозв'язки між її частинами та відображають апаратні й програмні компоненти, необхідні для виконання застосунку [11].

#### *Діаграма компонентів*

На діаграмі компонентів (рис. 3.7) представлено структуру мобільного застосунку інтернет-магазину, поділену на три шари: app, data та domain:

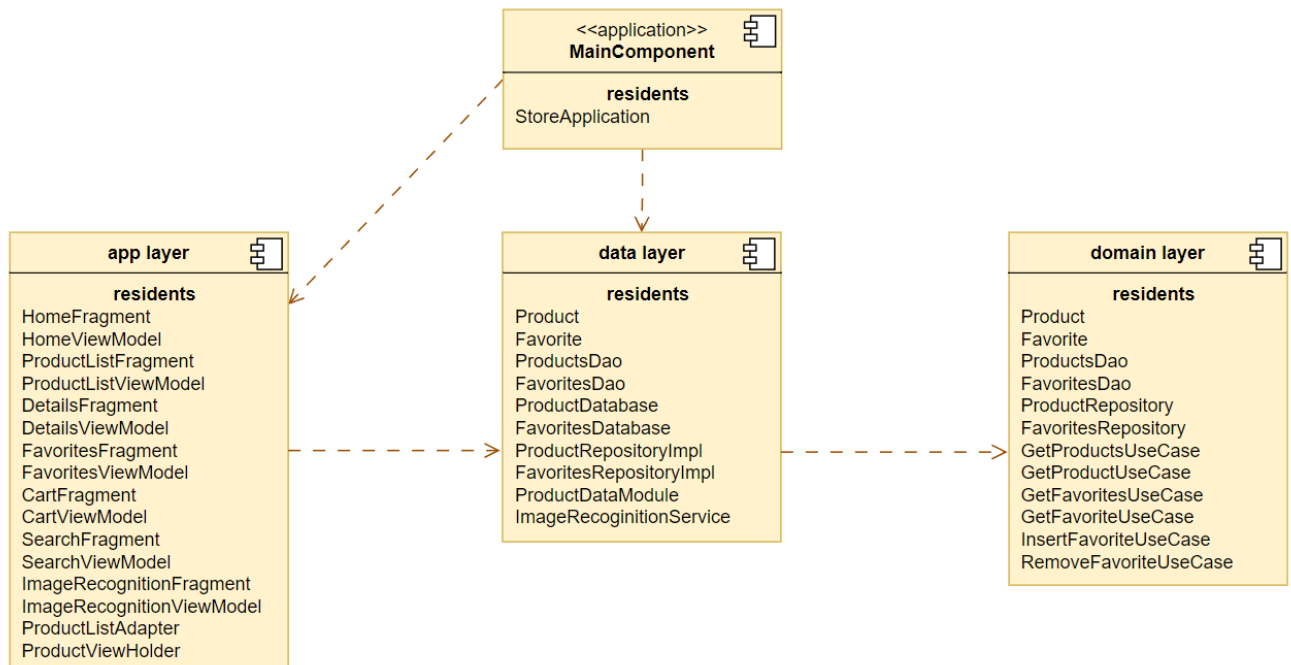


Рисунок 3.7 – Діаграма компонентів

- 1) app layer містить компоненти, що відповідають за інтерфейс користувача та взаємодію з користувачем;
- 2) data layer включає компоненти, що забезпечують доступ до даних та їх збереження;
- 3) domain layer охоплює логіку застосунку та бізнес-процеси.

**Діаграма розгортання**

На діаграмі розгортання (рис. 3.8) представлено фізичну інфраструктуру мобільного застосунку з використанням технології розпізнавання зображень.

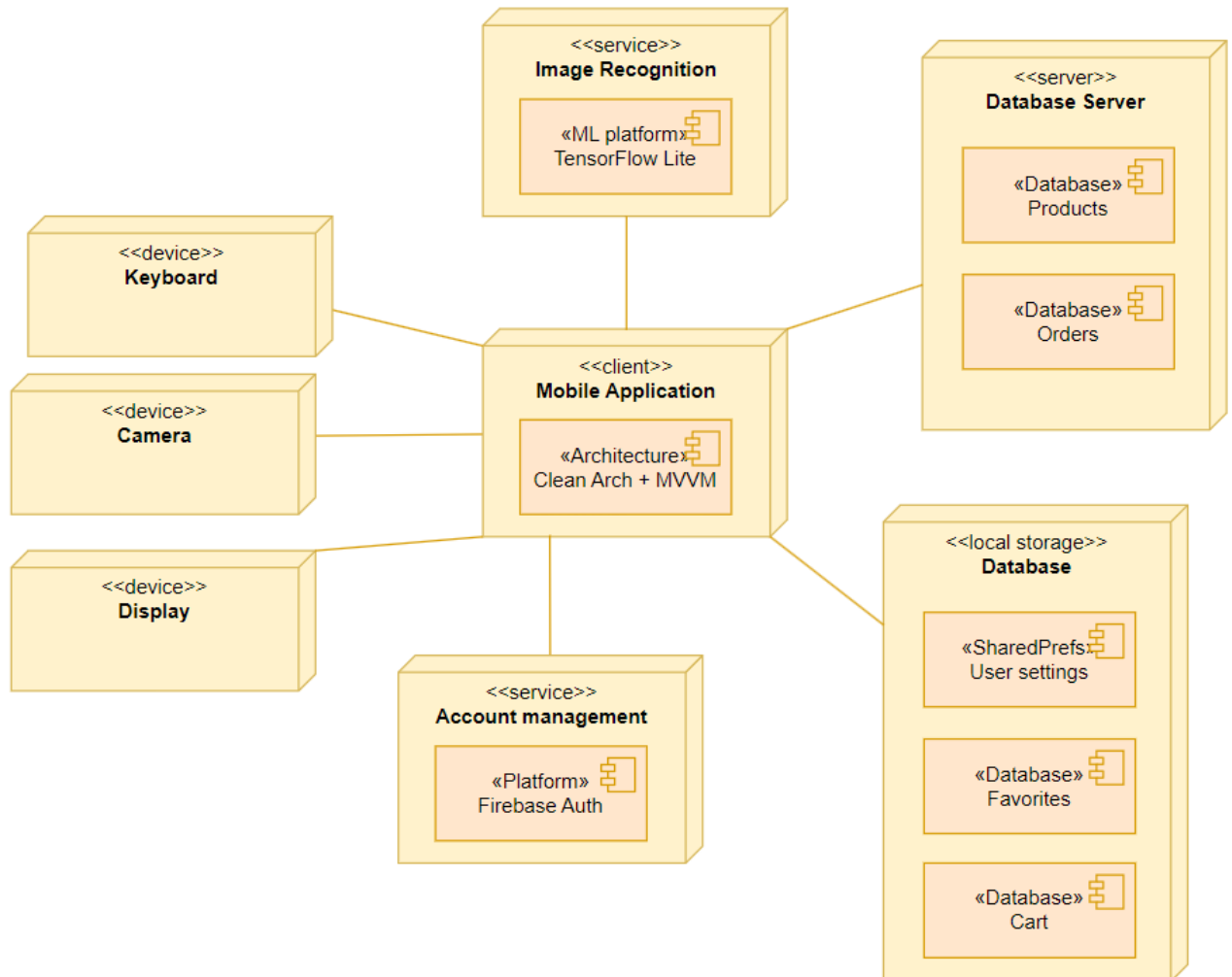


Рисунок 3.8 – Діаграма розгортання

Ця діаграма відображає, як компоненти застосунку взаємодіють між собою та з апаратним забезпеченням для забезпечення функціональності застосунку [12]:

- 1) Mobile Application. Використовує Clean Architecture + MVVM. Підключається до інших компонентів через сервіси та пристрої.
- 2) Image Recognition. Використовує платформу TensorFlow Lite для розпізнавання зображень.
- 3) Account Management. Використовує платформу Firebase Auth для управління обліковими записами користувачів.

- 4) Database Server. Зберігає дані про продукти та замовлення в базі даних.
- 5) Local Storage. Містить налаштування користувача (SharedPreferences), а також бази даних для обраного, замовлень та кошика.
- 6) Devices. Пристрої, що взаємодіють із застосунком: клавіатура моб. пристрою, камера та дисплей.

### 3.1.4 Діаграма пакетів

Діаграми пакетів використовуються для організації елементів програмного забезпечення у вигляді логічних груп. Це допомагає структурувати код та забезпечити його модульність і легкість у підтримці. Діаграми пакетів також відображають відносини між різними пакетами, що складають систему [10].

На діаграмі пакетів (рис. 3.9) представлено структуру МЗ, розділену на три основні шари: application layer, data layer та domain layer.

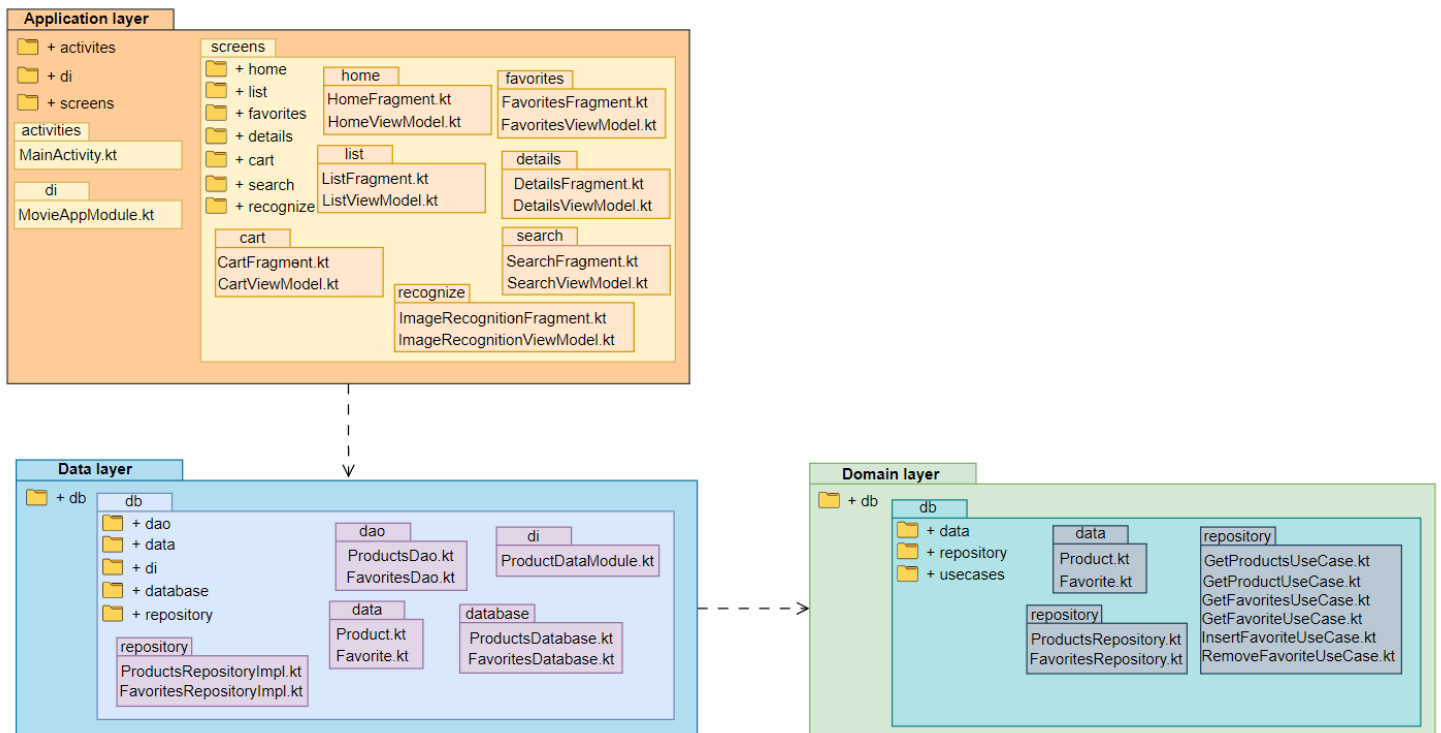


Рисунок 3.9 – Діаграма пакетів

Application Layer включає пакети, що відповідають за інтерфейс користувача та основну логіку застосунку.

Основні пакети:

- 1) activities містить головні активності застосунку (MainActivity.kt).
- 2) di-модуль для ін'єкції залежностей (MovieAppModule.kt).
- 3) screens включає різні екрани застосунку, кожен з яких представлений окремими фрагментами та моделями перегляду (наприклад, HomeFragment.kt, HomeViewModel.kt, CartFragment.kt, CartViewModel.kt тощо).

Data Layer відповідає за збереження та доступ до даних.

Основні пакети:

- 1) dao містить об'єкти доступу до даних (ProductsDao.kt, FavoritesDao.kt).
- 2) database бази даних застосунку (ProductDatabase.kt, FavoritesDatabase.kt).
- 3) repository реалізації репозиторіїв для доступу до даних (ProductsRepositoryImpl.kt, FavoritesRepositoryImpl.kt).
- 4) di модуль для ін'єкції залежностей у шарі даних (ProductDataModule.kt).

Domain Layer включає бізнес-логіку та випадки використання.

Основні пакети:

- 1) data моделі даних (Product.kt, Favorite.kt).
- 2) repository інтерфейси репозиторіїв (ProductsRepository.kt, FavoritesRepository.kt).
- 3) usecases випадки використання функціоналу МЗ (GetProductsUseCase.kt, GetProductUseCase.kt, InsertFavoriteUseCase.kt тощо).

### 3.1.5 Модель бази даних

Модель БД (рис. 3.10) складається з наступних сутностей: Кошик (Cart), Товар (Product), Категорія (Category), Замовлення (Order), Користувач (User), Реклама (Advertisement), Список бажань (Wishlist), Відгук (Review), Служба розпізнавання зображень (ImageRecognitionService) та Історія пошуку

(SearchHistory). Кожна з цих сутностей має свої атрибути та визначені взаємозв'язки з іншими сутностями.

Взаємозв'язки між сутностями:

- Кошик (Cart) містить зв'язок з User (користувачем), який вказує, що кошик належить користувачу, а також з Product (товаром), який вказує на товари в кошику.
- Замовлення (Order) містить зв'язок з Cart (кошиком), Product (товарами) та User (користувачем), вказуючи на відповідні відносини.
- Реклама (Advertisement) має зв'язок з Category (категорією) та User (користувачем).
- Список бажань (Wishlist) зв'язаний з User (користувачем) та Product (товарами).
- Відгук (Review) зв'язаний з Product (товаром) та User (користувачем).
- Служба розпізнавання зображень (ImageRecognitionService) зв'язана з Product (товаром).
- Історія пошуку (SearchHistory) зв'язана з User (користувачем).

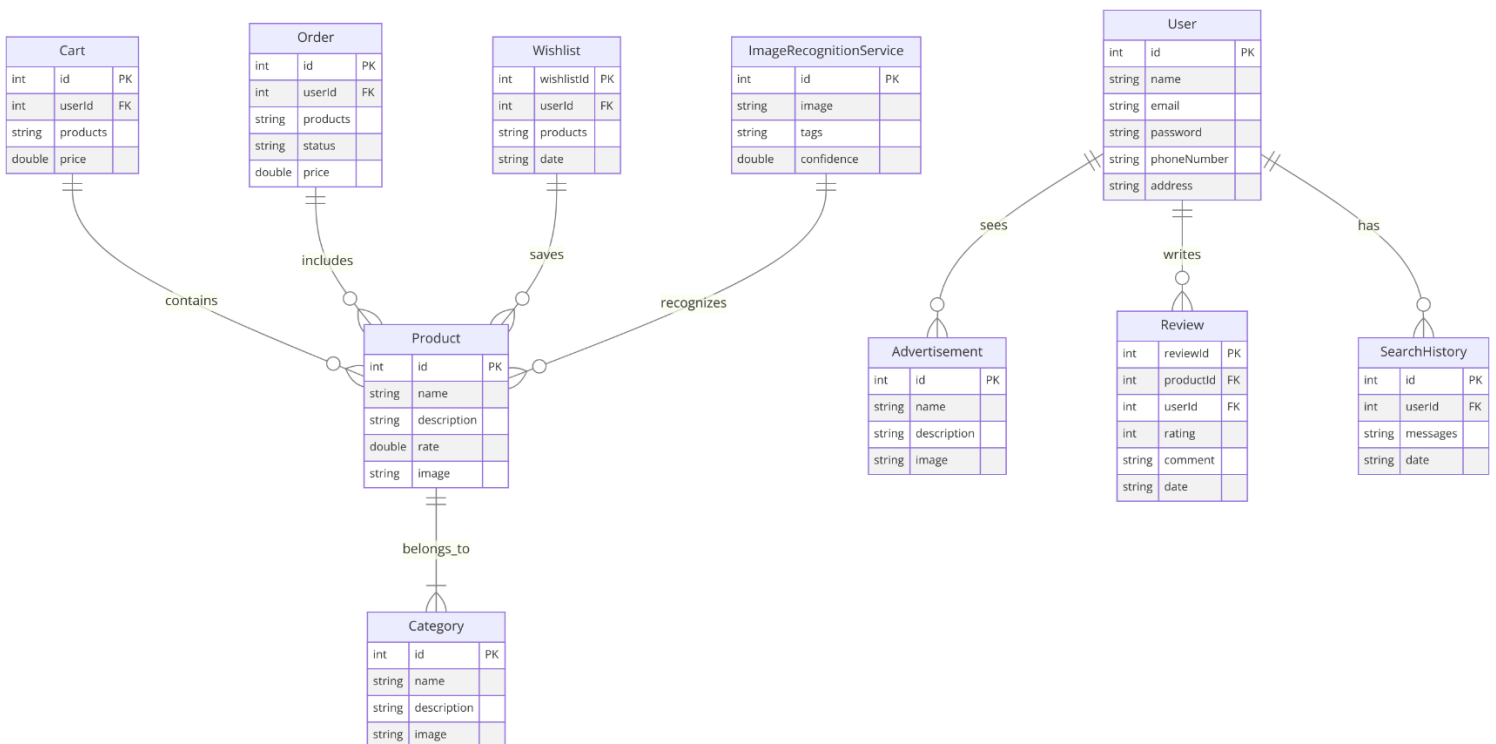


Рисунок 3.10 – Модель бази даних

Модель БД є важливою складовою будь-якої інформаційної системи. Вона забезпечує структурований спосіб зберігання даних, що дозволяє ефективно управляти інформацією та забезпечує високу продуктивність системи. В цій роботі представлена модель бази даних для онлайн-магазину, що включає сутності, атрибути та взаємозв'язки між ними.

### 3.2 Огляд стеку технологій

У процесі розробки мобільного застосунку для інтернет-магазину з використанням технології розпізнавання зображень було використано сучасний стек технологій, який забезпечує високу продуктивність, зручність у розробці та підтримку застосунку. Огляд ключових технологій, що були використані:

1. Мова програмування Kotlin. Kotlin було обрано як основну мову програмування завдяки її сучасним можливостям, безпеці та зручності у використанні. Вона забезпечує високу продуктивність і сумісність з Java, що дозволяє легко інтегрувати існуючі бібліотеки.

2. Архітектура Clean Architecture + MVVM. Для структурування коду застосунку використано архітектурний підхід Clean Architecture у поєднанні з архітектурним патерном MVVM (Model-View-ViewModel). Це забезпечує чітке розділення відповідальностей між різними частинами застосунку, що робить його легким для тестування і підтримки [5, 6].

3. Технологія розпізнавання зображень TensorFlow Lite. Для реалізації функціоналу розпізнавання зображень використано TensorFlow Lite – легку версію платформи TensorFlow, яка оптимізована для мобільних пристроїв. Це дозволяє виконувати машинне навчання безпосередньо на пристрої користувача, знижуючи затримки і підвищуючи конфіденційність даних.

4. Firebase Authentication. Для управління обліковими записами користувачів і їхньої авторизації використано Firebase Authentication. Ця платформа забезпечує безпечний і зручний спосіб автентифікації користувачів за допомогою різних методів, таких як електронна пошта, соціальні мережі тощо.

5. Room Database. Для локального зберігання даних застосунку використано бібліотеку Room Database, яка забезпечує зручний спосіб роботи з реляційними базами даних в Android-застосунках. Вона забезпечує абстракцію над SQLite і дозволяє зменшити кількість коду, необхідного для взаємодії з базою даних.

6. Koin. Для впровадження залежностей у застосунку використано Koin – легку і просту у використанні бібліотеку для DI в Kotlin. Вона дозволяє зменшити кількість ручного коду для створення і управління залежностями між компонентами застосунку.

### 3.2.1 Мова програмування

Основною мовою програмування для розробки мобільного застосунку інтернет-магазину з використанням технології розпізнавання зображень було обрано Kotlin (рис. 3.11). Kotlin є сучасною, статично типізованою мовою програмування, яка набула широкого використання серед розробників Android-застосунків завдяки своїм численним перевагам [9].



Рисунок 3.11 – Логотип Kotlin

Основні переваги Kotlin:

1. Безпека. Kotlin забезпечує високий рівень безпеки за рахунок вбудованої підтримки null-безпеки. Це зменшує кількість помилок, пов'язаних з NullPointerException, що часто зустрічаються у Java.

2. Сумісність з Java. Kotlin повністю сумісний з Java, що дозволяє використовувати існуючі бібліотеки та фреймворки, написані на Java. Це значно полегшує перехід з Java на Kotlin і дає можливість поступово переписувати існуючі проекти.

3. Лаконічність та читабельність коду. Код на Kotlin більш лаконічний і зрозумілий у порівнянні з Java, що дозволяє розробникам писати менше коду для досягнення тієї ж функціональності. Це полегшує підтримку і тестування коду.

4. Функціональне програмування. Kotlin підтримує функціональні елементи програмування, такі як лямбда-вирази, вищі функції, а також розширення функцій. Це дозволяє розробникам писати більш виразний і гнучкий код.

5. Інтероперабельність з Android SDK. Kotlin інтегрується з Android SDK без будь-яких обмежень, що дозволяє використовувати всі можливості платформи Android.

6. Підтримка корутин. Kotlin пропонує потужну підтримку корутин, що спрощує асинхронне програмування та дозволяє легко управляти потоками виконання без використання складних колбеків або сторонніх бібліотек.

У процесі розробки мобільного застосунку інтернет-магазину Kotlin використовувався для створення всіх основних компонентів застосунку, таких як UI-компоненти, модулі доступу до даних, бізнес-логіка та обробка запитів до серверу. Завдяки Kotlin вдалося досягти високої продуктивності, надійності та легкості у підтримці коду, що є важливими аспектами для успішного розроблення сучасного програмного забезпечення.

### **3.2.2 Технології розробки інтерфейсу користувача**

Front-end розробка мобільного застосунку інтернет-магазину з використанням технології розпізнавання зображень включає в себе створення інтерфейсу користувача та забезпечення взаємодії користувача із системою. У цьому проекті використовувались такі основні технології та інструменти для front-end розробки:

1. Android SDK є основним набором інструментів для розробки Android-застосунків. Він включає в себе бібліотеки, API та інші інструменти, необхідні для створення, тестування та налагодження застосунків. Android SDK



використовувався для створення всіх UI-компонентів застосунку, обробки подій користувача та інтеграції з іншими службами Android.

2. **Material Design** – це дизайн-система, розроблена Google, яка забезпечує єдиний стиль інтерфейсів для Android-застосунків. Вона включає в себе набір візуальних, рухомих та інтерактивних елементів, що робить застосунки більш привабливими та зручними у використанні [3]. **Material Design** використовувався для створення візуально привабливих і інтуїтивно зрозумілих інтерфейсів користувача, що відповідають сучасним стандартам дизайну.

3. **ViewModel** та **LiveData** – це компоненти архітектурного патерну **MVVM**, що використовуються для управління даними інтерфейсу користувача і забезпечення їх життєвого циклу. **ViewModel** використовувався для зберігання даних, пов'язаних з інтерфейсом користувача, а **LiveData** забезпечував автоматичне оновлення UI при зміні даних.

4. **Navigation Component** – це інструмент з **Android Jetpack**, який забезпечує простий спосіб управління навігацією між екранами застосунку. **Navigation Component** використовувався для організації навігації між різними екранами застосунку, що дозволило забезпечити зручний і логічний спосіб переміщення користувача по застосунку.

5. **Data Binding** – це бібліотека, що дозволяє зв'язувати UI-компоненти із джерелами даних у застосунку, забезпечуючи автоматичне оновлення UI при зміні даних. **Data Binding** використовувався для спрощення зв'язку між UI-компонентами та даними, що зменшило кількість коду і підвищило зручність у підтримці.

Використання цих технологій дозволило створити функціональний, зручний та привабливий інтерфейс користувача, що відповідає сучасним вимогам до якості мобільних застосунків.

### 3.2.3 Технології зберігання та управління даними

Для забезпечення надійного зберігання та управління даними у мобільному застосунку інтернет-магазину з використанням технології розпізнавання зображень було використано дві основні технології: Firebase Firestore та Room.

#### *Firestore*

Firestore – це хмарна база даних від Google, яка забезпечує гнучкий і масштабований спосіб зберігання та синхронізації даних у реальному часі (рис. 3.12). Вона підтримує структуру даних у вигляді документів, які організовані у колекції [7].



Рисунок 3.12 – Логотип Firebase

Firestore використовувався для зберігання та синхронізації даних, що потребують доступу у реальному часі, таких як інформація про користувачів, замовлення та товари. Ця технологія забезпечила надійне зберігання даних та їх миттєву доступність для всіх користувачів застосунку.

Основні переваги використання Firestore включають:

- забезпечення миттєвого оновлення даних між клієнтами;
- підтримка великого обсягу даних та високе навантаження;
- інтегрування з Firebase Authentication для забезпечення безпеки даних.

#### *Room*

Room – це бібліотека з Android Jetpack, яка надає зручний та ефективний спосіб роботи з локальними реляційними базами даних. Вона забезпечує абстракцію над SQLite і дозволяє уникнути написання великої кількості шаблонного коду.

Room використовувався для локального зберігання даних на пристрої користувача, таких як кешовані дані про товари, обрані елементи та налаштування

користувача. Використання Room дозволило забезпечити швидкий доступ до даних без необхідності постійного звернення до серверу.

Основні переваги використання Room включають:

- зручний API для взаємодії з базою даних;
- забезпечення автоматичного оновлення UI при зміні даних;
- забезпечення перевірки коректності SQL-запитів під час компіляції;
- інтеграція Firebase Firestore та Room.

Використання комбінації Firebase Firestore та Room дозволило створити гібридну систему зберігання даних, яка поєднує переваги хмарного та локального зберігання. Це забезпечило синхронізацію у реальному часі для критичних даних за допомогою Firestore та локальне кешування та офлайн-доступ для забезпечення безперебійної роботи застосунку за допомогою Room.

### 3.2.4 Технологія розпізнавання зображень

У цьому проекті технологія розпізнавання зображень відіграє ключову роль у забезпеченні основного функціоналу мобільного застосунку для інтернет-магазину. Для реалізації цієї технології було використано TensorFlow Lite у поєднанні з моделлю, створеною за допомогою Teachable Machine.

#### *TensorFlow Lite*

TensorFlow Lite – це полегшена версія TensorFlow, оптимізована для роботи на мобільних та вбудованих пристроях (рис. 3.13). Вона дозволяє виконувати машинне навчання безпосередньо на пристрої користувача, забезпечуючи швидкість і ефективність обробки [8].



Рисунок 3.13 – Логотип TensorFlow Lite

### Основні переваги TensorFlow Lite:

- забезпечує високу продуктивність на мобільних пристроях завдяки оптимізованим моделям і алгоритмам;
- споживає менше ресурсів у порівнянні з повною версією TensorFlow, що дозволяє працювати на пристроях з обмеженими обчислювальними можливостями;
- підтримує різні платформи, включаючи Android, iOS та мікроконтролери, що робить її універсальним рішенням для розробки мобільних застосунків;
- дозволяє використовувати попередньо натреновані моделі або створювати власні, що забезпечує гнучкість у виборі та налаштуванні моделей машинного навчання.

### *Teachable Machine*

Teachable Machine – це інструмент від Google, який дозволяє легко створювати моделі машинного навчання без необхідності писати код.

Використання Teachable Machine для створення моделей розпізнавання зображень має кілька ключових переваг:

- інтуїтивно зрозумілий інтерфейс дозволяє створювати моделі навіть користувачам без глибоких знань у галузі машинного навчання;
- процес створення моделі займає мінімальний час, що дозволяє швидко експериментувати з різними варіантами;
- користувачі можуть легко завантажувати власні дані для навчання моделі, що забезпечує високу точність і адаптивність до конкретних задач;
- дозволяє експортувати створені моделі у формат TensorFlow Lite, що спрощує інтеграцію з мобільними застосунками.

Використання TensorFlow Lite у поєднанні з моделлю, створеною за допомогою Teachable Machine, дозволило забезпечити точне та ефективне розпізнавання зображень у мобільному застосунку, що підвищує його функціональність і зручність для користувачів.

### **Висновки до розділу 3**

У третьому розділі було детально розглянуто та проаналізовано основні аспекти розробки мобільного застосунку інтернет-магазину з використанням технології розпізнавання зображень.

Створено різні типи UML-діаграм, які допомогли візуалізувати архітектуру застосунку, його компоненти та взаємодії між ними. Використання діаграм класів, компонентів, станів та переходів дозволило чітко визначити структуру та логіку роботи системи.

Проаналізовано основні технології, використані для розробки застосунку. Зокрема, обрана мова програмування Kotlin, яка забезпечує високий рівень безпеки, сумісність з Java та лаконічність коду. Розглянуто технології front-end розробки, включаючи Android SDK, Jetpack Compose, Material Design, ViewModel, LiveData, Navigation Component та DataBinding.

Описано використання TensorFlow Lite для реалізації функціоналу розпізнавання зображень. Створення моделей за допомогою Teachable Machine забезпечило простоту та швидкість навчання моделей, а також їхню інтеграцію в мобільний застосунок.

## 4 ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ЗАСТОСУНКУ ТА ПРОВЕДЕННЯ ТЕСТУВАННЯ

Мобільний застосунок містить кілька основних екранів, зокрема: головний, каталог товарів, кошик, улюблені товари, налаштування та інші. Основним користувачем системи є клієнт, який має доступ до функціональних можливостей застосунку.

### 4.1 Огляд дизайну мобільного застосунку

#### *Екран авторизації*

Екран авторизації (рис. 4.1) є одним з ключових екранів мобільного застосунку інтернет-магазину. Він дозволяє користувачам увійти до системи, використовуючи свій обліковий запис, або зареєструватися за допомогою сторонніх сервісів.

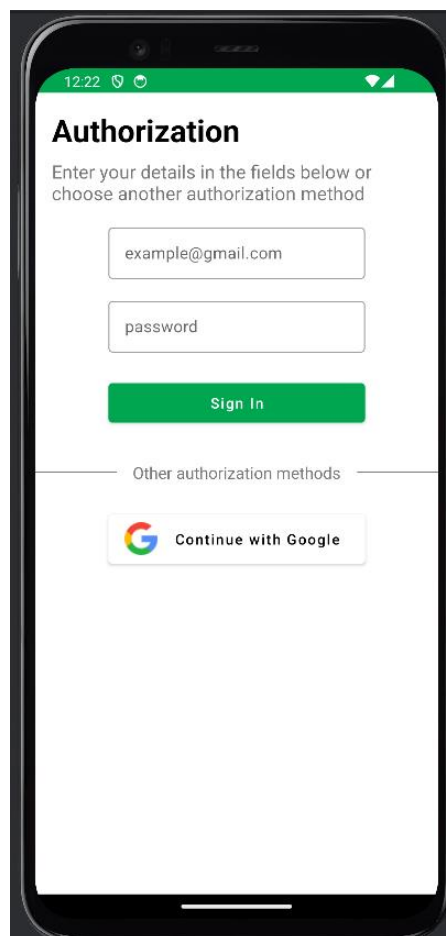


Рисунок 4.1 – Екран авторизації

Дизайн цього екрану був розроблений з урахуванням принципів зручності та мінімалізму, що забезпечує простоту використання і швидкий доступ до функціоналу застосунку.

### ***Головний екран***

Головний екран (рис. 4.1) мобільного застосунку інтернет-магазину з використанням технології розпізнавання зображень забезпечує користувачам швидкий доступ до основних категорій товарів, популярних продуктів та спеціальних пропозицій.

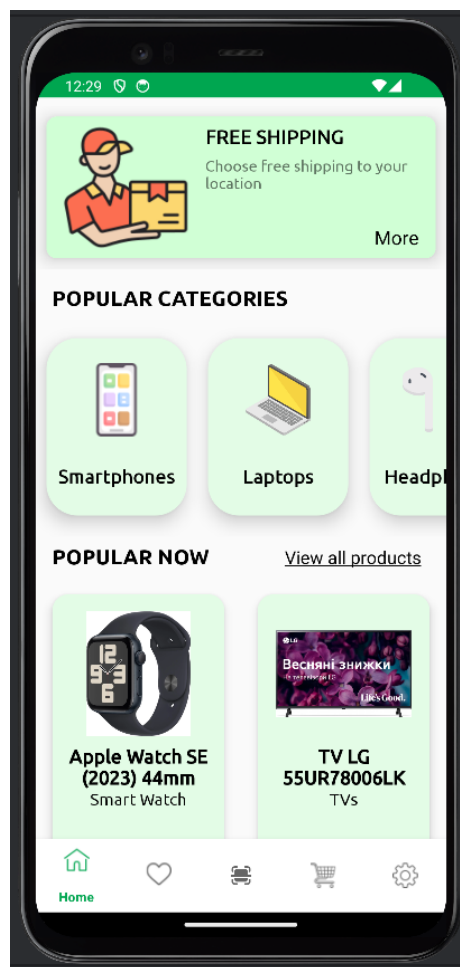


Рисунок 4.2 – Головний екран

Дизайн головного екрану було розроблено з урахуванням зручності та інтуїтивності, щоб користувачі могли легко знайти необхідні товари та ознайомитись із актуальними акціями.

### ***Екран з переліком товарів***

Екран з товарами (рис. 4.3) мобільного застосунку інтернет-магазину відображає перелік доступних товарів, дозволяючи користувачам швидко переглядати і знаходити необхідні продукти.

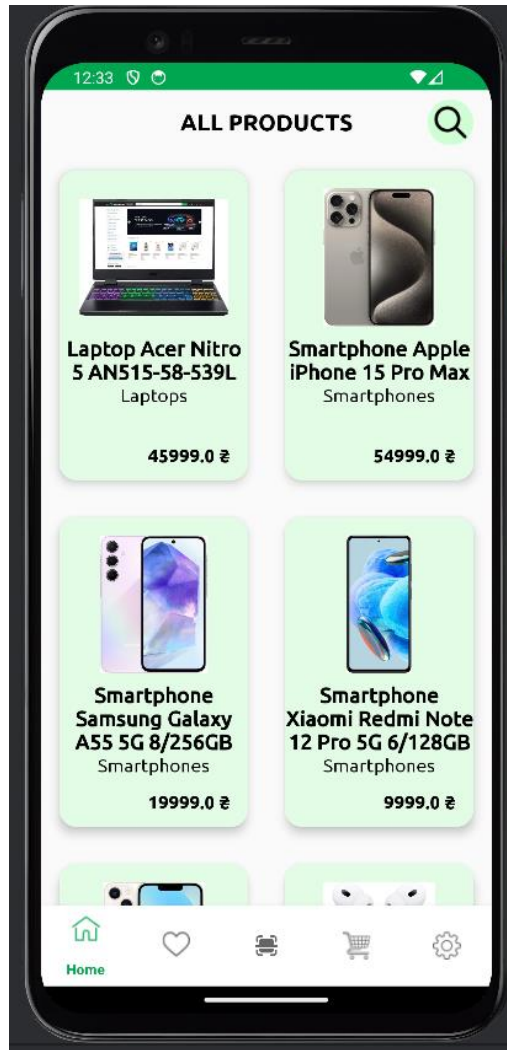


Рисунок 4.3 – Екран з переліком товарів

Дизайн цього екрану забезпечує зручну навігацію та доступ до інформації про товари.

### ***Екран з улюбленими товарами***

Екран з улюбленими товарами (рис. 4.4) мобільного застосунку інтернет-магазину дозволяє користувачам переглядати та управляти списком товарів, які вони відзначили як улюблені.





Рисунок 4.4 – Екран з улюбленими товарами

Цей екран забезпечує швидкий доступ до улюблених продуктів, що спрощує процес здійснення покупок.

#### ***Екран з детальною інформацією про товар***

Екран з детальною інформацією про товар (рис. 4.5) в мобільному застосунку інтернет-магазину надає користувачам повну інформацію про обраний товар.

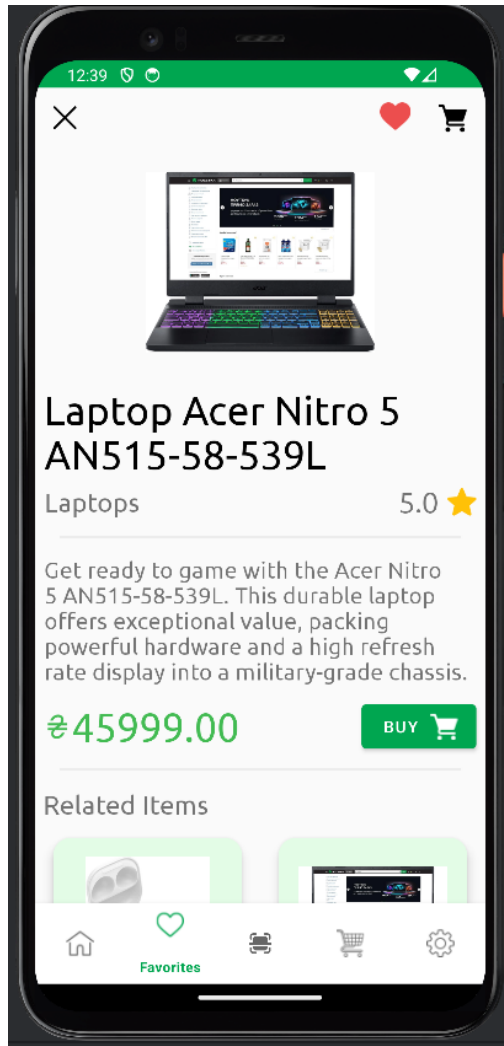


Рисунок 4.5 – Екран з детальною інформацією про товар

Дизайн цього екрану забезпечує зручний доступ до всіх необхідних деталей, що допомагає користувачам прийняти рішення щодо покупки.

### ***Екран розпізнавання зображень***

Екран розпізнавання зображень (рис. 4.6) у мобільному застосунку інтернет-магазину дозволяє користувачам використовувати камеру свого пристрою для ідентифікації продуктів.

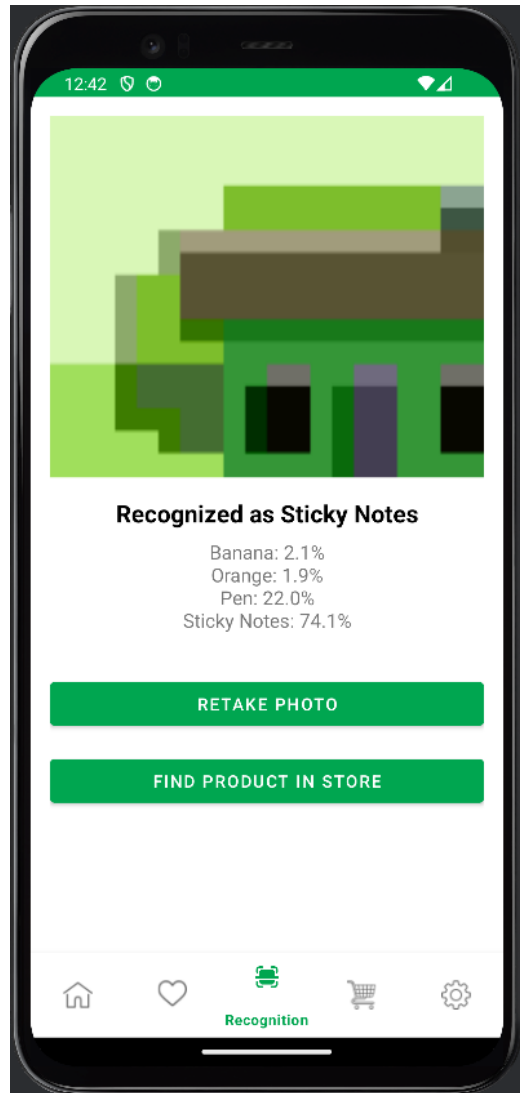


Рисунок 4.6 – Екран розпізнавання зображень

Дизайн цього екрану забезпечує простий і зрозумілий процес розпізнавання зображень, відображаючи результати та надаючи можливість подальших дій.

### ***Екран пошуку***

Екран пошуку (рис. 4.7) мобільного застосунку інтернет-магазину надає користувачам можливість швидко знаходити необхідні товари, використовуючи ключові слова.

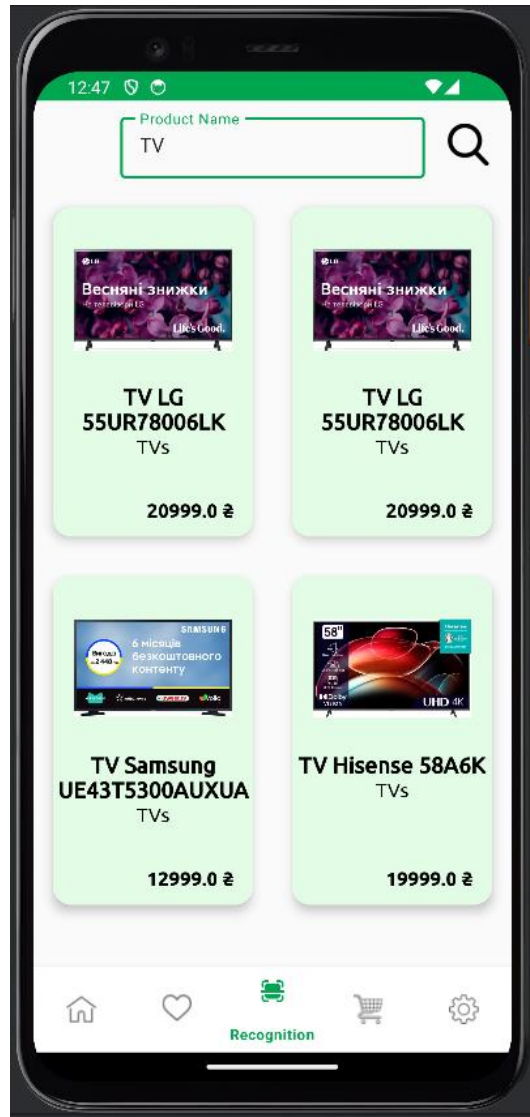


Рисунок 4.7 – Екран пошуку

Дизайн цього екрану забезпечує зручний і ефективний процес пошуку, відображаючи результати у зрозумілому форматі.

### ***Екран кошика***

Екран кошика (рис. 4.8) в мобільному застосунку інтернет-магазину надає користувачам можливість переглянути та керувати товарами, які вони додали до свого кошика для подальшого оформлення замовлення.

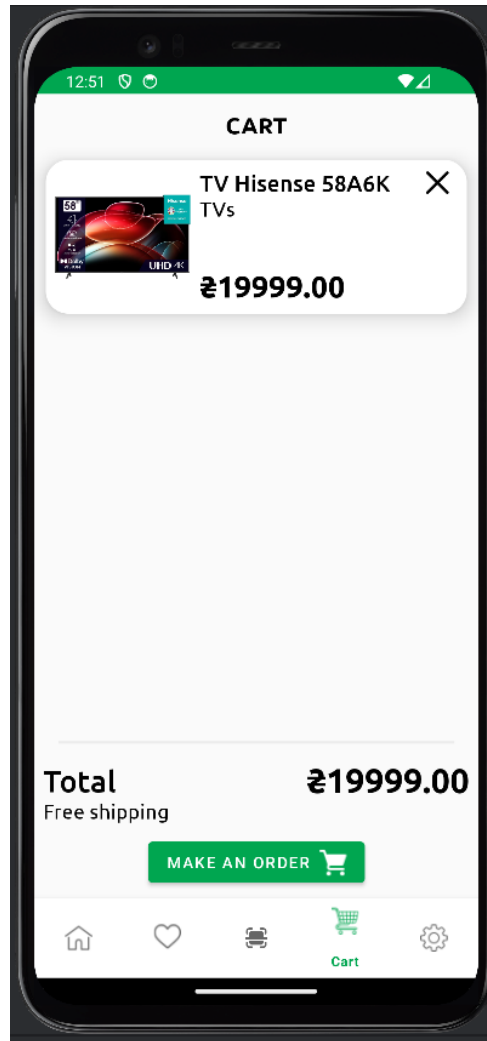


Рисунок 4.8 – Екран пошуку

Дизайн цього екрану забезпечує зручний і простий доступ до інформації про товари в кошику, дозволяючи користувачам легко завершити покупку.

#### **4.2 Реалізація програмного компоненту розпізнавання зображень**

У даному підрозділі розглянуто реалізацію програмного компоненту розпізнавання зображень у мобільному застосунку інтернет-магазину. Цей компонент дозволяє користувачам використовувати камеру свого пристрою для зйомки фотографій товарів, після чого застосунок виконує аналіз зображення та визначає, до якої категорії належить об'єкт.

Для реалізації функціоналу розпізнавання зображень використовувалась бібліотека TensorFlow Lite, яка забезпечує можливість виконання машинного навчання на мобільних пристроях.

### ***Ініціалізація компонентів***

На початку відбувається ініціалізація необхідних UI-компонентів у методі onCreateView(). Серед них: TextView для відображення результатів розпізнавання і рівня впевненості, ImageView для відображення знятого зображення, а також кнопки для зйомки фотографії та пошуку товару.

```
class RecognitionFragment : Fragment() {
    private lateinit var result: TextView
    private lateinit var confidence: TextView
    private lateinit var imageView: ImageView
    private lateinit var picture: Button
    private lateinit var btn: Button
    private val imageSize = 224

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val view = inflater.inflate(R.layout.fragment_recognition,
            container, false)
        result = view.findViewById(R.id.result)
        confidence = view.findViewById(R.id.confidence)
        imageView = view.findViewById(R.id.imageView)
        picture = view.findViewById(R.id.button)
        btn = view.findViewById(R.id.button_find)
        return view
    }
}
```

### ***Обробка подій зйомки фотографії***

При натисканні кнопки для зйомки фотографії перевіряються дозволи на використання камери. Якщо дозволи надано, відкривається камера для зйомки. Після зйомки фотографії відображається зроблене зображення.

```
picture.setOnClickListener {
    if (ContextCompat.checkSelfPermission(
        requireContext(),
        Manifest.permission.CAMERA
    ) == PackageManager.PERMISSION_GRANTED
    ) {
        val cameraIntent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
        startActivityForResult(cameraIntent, 1)
    }
}
```

```
    } else {  
        requestPermissions(arrayOf(Manifest.permission.CAMERA), 100)  
    }  
  
    btn.postDelayed({  
        btn.visibility = View.VISIBLE  
        picture.text = "Retake photo"  
    }, 2000)  
}
```

### ***Класифікація зображення***

При натисканні кнопки для зйомки фотографії перевіряються дозволи на використання камери. Якщо дозволи надано, відкривається камера для зйомки. Після зйомки фотографії відображається зроблене зображення.

```
private fun classifyImage(image: Bitmap) {  
    try {  
        val model = Model.newInstance(requireContext())  
  
        val inputFeature0 = TensorBuffer.createFixedSize(intArrayOf(1, 224,  
224, 3), DataType.FLOAT32)  
        val byteBuffer = ByteBuffer.allocateDirect(4 * imageSize *  
imageSize * 3)  
        byteBuffer.order(ByteOrder.nativeOrder())  
  
        val intValues = IntArray(imageSize * imageSize)  
        image.getPixels(intValues, 0, image.width, 0, 0, image.width,  
image.height)  
  
        var pixel = 0  
        for (i in 0 until imageSize) {  
            for (j in 0 until imageSize) {  
                val value = intValues[pixel++] // RGB  
                byteBuffer.putFloat(((value shr 16) and 0xFF) * (1f /  
255f))  
  
                byteBuffer.putFloat(((value shr 8) and 0xFF) * (1f / 255f))  
                byteBuffer.putFloat((value and 0xFF) * (1f / 255f))  
            }  
        }  
  
        inputFeature0.loadBuffer(byteBuffer)  
  
        val outputs = model.process(inputFeature0)  
        val outputFeature0 = outputs.outputFeature0AsTensorBuffer  
  
        val confidences = outputFeature0.floatArray  
        var maxPos = 0  
        var maxConfidence = 0f  
        for (i in confidences.indices) {  
            if (confidences[i] > maxConfidence) {  
                maxConfidence = confidences[i]  
                maxPos = i  
            }  
        }  
    }  
}
```

```

    }
    val classes = arrayOf("USB", "Lightning", "HDMI", "TV",
"Smartphone", "Game Console")
    result.text = "Recognized as ${classes[maxPos]}"

    btn.setOnClickListener{
        findNavController().navigate(
RecognitionFragmentDirections.actionRecognitionFragmentToSearchFragment(classes[maxPos])
        )
    }

    val s = StringBuilder()
    for (i in classes.indices) {
        s.append(String.format("%s: %.1f%%\n", classes[i],
confidences[i] * 100))
    }
    confidence.text = s.toString()

    model.close()
} catch (e: IOException) {
    e.printStackTrace()
}
}

```

### ***Обробка результатів зйомки***

При натисканні кнопки для зйомки фотографії перевіряються дозволи на використання камери. Якщо дозволи надано, відкривається камера для зйомки. Після зйомки фотографії відображається зроблене зображення.

```

override fun onActivityResult(requestCode: Int, resultCode: Int, data:
Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (requestCode == 1 && resultCode == AppCompatActivity.RESULT_OK) {
        val image = data?.extras?.get("data") as Bitmap
        val dimension = minOf(image.width, image.height)
        var thumbnail = ThumbnailUtils.extractThumbnail(image, dimension,
dimension)
        imageView.setImageBitmap(thumbnail)

        thumbnail = Bitmap.createScaledBitmap(thumbnail, imageSize,
imageSize, false)
        classifyImage(thumbnail)
    }
}

```

### ***Створення моделі TensorFlow Lite***

Для створення моделі розпізнавання зображень було використано сервіс ТМ від Google. Цей сервіс дозволяє легко і швидко створювати моделі машинного



навчання, використовуючи простий і зрозумілий інтерфейс. Нижче представлено процес навчання моделі.

У процесі навчання моделі (рис. 4.9) було використано кілька категорій зображень: навушники, годинники та смартфони. Кожна категорія містила певну кількість зразків зображень, які були завантажені або зроблені за допомогою веб-камери. Після завантаження всіх зразків було розпочато навчання моделі.

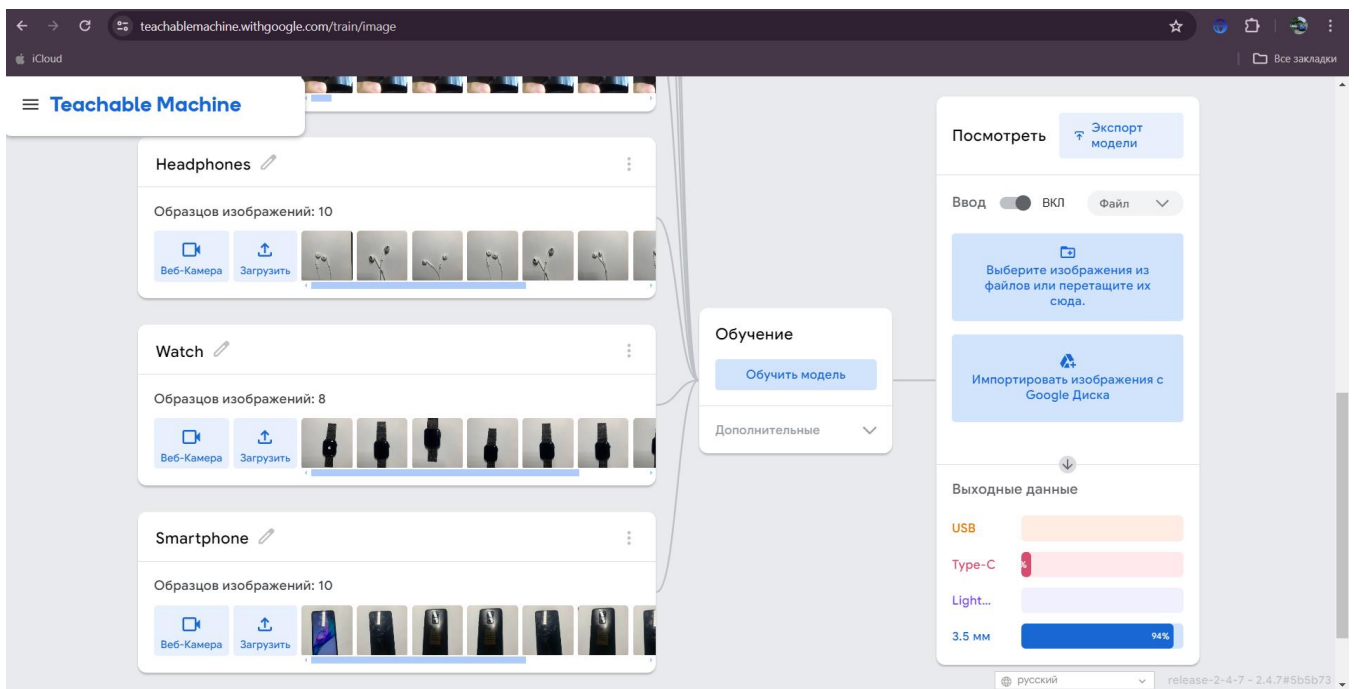


Рисунок 4.9 – Процес навчання моделі

Після успішного завершення навчання модель була експортована у формат TFL, що дозволило інтегрувати її у мобільний застосунок. Це забезпечило можливість виконання розпізнавання зображень безпосередньо на пристрої користувача, що підвищило швидкість обробки та безпеку даних.

Ця реалізація програмного компоненту розпізнавання зображень демонструє, як сучасні технології машинного навчання можуть бути інтегровані в мобільні застосунки для покращення користувацького досвіду.

### 4.3 Тестування мобільного застосунку

Для забезпечення високої якості та зручності використання мобільного застосунку, перед його релізом необхідно провести всебічне тестування. У разі виявлення проблем або багів їх слід виправити до того, як застосунок буде надано користувачам. Результати тестування включають успішні сценарії виконання та розширення, а також опис ситуацій, в яких система виводить попередження про помилки. Результати тестування включають успішні сценарії виконання та розширення, а також опис ситуацій, в яких система виводить попередження про помилки. Всі ці результати представлені в таблицях 4.1 – 4.3.

Таблиця 4.1 – Реєстрація в мобільному застосунку

<b>Діючі особи</b>	Неzareєстрований користувач, система
<b>Мета</b>	Створення облікового запису
<b>Передумова</b>	Користувач не zareєстрований в системі
<p><b>Успішний сценарій:</b></p> <ol style="list-style-type: none"> <li>Користувач переходить до форми реєстрації.</li> <li>Користувач заповнює необхідні дані (ім'я, прізвище, e-mail, статус, отримання розсилки).</li> <li>Користувач надсилає дані системі.</li> <li>Система обробляє обліковий запис та надсилає лист із паролем для входу та посиланням на підтвердження пошти на вказану електронну адресу.</li> <li>Користувач переходить за посиланням.</li> <li>Система зберігає обліковий запис.</li> <li>Система переводить користувача на головну сторінку.</li> </ol>	
<b>Результат:</b> Сценарій успішний. Збережено новий обліковий запис.	
<b>Розширення</b>	
<b>1a</b>	Не всі поля заповнені. Система виводить відповідне повідомлення під кожним незаповненим полем. Результат: Користувач незареєстрований.

Кінець таблиці 4.1

<b>2а</b>	Користувач вводить вже існуючі дані. Система виводить відповідне повідомлення. Результат: Користувач незареєстрований.
<b>Висновок:</b> Усі сценарії розширення успішно виконані.	

Таблиця 4.2 – Авторизація в мобільному застосунку

<b>Діючі особи</b>	Незареєстрований користувач, система
<b>Мета</b>	Виконати авторизацію
<b>Передумова</b>	Користувач не авторизований в системі
<b>Успішний сценарій:</b>	
<ol style="list-style-type: none"> <li>1. Користувач переходить на сторінку авторизації.</li> <li>2. Користувач заповнює необхідні дані (пошту, пароль).</li> <li>3. За бажанням користувач відмічає функцію «Запам'ятати мене».</li> <li>4. Система валідує дані у базі даних.</li> <li>5. Система переводить користувача на головну сторінку.</li> </ol>	
<b>Результат:</b> Сценарій успішний. Збережено новий обліковий запис.	
<b>Розширення</b>	
<b>1а</b>	Не всі поля заповнені. Система виводить відповідне повідомлення під кожним незаповненим полем. Результат: Користувач незареєстрований.
<b>2а</b>	Не існує такого користувача або невірний пароль. Система виводить відповідне повідомлення. Результат: Користувач не авторизований.
<b>Висновок:</b> Усі сценарії розширення успішно виконані.	

Таблиця 4.3 – Здійснення пошуку товарів за допомогою розпізнавання зображень

<b>Діючі особи</b>	Авторизований користувач, система
<b>Мета</b>	Виконати пошук товарів за допомогою розпізнавання зображень
<b>Передумова</b>	Користувач авторизований в системі та надав дозвіл до камери.

### Кінець таблиці 4.3

<b>Успішний сценарій:</b> <ol style="list-style-type: none"><li>1. Користувач переходить на сторінку розпізнавання зображень.</li><li>2. Користувач активує камеру та робить фотографію товару.</li><li>3. Система обробляє зображення та виконує розпізнавання.</li><li>4. Система відображає результат розпізнавання, показуючи категорію та можливі варіанти товарів.</li><li>5. Користувач вибирає опцію "Знайти товар у магазині".</li><li>6. Система відображає список товарів, що відповідають результатам розпізнавання.</li></ol>	
<b>Результат:</b> Сценарій успішний. Пошук товарів виконано.	
<b>Розширення</b>	
<b>1a</b>	Камера недоступна або не має дозволу. Система виводить повідомлення про необхідність надати доступ до камери. Результат: Користувач не може здійснити пошук за допомогою розпізнавання зображень.
<b>2a</b>	Зображення не розпізнано. Система виводить повідомлення про неможливість розпізнати об'єкт на зображенні. Результат: Користувач не отримує результатів пошуку.
<b>Висновок:</b> Усі сценарії розширення успішно виконані.	

Проведено тестування трьох основних сценаріїв використання мобільного застосунку: реєстрація, авторизація та здійснення пошуку товарів за допомогою розпізнавання зображень. Усі основні та розширені сценарії успішно виконані, що підтвердило правильність роботи функціоналу застосунку.

### Висновки до розділу 4

У четвертому розділі було розглянуто програмну реалізацію та тестування мобільного застосунку інтернет-магазину з використанням технології розпізнавання зображень. Основні аспекти, які були висвітлені:

Представлено основні екрани застосунку, такі як екран авторизації, головний екран, екран з товарами, екран з улюбленими товарами, екран з детальною інформацією про товар, екран розпізнавання зображень та екран пошуку товарів. Дизайн кожного екрану забезпечує зручний та інтуїтивно зрозумілий інтерфейс для користувачів.

Детально описано процес реалізації функціоналу розпізнавання зображень за допомогою бібліотеки TensorFlow Lite. Включено ключові фрагменти коду, які відповідають за зйомку фотографії, обробку зображення, класифікацію об'єктів та відображення результатів розпізнавання. Використання машинного навчання дозволило реалізувати ефективний інструмент для ідентифікації товарів за допомогою камери.

Проведено всебічне тестування застосунку для забезпечення його якості та зручності використання. Тестування включало перевірку різних сценаріїв, таких як реєстрація користувачів, авторизація, та пошук товарів за допомогою розпізнавання зображень. Успішно виконано всі сценарії, включаючи розширення, які передбачають обробку помилкових ситуацій.

## ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра автоматизовано процес покупки товарів за допомогою розробки мобільного застосунку для інтернет-магазину з використанням ТРЗ.

Для досягнення поставленої мети виконано наступні завдання:

- проведено аналіз предметної області та існуючих аналогів мобільних застосунків для інтернет-магазинів;
- визначено основні проблеми та знайдено шляхи їх вирішення;
- сформульовано функціональні вимоги до застосунку;
- розроблено блок-схеми алгоритмів роботи застосунку та необхідні UML-діаграми;
- створено дизайн інтерфейсу користувача та макети основних екранів застосунку;
- зібрано та проаналізовано дані для методичної частини;
- розроблено front-end частину мобільного застосунку за допомогою Kotlin;
- реалізовано back-end частину мобільного застосунку з використанням Firebase для авторизації та бази даних.

Проведене дослідження предметної області підтвердило актуальність автоматизації процесу покупки товарів за допомогою ТРЗ. Аналіз аналогічних застосунків дозволив визначити їх переваги та недоліки, що стало основою для формування вимог до розроблюваного ПЗ. У результаті складено специфікацію вимог до мобільного застосунку та ТЗ.

Для моделювання та проектування системи було створено сім видів UML-діаграм, які наочно демонструють роботу та структуру застосунку.

Результатом проведеної роботи є мобільний застосунок інтернет-магазину з інтегрованою технологією розпізнавання зображень, який забезпечує зручний та ефективний процес покупки товарів для користувачів.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Mobile E-commerce Trends. URL: <https://www.shopify.com/blog/mobile-commerce-trends> (дата звернення: 05.05.2024).
2. How to Design a Mobile App. URL: <https://www.creativebloq.com/how-to-design-a-mobile-app> (дата звернення: 05.05.2024).
3. Android Architecture Components. URL: <https://developer.android.com/jetpack/guide> (дата звернення: 05.05.2024).
4. UML Class Diagrams. URL: <https://www.uml-diagrams.org/class-diagrams.html> (дата звернення: 05.05.2024).
5. Mobile Application Architecture. URL: <https://www.altexsoft.com/blog/mobile-application-architecture/> (дата звернення: 05.05.2024).
6. Best Practices for Mobile App Development. URL: <https://www.techrepublic.com/article/best-practices-for-mobile-app-development/> (дата звернення: 05.05.2024).
7. Firebase for Android. URL: <https://firebase.google.com/docs/android/setup> (дата звернення: 05.05.2024).
8. Implementing Image Recognition with TensorFlow. URL: [https://www.tensorflow.org/lite/models/image\\_classification/overview](https://www.tensorflow.org/lite/models/image_classification/overview) (дата звернення: 05.05.2024).
9. Kotlin programming language. URL: <https://kotlinlang.org/docs/home.html> (дата звернення: 29.05.2024).
10. Package diagram. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-package-diagram/> (дата звернення: 29.05.2024).
11. Component diagram. URL: <https://www.lucidchart.com/pages/uml-component-diagram> (дата звернення: 29.05.2024).
12. Deployment diagram. URL: <https://www.lucidchart.com/pages/uml-deployment-diagram> (дата звернення: 29.05.2024).