

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

_____ проф. Приходько С.Б.

«___» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

на тему: Розробка мобільного застосунку "eTrick" для продажу аксесуарів для мобільних телефонів

Виконав: студент групи 4151

 Скакунов Г.О.

(підпис, ПІБ)

Керівник роботи:

доцент кафедри ПЗАС, к.т.н., доцент

(посада, науковий ступень вчене звання)

_____ Макарова Л.М.

(підпис, ПІБ)

Миколаїв – 2023 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
 Кафедра програмного забезпечення автоматизованих систем
 Спеціальність 121 «Інженерія програмного забезпечення»
 Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»
 Гарант освітньої програми
 _____ доц. Макарова Л.М.
 (підпис)
 «__» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студенту Скакунов Георгій Олександрович

1. Тема роботи: Розробка мобільного застосунку "eTrick" для продажу аксесуарів для мобільних телефонів

Керівник роботи: Макарова Лідія Миколаївна
 Затверджені наказом ректора №257-уч від «17» березня 2023 р.

2. Термін подання роботи: 29.05.2023 р.

3. Вихідні дані по роботі:

4. Перелік питань, що належать до розробки (найменування розділів)

Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Аналіз практичної задачі інженерії програмного забезпечення; Проект програмного забезпечення; Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів

Титульний слайд, Мета та завдання кваліфікаційної роботи, Аналіз існуючих рішень, Аналіз вимог, Діаграма варіантів використання, Технічний проект, Концептуальна модель даних, Інструменти для розробки мобільного застосунку "eTrick" для продажу мобільних аксесуарів, Реалізація та тестування мобільного застосунку, Результати розробки мобільного застосунку, Висновки, Заключний слайд.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 20.03.2023 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	24.03.2023	ПП*
2.	Аналіз практичної задачі інженерії ПЗ	31.03.2023	ПП*
3.	Аналіз вимог до ПЗ	07.04.2023	ПП*
4.	Розробка архітектури ПЗ	21.04.2023	
5.	Розробка проекту ПЗ	05.05.2023	
6.	Реалізація та тестування ПЗ	12.05.2023	
7.	Підготовка розділу Результати розробки ПЗ	17.05.2023	
8.	Підготовка розділу з охорони праці	19.05.2023	ПП*
9.	Підготовка розділу ВИСНОВКИ	24.05.2023	
10.	Оформлення списку використаних джерел та додатків	26.05.2023	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	29.05.2023	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
12.	Підготовка презентації та доповіді	02.06.2023	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	05.06.2023	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді, а також Авторського договору з додатком	12.06.2023	

* - за результатами переддипломної практики (ПП), яка була з 01.02.2023 до 19.03.2023 р.

Студент


(підпис)

Скакунов Г.О.

(ПІБ)

Керівник роботи

(підпис)

Макарова Л.М.

(ПІБ)

АНОТАЦІЯ

Кваліфікаційна робота присвячена вирішенню практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробці мобільного застосунку "eTrick" для продажу аксесуарів для мобільних телефонів.

Об'єктом даної роботи є процес розробки мобільного застосунку "eTrick" для продажу аксесуарів для мобільних телефонів. Проведено аналіз практичної задачі інженерії програмного забезпечення та проаналізовано існуючі програмні продукти та методи розробки мобільних застосунків. Було розроблено мобільний застосунок "eTrick" для пошуку та придбання мобільних аксесуарів та представлено частину коду. Проведено опис аспектів охорони праці при роботі з мобільними телефонами.

Кваліфікаційна робота викладена на 96 сторінках друкованого тексту, містить 32 рисунка, 9 таблиць, 5 додатків та список використаних джерел із 13 найменувань.

ABSTRACT

The qualification work is devoted to the analysis of a practical software engineering problem characterized by complexity and uncertainty of conditions, namely the development of the mobile application "eTrick" for the sale of accessories for mobile phones.

The object of this work is the process of developing a mobile application "eTrick" for the sale of accessories for mobile phones. An analysis of the subject area was carried out and existing software products and methods for developing mobile applications were analyzed. The mobile application "eTrick" for searching and purchasing mobile accessories was developed and part of the code was presented. The aspects of labor protection when working with mobile phones are described.

The qualification work is presented on 96 pages of printed text, contains 32 figures, 9 tables, 5 appendices and a list of references of 13 titles.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ МОБІЛЬНОГО ЗАСТОСУНКУ "eTrick"	9
1.1 Аналіз практичної задачі інженерії програмного забезпечення	9
1.2 Аналіз існуючих аналогів та обґрунтування необхідності розробки	11
1.3 Постановка задачі на розробку мобільного застосунку "eTrick" для продажу мобільних аксесуарів	14
2 РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ "eTrick" ДЛЯ ПРОДАЖУ МОБІЛЬНИХ АКСЕСУАРІВ.....	18
2.1 Аналіз вимог до мобільного застосунку "eTrick" для продажу мобільних аксесуарів	18
2.1.1 Побудова моделі варіантів використання мобільного застосунку "eTrick" для продажу мобільних аксесуарів	18
2.1.2 Специфікації варіантів використання мобільного застосунку "eTrick" для продажу мобільних аксесуарів	20
2.2 Розробка архітектури мобільного застосунку "eTrick" для продажу мобільних аксесуарів	25
2.2.1 Розробка діаграми компонентів мобільного застосунку "eTrick" для продажу мобільних аксесуарів.....	26
2.2.2 Розробка діаграми розміщення мобільного застосунку "eTrick" для продажу мобільних аксесуарів.....	28
2.3 Проект мобільного застосунку "eTrick" для продажу мобільних аксесуарів	29
2.3.1 Ескізний проект мобільного застосунку "eTrick" для продажу мобільних аксесуарів	30

2.3.2 Технічний проект мобільного застосунку "eTrick" для продажу мобільних аксесуарів	35
2.3.3 Робочий проект мобільного застосунку "eTrick" для продажу мобільних аксесуарів	43
3 РЕЗУЛЬТАТИ РОЗРОБКИ МОБІЛЬНОГО ЗАСТОСУНКУ "eTrick" ДЛЯ ПРОДАЖУ МОБІЛЬНИХ АКСЕСУАРІВ	51
4 ОХОРОНА ПРАЦІ	56
4.1 Аналіз шкідливих та небезпечних факторів під час роботи з мобільними пристроями	57
4.2 Методи нейтралізації або зменшення впливу негативних факторів під час роботи зі мобільними пристроями	59
ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63
ДОДАТОК А – Технічне завдання на розробку мобільного застосунку "eTrick" для продажу мобільних аксесуарів.....	65
ДОДАТОК Б – Вихідні тексти програмних модулів	71
ДОДАТОК В – Опис мобільного застосунку "eTrick" для продажу мобільних аксесуарів	88
ДОДАТОК Г – Інструкція користувача мобільного застосунку "eTrick" для продажу мобільних аксесуарів	92
ДОДАТОК Д – Програма та методика випробувань мобільного застосунку "eTrick" для продажу мобільних аксесуарів.....	93

ВСТУП

У сучасному світі мобільні телефони стали невід'ємною частиною нашого життя. Розмаїтість моделей та функцій мобільних пристроїв зробила їх незамінними помічниками в різних аспектах нашого повсякденного і професійного життя. Разом зі зростанням популярності мобільних телефонів збільшується і попит на аксесуари, що покращують їх функціональність, забезпечують комфорт та стиль.

Однак, існуючі ринкові рішення для продажу аксесуарів для мобільних телефонів не завжди задовольняють потреби користувачів. Складність пошуку певного аксесуару, недостатня інформація про продукт, обмежений вибір – це лише деякі з проблем, які зустрічаються при покупках в цій сфері.

Тому метою кваліфікаційної роботи є вирішення практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, розробка мобільного застосунку, який надасть зручну та ефективну платформу для продажу аксесуарів для мобільних телефонів. Використання сучасних інформаційних технологій дозволить досягти цієї мети.

Для досягнення мети при розробці мобільного застосунку "eTrick", необхідно виконати декілька завдань та реалізувати функціонал застосунку, котрий дає змогу ефективно і правильно обрати товари, які повністю задовільняють покупця:

1. Аналіз ринку аксесуарів для мобільних телефонів:

- Аналіз існуючих рішень на ринку.
- Виявлення недоліків та пропозицій для поліпшення існуючих рішень.

2. Розробка функціональності та інтерфейсу застосунку:

- Встановлення основних функцій, які повинні бути доступні в застосунку.
- Проектування зручного та інтуїтивно зрозумілого інтерфейсу користувача.

- Врахування ергономічних принципів та найкращих практик розробки мобільних застосунків.

3. Створення системи керування товарами та замовленнями:

- Створення бази даних для зберігання інформації про товари, їх характеристики та наявність.
- Розробка механізму пошуку та фільтрації товарів за різними критеріями.
- Реалізація функціональності додавання товарів до кошика та оформлення замовлення.

4. Тестування та валідація застосунку:

- Виконання різноманітних тестів для перевірки функціональності та стабільності застосунку.
- Валідація застосунку на різних пристроях та платформах для забезпечення сумісності.
- Виявлення та виправлення помилок та недоліків після тестування.

Об'єктом даної роботи є процес розробки мобільного застосунку "eTrick" для продажу аксесуарів для мобільних телефонів.

Результати цієї роботи можуть бути корисними для бізнесу, що займається продажем аксесуарів для мобільних телефонів, а також для користувачів, які шукають зручний та надійний спосіб придбати необхідні аксесуари. Впровадження розробленого застосунку може покращити якість обслуговування, збільшити задоволеність клієнтів та сприяти розвитку електронної комерції в галузі аксесуарів для мобільних телефонів.

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ МОБІЛЬНОГО ЗАСТОСУНКУ "eTrick"

1.1 Аналіз практичної задачі інженерії програмного забезпечення

Аналіз практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробка мобільного застосунку "eTrick" для продажу аксесуарів для мобільних телефонів, включає детальне вивчення різних аспектів та вимог, що пов'язані з розробкою та експлуатацією такого застосунку.

Одним з ключових аспектів є функціональний аналіз, який визначає основні функції та можливості, які повинен надавати мобільний застосунок користувачам. У цьому випадку, застосунок має забезпечувати зручний та інтуїтивний інтерфейс для перегляду, вибору та придбання аксесуарів для мобільних телефонів. Крім того, він повинен мати можливості пошуку, фільтрації та сортування товарів, а також функції для додавання товарів до кошика, здійснення оплати та вибору способу доставки.

Нефункціональний аналіз включає дослідження вимог щодо продуктивності, безпеки та стабільності застосунку. Важливо, щоб застосунок працював швидко та ефективно, забезпечуючи користувачам зручне та безпечне середовище для здійснення покупок. Для цього можуть бути використані різні технології та підходи, які забезпечать оптимальну продуктивність та захист даних користувачів.

Аналіз ринку та конкурентів є ще одним важливим аспектом. Він допомагає визначити особливості ринку аксесуарів для мобільних телефонів, виявити наявні конкурентні застосунки та їхні переваги та недоліки. Це дозволяє розробникам визначити свої унікальні продуктові пропозиції та знайти шляхи для

виокремлення на ринку. Також слід проаналізувати поведінку та потреби цільової аудиторії, щоб зрозуміти їхні вимоги та очікування від мобільного застосунку.

Крім того, аналіз практичної задачі повинен враховувати технічні обмеження та вимоги. Необхідно визначити платформу, на якій буде розроблено застосунок (наприклад, Android або iOS) та встановити мінімальні технічні характеристики, необхідні для коректної роботи застосунку на цій платформі. Також потрібно вивчити можливості та обмеження з використання певних технологій та мов програмування для розробки мобільних застосунків.

У сучасному світі з'явилася велика кількість мов та фреймворків для розробки різноманітних мобільних застосунків. Серед усіх для розробки мобільного застосунку було обрано мову Flutter [1] та хостинг сервіс Firebase [2]. Flutter – це мобільний фреймворк, розроблений Google, для створення високопродуктивних нативних застосунків для Android та iOS. Завдяки єдиній кодовій базі, можливо швидко розробляти застосунки для обох платформ. Використовування реактивної моделі програмування та архітектури на основі віджетів дає змогу легко створювати складні інтерфейси. Головна перевага Flutter полягає в можливості гарячого перезавантаження, яке дозволяє вносити зміни в код і одразу бачити їх в застосунку [3].

Firebase – це хмарний бекенд-сервіс, який надає різноманітні функції для розробки мобільних та веб-застосунків. Дає змогу скористатися базою даних в режимі реального часу, хмарним сховищем та послугами автентифікації для легкої розробки масштабованих, безсерверних застосунків. Firebase має простий інтерфейс, який допомагає скоротити час розробки. Проте, налаштування Firebase за межами готових функцій є досить складним завданням.

Таким чином, використання Flutter разом з Firebase дозволяє створювати мобільні застосунки з високою продуктивністю і широким функціоналом, простою розробкою і масштабованістю.

Загальний аналіз практичної задачі інженерії програмного забезпечення дає розробникам глибше розуміння вимог та викликів, пов'язаних з розробкою мобільного застосунку для продажу аксесуарів для мобільних телефонів. Він

надає основу для подальшого проектування та реалізації застосунку з урахуванням потреб користувачів та вимог ринку.

1.2 Аналіз існуючих аналогів та обґрунтування необхідності розробки

Проведений аналіз конкурентного середовища дозволив з'ясувати поточний стан ринку мобільних аксесуарів та існуючих конкурентів в цій сфері.

Rozetka – це мобільний застосунок, який пропонує широкий вибір товарів українського і міжнародного ринків [4]. Цей застосунок розроблений для платформ iOS та Android і дозволяє користувачам легко здійснювати покупки на rozetka.com.ua, одному з найбільших і найпопулярніших онлайн-магазинів в Україні.

У застосунку Rozetka можливо знайти широкий асортимент товарів, таких як електроніка, побутова техніка, комп'ютери, мобільні пристрої, одяг, косметика, спортивні товари та багато іншого (рисунк 1.1). Кожен товар супроводжується детальним описом, фотографіями, відгуками користувачів і ціною, що дозволяє зробити обґрунтований вибір під час покупки.

Однак Rozetka має деякі мінуси, які можуть зробити його менш придатним для потреб покупців. По-перше, Rozetka розроблений як маркетплейс. Тому дуже велика кількість продавців та величезний і дуже часто повторюючийся асортимент (рисунк 1.2) збиває користувача з пантелику. Також через те, що кожен може продавати свої товари на Rozetka сама компанія не може нести відповідальності за якість товару. Також одним із найбільших мінусів Rozetka є вартість товару. Саме через завищену вартість користувачі часто вдаються до пошуку альтернативних сайтів.

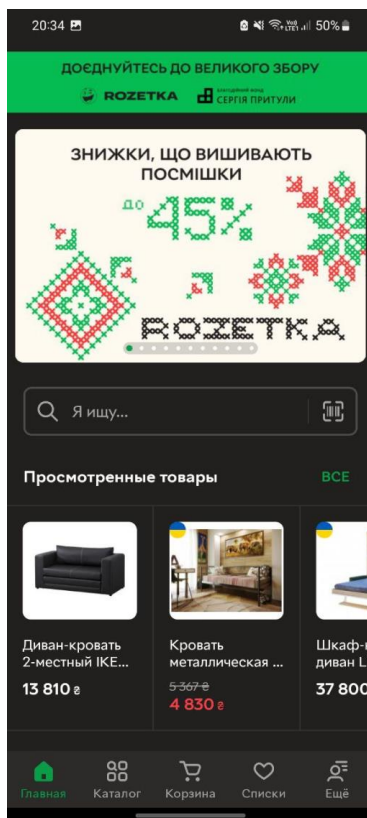


Рисунок 1.1 – Головна сторінка мобільного застосунку Rozetka [4]



Рисунок 1.2 – Сторінка «Каталог» мобільного застосунку Rozetka [4]

ЦИТРУС – це мобільний застосунок, який надає можливість зручно та швидко здійснювати покупки в одному з найбільших онлайн-магазинів електроніки та побутової техніки в Україні [5] (рисунок 1.3). Мобільний застосунок Цитрус пропонує користувачам простий і інтуїтивно зрозумілий інтерфейс, що дозволяє швидко знайти потрібні товари. Є можливість переглядати різні категорії товарів, включаючи смартфони, планшети, комп'ютери, побутову техніку, аксесуари та багато іншого.

Хоча і Цитрус мав спеціалізацією мобільних товарів та аксесуарів, він змінив рід діяльності та розширив асортимент побутової техніки (рисунок 1.4), замість великої кількості товару, котрий належав до категорії мобільних аксесуарів. Також великим мінусом для покупців є завищені ціни у магазинах Цитрус.

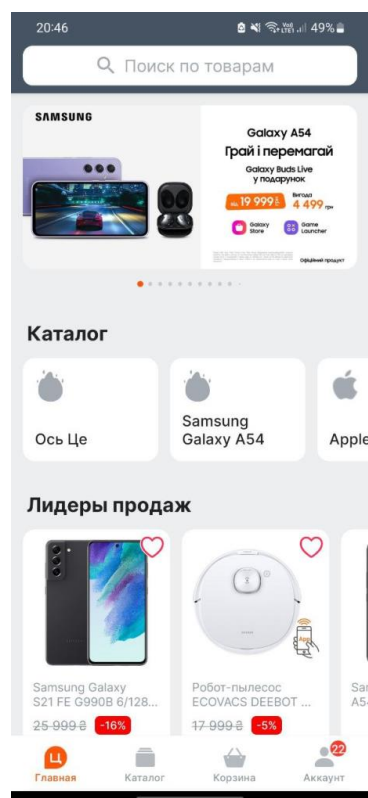


Рисунок 1.3 – головна сторінка мобільного застосунку Цитрус [5]

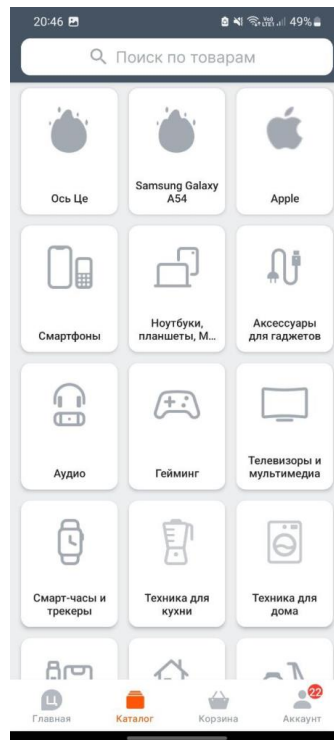


Рисунок 1.4 – Сторінка «Каталог» мобільного застосунку Цитрус [5]

Застосунок "eTrick" вирішує проблему недоступності актуальних даних про товари через нестабільне з'єднання з Інтернетом в Україні. Він дозволяє завантажувати список аксесуарів по категоріях через Wi-Fi, щоб користувачі могли переглядати їх у будь-який зручний час, навіть без доступу до Інтернету.

Це дає перевагу "eTrick" перед іншими застосунками, оскільки вона забезпечує передзавантаження даних і має широкий вибір аксесуарів. Користувачі в Україні зможуть комфортно шукати та купувати мобільні аксесуари, незалежно від доступності Інтернету.

Застосунок "eTrick" полегшує процес покупки аксесуарів та робить його зручним для всіх користувачів, не залежно від обмежень мережі.

1.3 Постановка задачі на розробку мобільного застосунку "eTrick" для продажу мобільних аксесуарів

Проаналізувавши предметну область та проаналізувавши недоліки існуючих аналогів, було вирішено розробити мобільний застосунок "eTrick".

Мета роботи – розробка мобільного застосунку "eTrick" для надання зручної та ефективної платформи для продажу аксесуарів для мобільних телефонів, яка задовольнятиме потреби користувачів і сприятиме розвитку електронної комерції в цій галузі.

Експлуатаційне призначення:

Експлуатаційним призначенням мобільного застосунку є пошук та замовлення аксесуарів для мобільних телефонів.

Функціональне призначення:

Функціональним призначенням мобільного застосунку є покращення та полегшення процесу пошуку та процесу замовлення мобільних аксесуарів.

Користувачі можуть:

- Переглядати список товарів та їх характеристики;
- Отримувати детальну інформацію про будь-який товар представлений у застосунку;
- Збирати кошик товарів та мати можливість їх замовити;
- Мати дані про доставку товару та сповіщення, щодо зміни статусу оброблення замовлення;
- Мати можливість додати товари їх у список «Вподобань» ;
- Мати можливість отримувати сповіщення;
- Мати можливість отримувати інформацію про доставку;

Адміністратор може:

- Редагувати список товарів;
- Перевіряти та видаляти коментарі під товарами;
- Додавати або видаляти фото товарів;
- Редагувати описи будь-яких товарів;

Вимоги до складу виконуваних функцій.

Функції користувача:

- Пошук і фільтрацію продуктів: Користувачі можуть швидко знайти потрібні аксесуари, використовуючи функцію пошуку або застосовуючи фільтри за категорією, брендом, ціною тощо;
- Детальні описи продуктів: Кожен аксесуар має свій власний сторінку з описом, зображеннями, характеристиками, відгуками користувачів та рекомендаціями;
- Кошик і оформлення замовлення: Користувачі можуть додавати продукти до кошика, переглядати замовлення, вибирати способи доставки та оплати, а також використовувати промокоди або купони для отримання знижок;
- Система оглядів і рейтингів: Користувачі можуть залишати відгуки та оцінки для продуктів, що дозволяє іншим покупцям зробити освідчений вибір;
- Оповіщення і відстеження замовлень: Користувачі можуть отримувати сповіщення про стан своїх замовлень, включаючи підтвердження покупки, статус доставки та інформацію про повернення.

Вимоги до організації вхідних та вихідних даних

Вхідні дані:

- Автентифікація або реєстрація користувача: Користувач має надати актуальну електронну адресу та пароль;
- Дані для доставки товару: актуальна адреса або номер поштового відділення, зручний час доставки;
- Дані для оплати товару: номер кредитної або дебетової карти, акаунт у Google або Apple pay;

Вихідні дані:

- Застосунок показує зібрані у кошик товари;
- Застосунок відображає повний і актуальний список бажань;
- Застосунок зберігає Вподобані товари;
- Застосунок надає дані про доставку товару: статус, номер замовлення, сума до сплати, тощо;
- Застосунок відображає відгуки і рейтинг товару;

Визначення вимог для мобільного застосунку

Система повинна задовольняти вказаним вимогам на комп'ютері наступної мінімальної комплекції:

- Процесор серії Exynos 990 Qualcomm Snapdragon 865 або потужніший;
- Оперативна пам'ять 2 ГБ або більше;
- Мінімум дискового простору 100 МБ для встановлення програми;
- Роздільна здатність екрану не менше 1280x800 пікселів;
- Інтернет-з'єднання: Застосунок вимагає доступу до Інтернету для завантаження інформації про продукти, оновлення даних та здійснення покупок. Доступ до Wi-Fi або мобільного Інтернету (3G, 4G, 5G) є необхідним;

Вимоги до програмної документації.

Склад програмної документації повинен включати в себе:

- технічне завдання;
- програму та методику випробувань;
- керівництво користувача;
- опис програми;
- код програми.

Більш докладно постановка задачі сформульована в технічному завданні, яке наведене в Додатку А.

2 РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ "eTrick" ДЛЯ ПРОДАЖУ МОБІЛЬНИХ АКСЕСУАРІВ

2.1 Аналіз вимог до мобільного застосунку "eTrick" для продажу мобільних аксесуарів

2.1.1 Побудова моделі варіантів використання мобільного застосунку "eTrick" для продажу мобільних аксесуарів

В ході аналізу предметної області та формулюванні вимог до мобільного застосунку було виявлено та визначено два актори для взаємодії з мобільним застосунком "eTrick": користувач та адміністратор.

Мобільний застосунок для продажу мобільних аксесуарів розроблений саме для користувачів, надалі їх названо «Користувач». Для редагування, додавання, видалення та для любых інших передбачених функціоналом дій з будь-яким товаром додано спеціальну особу, котру надалі було названо «Адміністратор». Стислий опис дій «Користувача» та «Адміністратора» наведено далі у таблиці 2.1.

Таблиця 2.1 – Актори мобільного застосунку "eTrick" для продажу мобільних аксесуарів

Актор	Короткий опис
Користувач	Особа, котра має можливість переглядати каталог мобільних аксесуарів, оформлює замовлення.
Адміністратор	Особа, котра редагує інформацію про товари та обробляє оформленні замовлення.

Виявлені варіанти використання мобільного застосунку "eTrick" для продажу мобільних аксесуарів наведені у таблиці 2.2.

Таблиця 2.2 – Виявлені варіанти використання мобільного застосунку "eTrick" для продажу мобільних аксесуарів.

Актор	Назва	Короткий опис
Адміністратор	Управління категоріями	Дозволяє адміністратору додавати, видаляти та змінювати категорії товарів.
Адміністратор	Управління товарами	Дозволяє адміністратору додавати, видаляти та змінювати товари в категоріях.
Адміністратор	Управління замовленнями	Дозволяє адміністратору переглядати та керувати замовленнями, включаючи підтвердження, відміну або виконання.
Користувач	Реєстрація	Дозволяє користувачу створити власний аккаунт у застосунку.
Користувач	Пошук товарів	Дозволяє користувачеві шукати аксесуари за різними критеріями, такими як назва, категорія, ціновий діапазон тощо.
Користувач	Перегляд товарів	Дозволяє користувачеві переглядати детальну інформацію про товари, включаючи зображення, опис, ціну та наявність.
Користувач	Додавання товару в кошик	Дозволяє користувачеві додавати обрані товари в кошик для подальшого оформлення замовлення.
Користувач	Оформлення замовлення	Дозволяє користувачеві заповнити необхідну інформацію та оформити замовлення на придбання аксесуарів.
Користувач	Відстеження замовлення	Дозволяє користувачеві відстежувати статус свого замовлення та отримувати інформацію про його доставку.
Користувач	Отримання сповіщень	Дозволяє користувачеві отримувати сповіщення про нові товари, знижки, акції та інші спеціальні пропозиції.

Ці варіанти використання допомагають розуміти, як користувачі будуть взаємодіяти з мобільним застосунком "eTrick" та які функції та можливості повинні бути реалізовані для задоволення їх потреб. Аналіз і уточнення цих варіантів використання допомагає забезпечити ефективну та задовільняючу взаємодію користувачів з додатком. Рисунок 2.1 відображає діаграму варіантів використання для акторів «Користувач» та «Адміністратор» у мобільному застосунку "eTrick" для продажу мобільних аксесуарів.

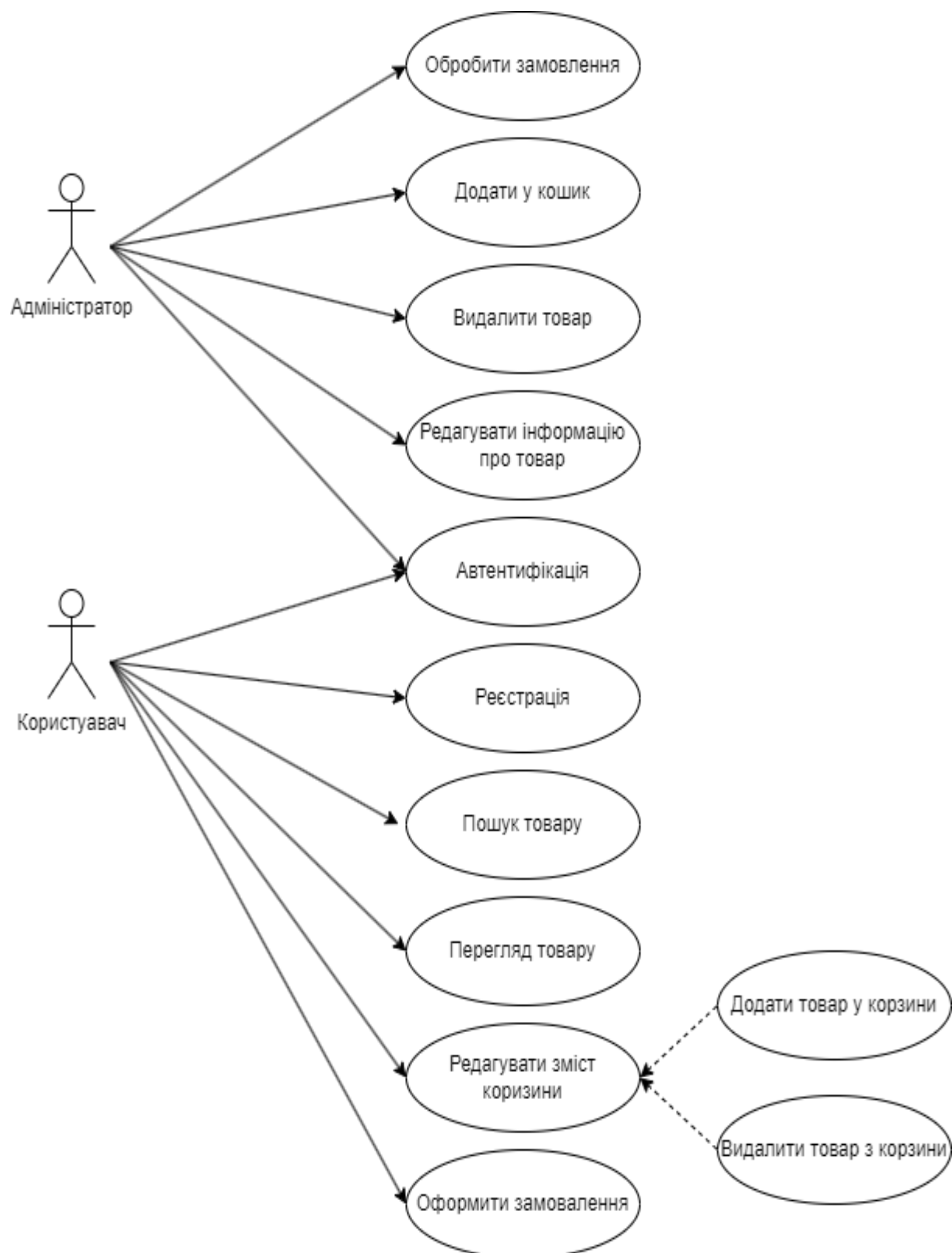


Рисунок 2.1 – Діаграма варіантів використання мобільного застосунку "eTrick" для продажу мобільних аксесуарів

2.1.2 Специфікації варіантів використання мобільного застосунку "eTrick" для продажу мобільних аксесуарів

Специфікації варіантів використання мобільного застосунку "eTrick" для продажу мобільних аксесуарів наведено нижче у таблицях 2.3 – 2.8.

Таблиця 2.3 – Специфікація варіанту використання «Автентифікація»

Складова	Опис
Опис прецеденту	Автентифікація
Дійові особи прецеденту	Користувач, адміністратор
Передумови	Користувач або адміністратор попередньо зареєстровані у системі
Потік подій	Прецедент починається, коли актор відкриває мобільний застосунок
Базовий потік	1. «Користувач» або натискає кнопку «Увійти». 2. Застосунок переадресує актора на сторінку автентифікації. 3. «Користувач» або вводить свій логін та пароль. 4. Система перевіряє ці дані на збіг з даними у базі даних та автентифікує користувача.
Альтернативні потоки	5. «Користувач» або «Адміністратор» вводить свої дані. 6. Система перевіряє ці дані на збіг з даними у базі даних та не знаходить збіг таких даних. 7. Система відображає повідомлення про неправильний логін або пароль.
Постумови	Якщо користувач чи адміністратор були успішно автентифіковані, то відкривається головна сторінка мобільного застосунку, а у віконці вгорі показано автентифікованого користувача.

Таблиця 2.4 – Специфікація варіанту використання «Пошук товару»

Складова	Опис
1	2
Опис прецеденту	Пошук товарів у мобільному застосунку
Дійові особи прецеденту	Користувач, Адміністратор

Продовження таблиці 2.4

1	2
Передумови	Користувач або Адміністратор просто відкривають мобільний застосунок
Базовий потік	1. Користувач або Адміністратор відкриває мобільний застосунок 2. Користувач або Адміністратор натискають кнопку пошуку 3. Користувач або Адміністратор вводять назву товару 4. Користувачу або Адміністратору показує всі товари, корті збігаються зі словами у пошуку
Альтернативні потоки	5. Користувач або Адміністратор відкриває мобільний застосунок 6. Користувач або Адміністратор натискають кнопку пошуку 7. Користувач або Адміністратор вводять назву товару котрого не має у списку наявних товарів 8. Користувачу або Адміністратору показує повідомлення про відсутність шуканого товару
Постумови	Постумові відсутні

Таблиця 2.5 – Специфікація варіанту використання «Редагування інформації про товар»

Складова	Опис
Опис прецеденту	Редагування інформації про товар
Дійові особи прецеденту	Адміністратор
Передумови	Адміністратор повинен увійти в систему.
Базовий потік	1. Адміністратор відкриває мобільний застосунок 2. Адміністратор вибирає товар, котрий він планує редагувати 3. Адміністратор змінює необхідні дані 4. Адміністратор зберігає змінені дані
Альтернативні потоки	1. Адміністратор відкриває мобільний застосунок та вибирає товар, котрий він планує редагувати 2. Адміністратор змінює необхідні дані, проте вони не проходять валідацію та бачить повідомлення про неможливість збереження поточних даних.
Постумови	Повідомлення про успішне збереження нових даних

Таблиця 2.6 – Специфікація варіанту використання «Отримання сповіщень»

Складова	Опис
Опис прецеденту	Отримання сповіщень
Дійові особи прецеденту	Користувач
Передумови	Користувач повинен бути авторизованим у застосунку
Базовий потік	Користувач отримує сповіщення про новий товар, акції, тощо
Постумови	Якщо у користувача увімкнені сповіщення, то він буде бачити інформацію про нові товари, акції, тощо

Таблиця 2.7 – Специфікація варіанту використання «Реєстрація»

Складова	Опис
Опис прецеденту	Реєстрація.
Дійові особи прецеденту	Користувач.
Передумови	Користувач немає аккаунту у застосунку.
Потік подій	Прецедент розпочинається, коли актор натискає на кнопку «Профіль» у застосунку.
Базовий потік	1. Користувач натискає кнопку «Увійти». 2. Застосунок показує вікно автентифікації 3. Користувач натискає кнопку «Зареєструватися» 4. Користувач вводить необхідні для реєстрації данні валідні дані. 5. Користувач натискає кнопку «Зареєструватися». 6. Застосунок перевіряє дані на збіг у системі, та при відсутності ідентичних даних реєструє нового користувача.
Постумови	Якщо реєстрацію успішно пройдено, користувача буде перенаправлено у головне меню.

Таблиця 2.8 – Специфікація варіанту використання «Додати товар у кошик»

Складова	Опис
1	2
Опис прецеденту	Додати товар до кошика
Дійові особи прецеденту	Користувач

Продовження таблиці 2.8

1	2
Передумови	Користувач відкрив мобільний застосунок
Потік подій	Прецедент починається, коли Користувач переходить на сторінку списку товару
Базовий потік	7. Користувач обирає один або кілька продуктів зі списку. 8. Користувач натискає кнопку «Додати до кошика». 9. Застосунок додає товар у кошик
Альтернативні потоки	Відсутні
Постумови	Користувач має можливість перейти до кошика та переглянути доданий туди товар та оформити замовлення.

Після розробки специфікації варіантів використання маємо можливість продовжити розробку проекту.

2.2 Розробка архітектури мобільного застосунку "eTrick" для продажу мобільних аксесуарів

Для розробки застосунку "eTrick" була обрана клієнт-серверна архітектура з використанням трьохрівневої моделі. Ця архітектура відповідає потребам застосунку і забезпечує ефективну та масштабовану роботу системи. Клієнт-серверна архітектура з трьохрівневою моделлю є оптимальним вибором для застосунку "eTrick" з огляду на ефективну організацію, масштабованість, мобільність та розширюваність системи. Вона дозволяє ефективно виконувати різні функції та задовольняти потреби користувачів у зручному та надійному способі придбання мобільних аксесуарів.

2.2.1 Розробка діаграми компонентів мобільного застосунку "eTrick" для продажу мобільних аксесуарів

У ході розробки мобільного застосунку було обрано Клієнт-серверну архітектуру саме через її переваги:

1. Розділення відповідальностей: Клієнт-серверна архітектура дозволяє чітко розділити функціональність системи між клієнтською та серверною частинами. Клієнтська частина відповідає за інтерфейс користувача та представлення даних, тоді як серверна частина забезпечує обробку бізнес-логіки, доступ до бази даних та інші операції. Це розділення сприяє чистоті коду, полегшує розробку та підтримку системи.

2. Масштабованість: Трьохрівнева модель дозволяє гнучко масштабувати систему залежно від потреб. Клієнтська, серверна та база даних можуть бути розгорнуті на окремих серверах або навіть використовувати хмарні сервіси. Це дозволяє ефективно керувати навантаженням та забезпечити швидку та надійну роботу системи, навіть при збільшенні обсягу даних та кількості користувачів.

3. Мобільність: Клієнт-серверна архітектура підходить для мобільних застосунків, таких як "eTrick". Клієнтська частина може бути реалізована у вигляді мобільного додатку, що працює на смартфонах або планшетах, а серверна частина може бути розміщена на веб-сервері. Це дозволяє користувачам зручно використовувати додаток на своїх мобільних пристроях, незалежно від їх місцезнаходження.

4. Розширюваність: Клієнт-серверна архітектура з трьохрівневою моделлю дозволяє легко розширювати функціонал системи. Кожен рівень може бути модульним і замінюватись або розширюватись без впливу на інші компоненти. Це робить систему гнучкою та піддається змінам залежно від потреб бізнесу та користувачів.



Рисунок 2.2 – Діаграма компонентів мобільного застосунку "eTrick" для продажу аксесуарів для мобільних телефонів

У мобільному застосунку "eTrick" графічний інтерфейс користувача (View) реалізований з використанням нативних технологій, які обробляються на самому пристрої користувача. Контролер (Controller) відповідає за обробку запитів, які надходять від користувача, та передає їх відповідним сервісам (Service). Сервіси (Service) містять бізнес-логіку додатку і використовують моделі (Model) для отримання та обробки даних. Моделі (Model) представляють сутності, які присутні в застосунку, і взаємодіють з репозиторіями (Repository), щоб отримувати або зберігати дані в базі даних. Репозиторії (Repository) надають інтерфейс для взаємодії з базою даних, забезпечуючи доступ до неї. Графічно серію взаємодії зображено на рисунку 2.3

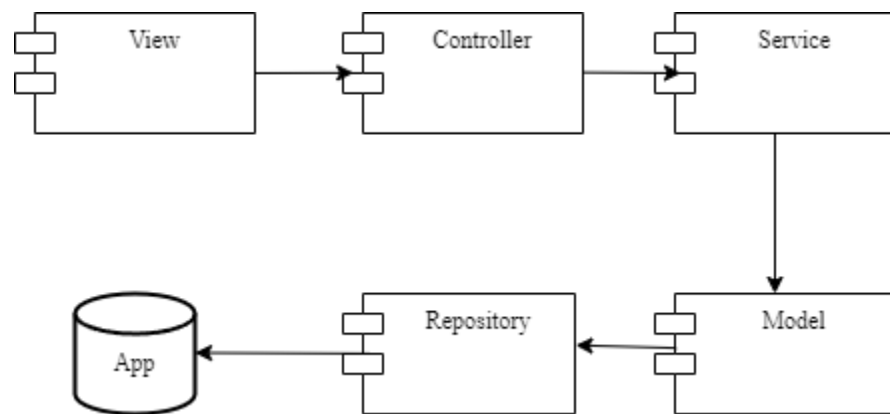


Рисунок 2.3 – Діаграма взаємодії компонентів мобільного застосунку "eTrick" для продажу аксесуарів для мобільних телефонів

Обрана архітектура є оптимальним вибором для мобільного застосунку "eTrick", оскільки вона забезпечує ефективну організацію та взаємодію його компонентів. Вона дозволяє забезпечити швидку та надійну роботу застосунку, зменшити залежність від мережевого підключення та забезпечити зручний та доступний інтерфейс для користувачів.

2.2.2 Розробка діаграми розміщення мобільного застосунку "eTrick" для продажу мобільних аксесуарів

Для представлення принципу розгорнутості програми фізично буде доцільним використання діаграми розміщення (рисунок 2.4). Діаграма розміщення є однією з діаграм уніфікованої мови моделювання (UML), яка використовується для відображення фізичної архітектури програмного забезпечення. Вона дозволяє показати, які програмні та апаратні засоби використовуються в системі, а також як вони взаємодіють між собою [6].

Діаграма розміщення є важливим інструментом, який дозволяє візуалізувати фізичні компоненти системи, такі як сервери, комп'ютери, додатки, бази даних і мережі, а також їх зв'язки. Вона допомагає зрозуміти, як ці компоненти розташовані фізично та як вони взаємодіють між собою. Це дає змогу отримати уявлення про архітектуру системи і показати зв'язки та залежності між різними її елементами.

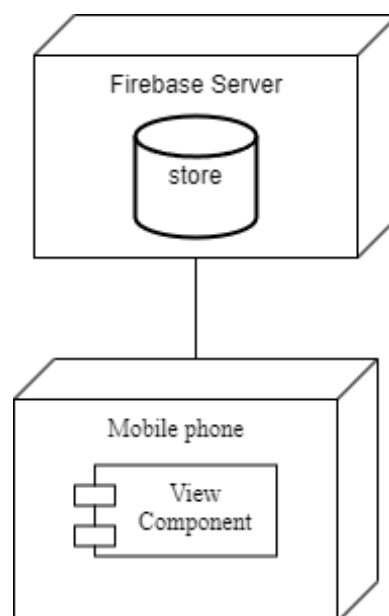


Рисунок 2.4 – Діаграма розміщення мобільного застосунку "eTrick" для продажу аксесуарів для мобільних телефонів

2.3 Проект мобільного застосунку "eTrick" для продажу мобільних аксесуарів

Під час розробки мобільного застосунку "eTrick" для продажу мобільних аксесуарів, було створено кілька важливих артефактів, які грають ключову роль у формалізації поведінки та дизайну системи.

Один з них - діаграми діяльності, які використовуються для зображення послідовності дій та поведінки в мобільному застосунку. Ці діаграми відображають потік дій, враховуючи умови для виконання кожної дії, а також вхідні та вихідні переходи між кроками. Вони надають зрозумілу ілюстрацію того, як система працює і які дії необхідні для досягнення певних результатів. Діаграми діяльності сприяють аналізу та визначенню правильної послідовності кроків, забезпечуючи ефективну роботу застосунку.

Другим артефактом є інформаційна модель даних, розроблена з використанням методології IDEF1X. Ця модель відображає структуру даних в мобільному застосунку та зв'язки між ними. Вона допомагає зрозуміти сутність даних, їх атрибути та взаємозв'язки, що дозволяє забезпечити цілісність та ефективне управління даними в системі. Інформаційна модель даних є важливим інструментом для проектування бази даних та забезпечення правильного збереження та обробки інформації.

Крім того, на етапі концептуального проектування було розроблено дизайн інтерфейсу користувача для мобільного застосунку "eTrick". Цей дизайн фокусується на створенні інтуїтивно зрозумілого та зручного інтерфейсу, що полегшує взаємодію користувача з програмним забезпеченням. Він враховує аспекти макетування, навігації та використання візуальних елементів для поліпшення користувацького досвіду та зручності використання застосунку. Дизайн інтерфейсу користувача сприяє залученню користувачів, забезпечуючи їм зручну та привабливу платформу для придбання мобільних аксесуарів.

Загалом, ці артефакти відіграють важливу роль у розробці мобільного застосунку "eTrick". Вони формалізують поведінку системи, визначають

структуру даних та забезпечують зручний інтерфейс для користувачів. Ці аспекти є ключовими для успішної реалізації та ефективного функціонування мобільного застосунку для продажу мобільних аксесуарів "eTrick".

2.3.1 Ескізний проект для мобільного застосунку "eTrick" для продажу мобільних аксесуарів

В ході розробки ескізного проекту мобільного застосунку "eTrick" для продажу мобільних аксесуарів було побудовано діаграми діяльності, що відображають сценарії основних варіантів використання. Діаграми діяльності є частиною уніфікованої мови моделювання (UML) і використовуються для опису поведінки системи або процесу. Вони дозволяють показати послідовність дій, які можуть бути виконані в мобільному застосунку "eTrick", а також умови, що потрібно виконати для виконання цих дій.

В уніфікованій мові моделювання (UML) діаграма діяльності – це графічне зображення виконаного набору дій процедурної системи та розглядається варіація діаграми стану. Діаграми діяльності описують паралельні та умовні дії, використовують випадки та системні функції на детальному рівні [7].

Кожна дія на діаграмі діяльності має вхідні та вихідні потоки, які показують переходи між кроками виконання дії. Умови можуть бути використані для показу умов, за яких дія може бути виконана або пропущена. Діаграма діяльності для варіанту використання "Автентифікація" в мобільному застосунку "eTrick" представлена на рисунку 2.5.

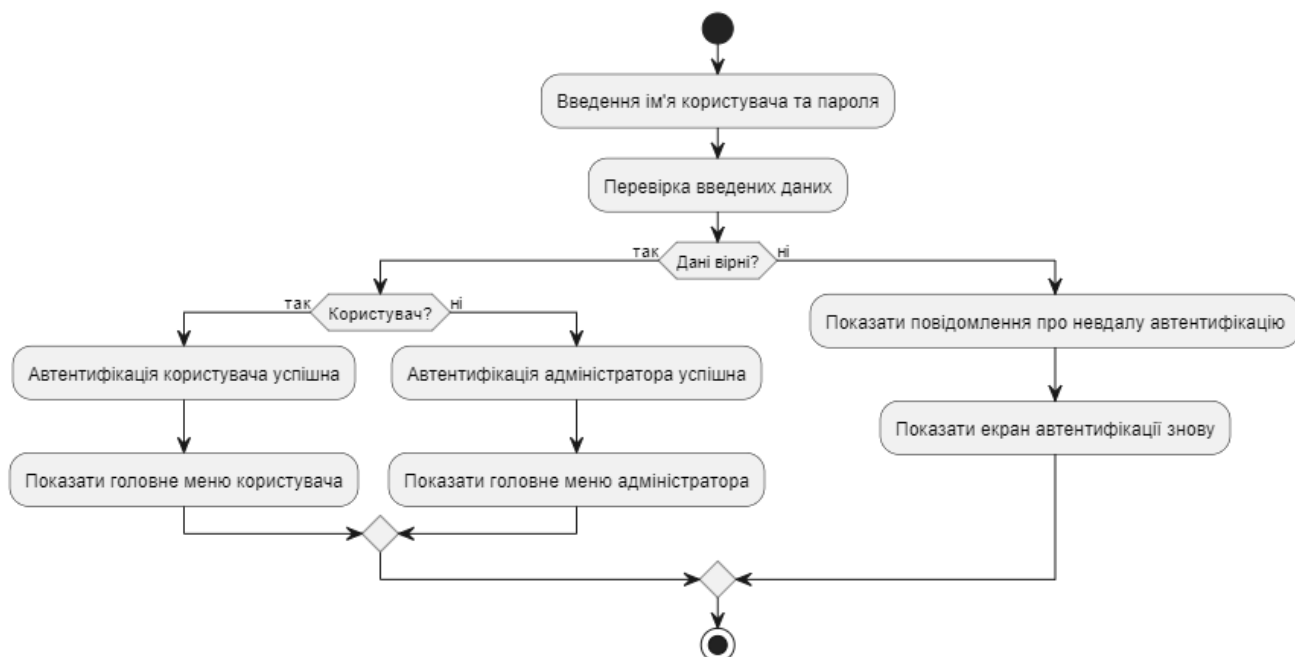


Рисунок 2.5 – Діаграма діяльності «Автентифікація»

Діаграма діяльності для варіанту використання «Пошук товару» представлена на рисунку 2.6.

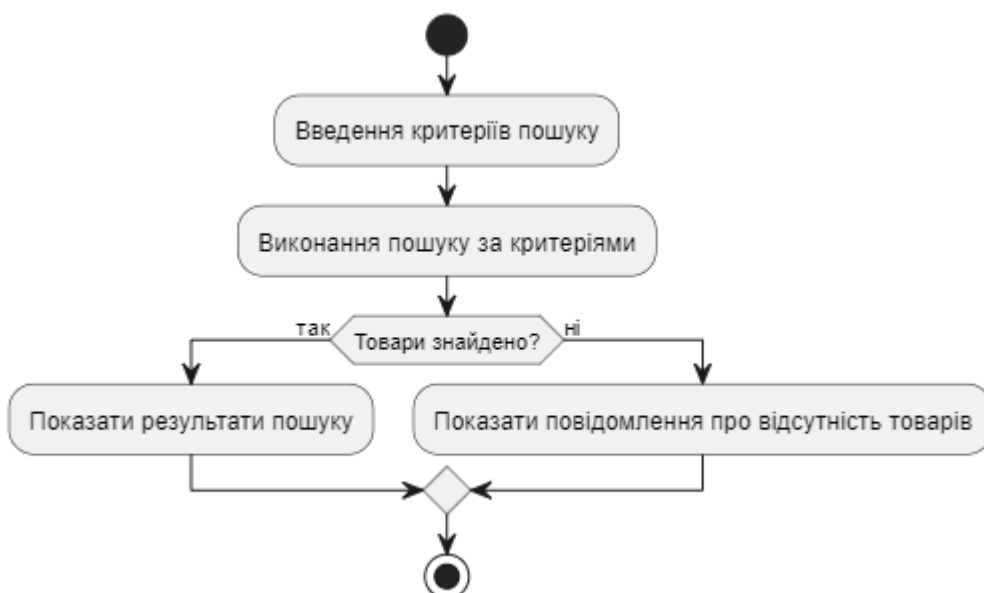


Рисунок 2.6 – Діаграма діяльності «Пошук товару»

Діаграма діяльності для варіанту використання «Редагування інформації про товар» представлена на рисунку 2.7.

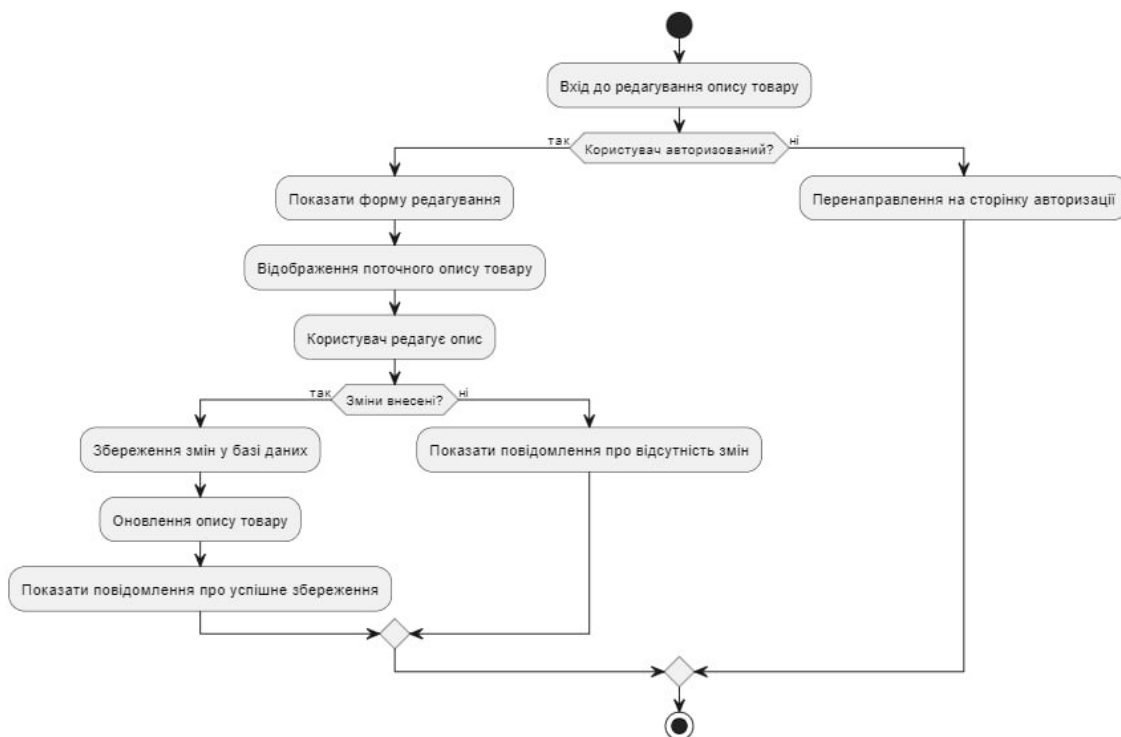


Рисунок 2.7 – Діаграма діяльності «Редагування інформації про товар»

Діаграма діяльності для варіанту використання «Отримання сповіщень» представлена на рисунку 2.8.

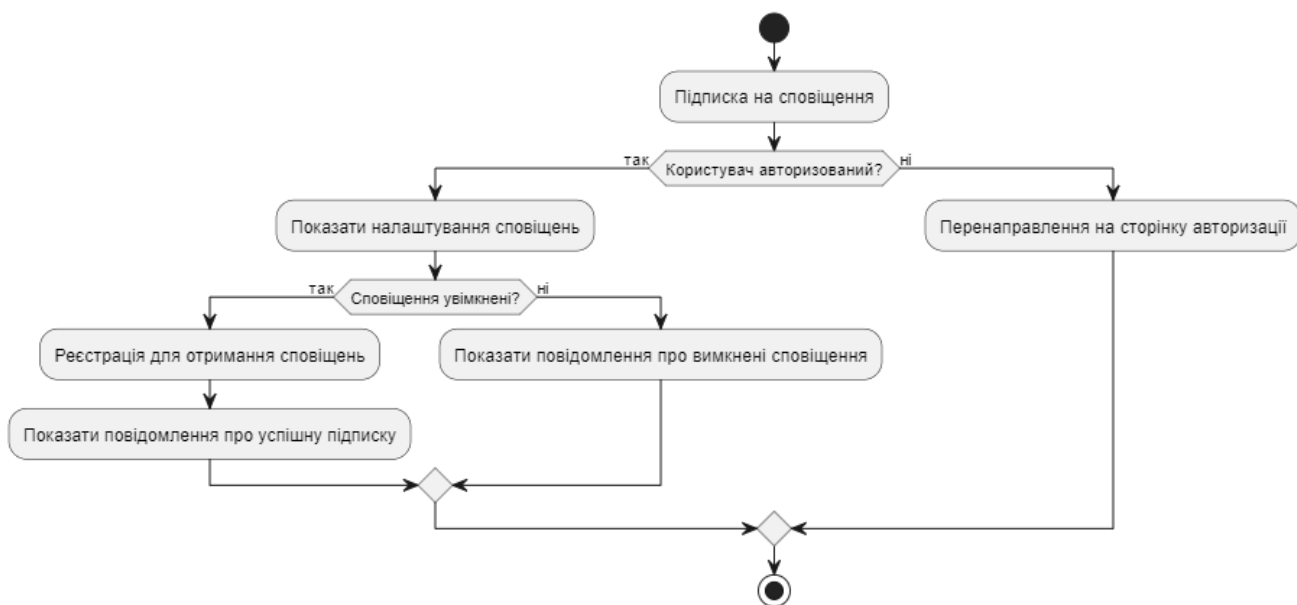


Рисунок 2.8 – Діаграма діяльності «Отримання сповіщень»

Діаграма діяльності для варіанту використання «Реєстрація» представлена на рисунку 2.9.

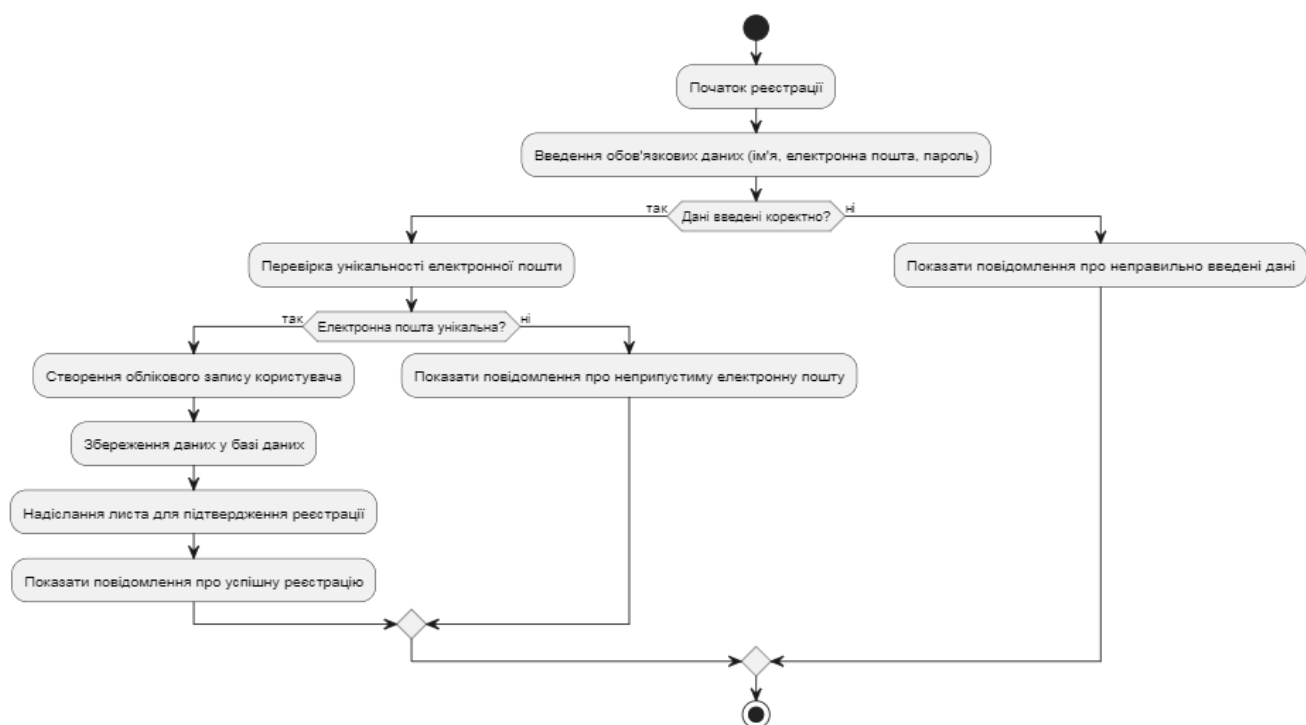


Рисунок 2.9 – Діаграма діяльності «Реєстрація»

Діаграма діяльності для варіанту використання «Додавання товару у кошик» представлена на рисунку 2.10.

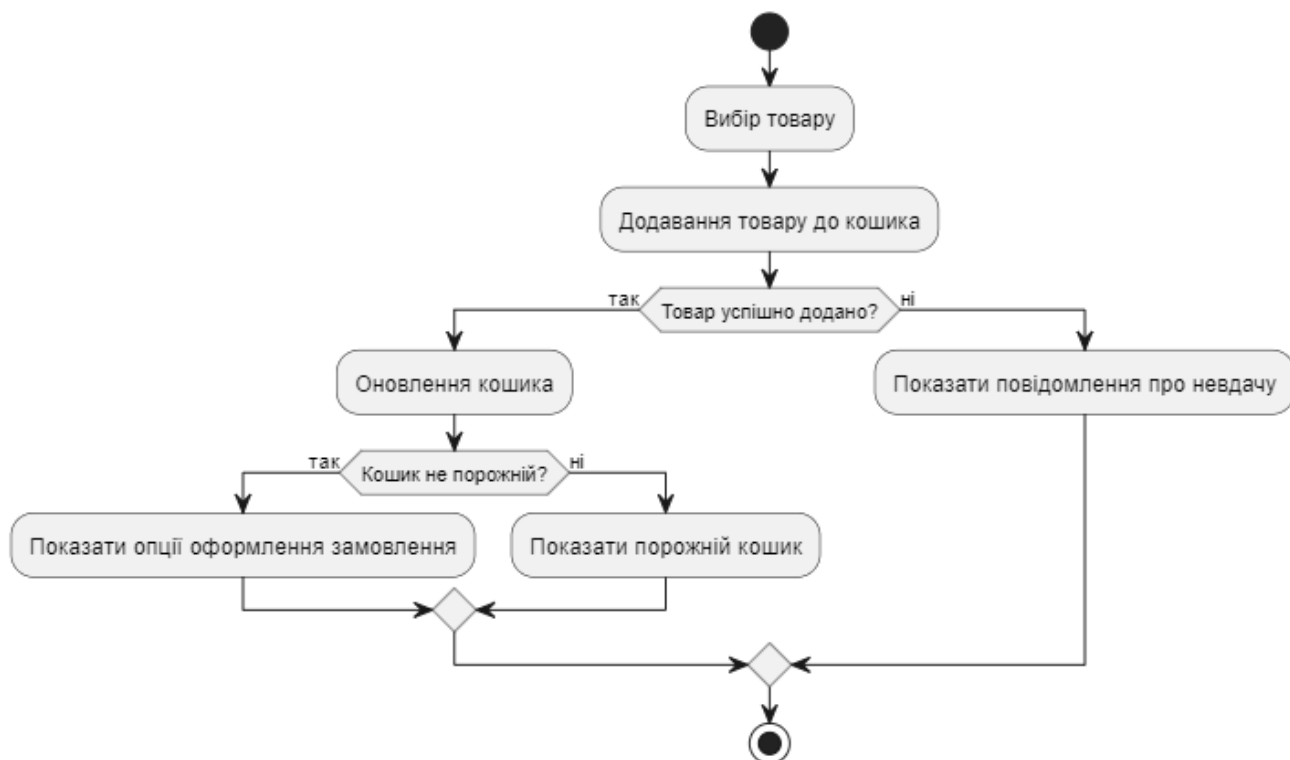


Рисунок 2.10 – Діаграма діяльності «Додавання товару у кошик»

На рисунку 2.11 зображено концептуальну модель даних для мобільного застосунку "eTrick". Ця модель є абстрактним представленням інформації, яка буде зберігатись у базі даних, і описує основні сутності, їх атрибути та зв'язки між ними.

У цій концептуальній моделі даних представлені основні сутності та їх атрибути для мобільного застосунку. Застосунок має розділення на рівні користувачів та адміністраторів. Товари поділені на категорії та бренди, і можуть мати огляди, написані користувачами. Користувачі можуть мати кошик з елементами кошика, які містять конкретні товари. Користувачі також можуть робити замовлення, які містять деталі замовлення, які знову ж таки містять конкретні товари.

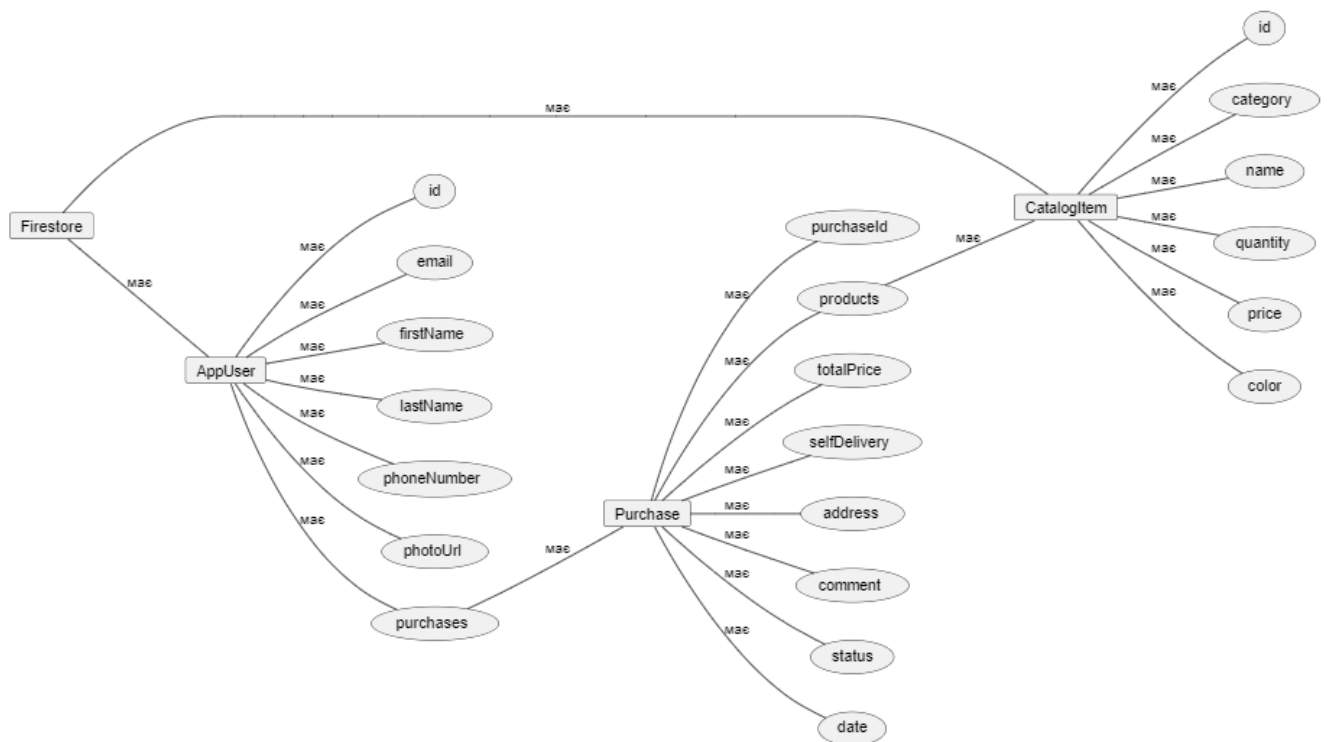


Рисунок 2.11 – Концептуальна модель мобільного застосунку "eTrick" для продажу мобільних аксесуарів

Концептуальна модель даних для мобільного додатку "eTrick" для продажу мобільних аксесуарів включає три основні сутності: Purchase (Покупка), CatalogItem (Елемент каталогу) та AppUser (Користувач додатку).

База даних Firestore містить у собі список AppUser, CatalogItem та Purchase.

Purchase (Покупка) містить поля, які описують окремі атрибути покупки, такі як ідентифікатор покупки (purchaseId), список товарів (products), загальна вартість (totalPrice), спосіб доставки (selfDelivery), адреса доставки (address), коментар (comment), статус (status) та дата (date). Кожне поле Purchase зв'язане з CatalogItem на діаграмі.

CatalogItem (Елемент каталогу) представляє окремий товар з каталогу і включає поля, такі як ідентифікатор товару (id), категорія (category), назва (name), кількість (quantity), ціна (price) та колір (color). Кожне поле CatalogItem також зв'язане з відповідним на діаграмі.

AppUser (Користувач додатку) має свої власні атрибути, включаючи ідентифікатор користувача (id), електронну пошту (email), ім'я (firstName), прізвище (lastName), номер телефону (phoneNumber), URL фото (photoUrl) та список покупок (purchases). Відповідні зв'язки відображають зв'язки між полями AppUser та Purchase на діаграмі.

Крім того, зв'язок між Purchase та CatalogItem показує, що кожна покупка (Purchase) містить список товарів (products), які є елементами каталогу (CatalogItem).

Ця концептуальна модель даних допомагає уявити структуру та зв'язки між сутностями мобільного застосунку "eTrick" для продажу мобільних аксесуарів. Вона відображає важливі поля та зв'язки, які необхідні для правильного функціонування додатку.

2.3.2 Технічний проект для мобільного застосунку "eTrick" для продажу мобільних аксесуарів

У діаграмі класів представлені основні класи, що використовуються в мобільному застосунку "eTrick". Користувачі можуть реєструватися,

авторизуватися, шукати товари, додавати їх у кошик, робити замовлення та залишати огляди. Адміністратори мають можливість авторизуватися, додавати, редагувати та видаляти товари, переглядати замовлення та видаляти огляди.

Діаграма класів – статичне представлення структури моделі в UML. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів може містити позначення для пакетів та може містити позначення для вкладених пакетів [8].

Діаграма класів (див. рис. 2.12) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини (зв'язки).

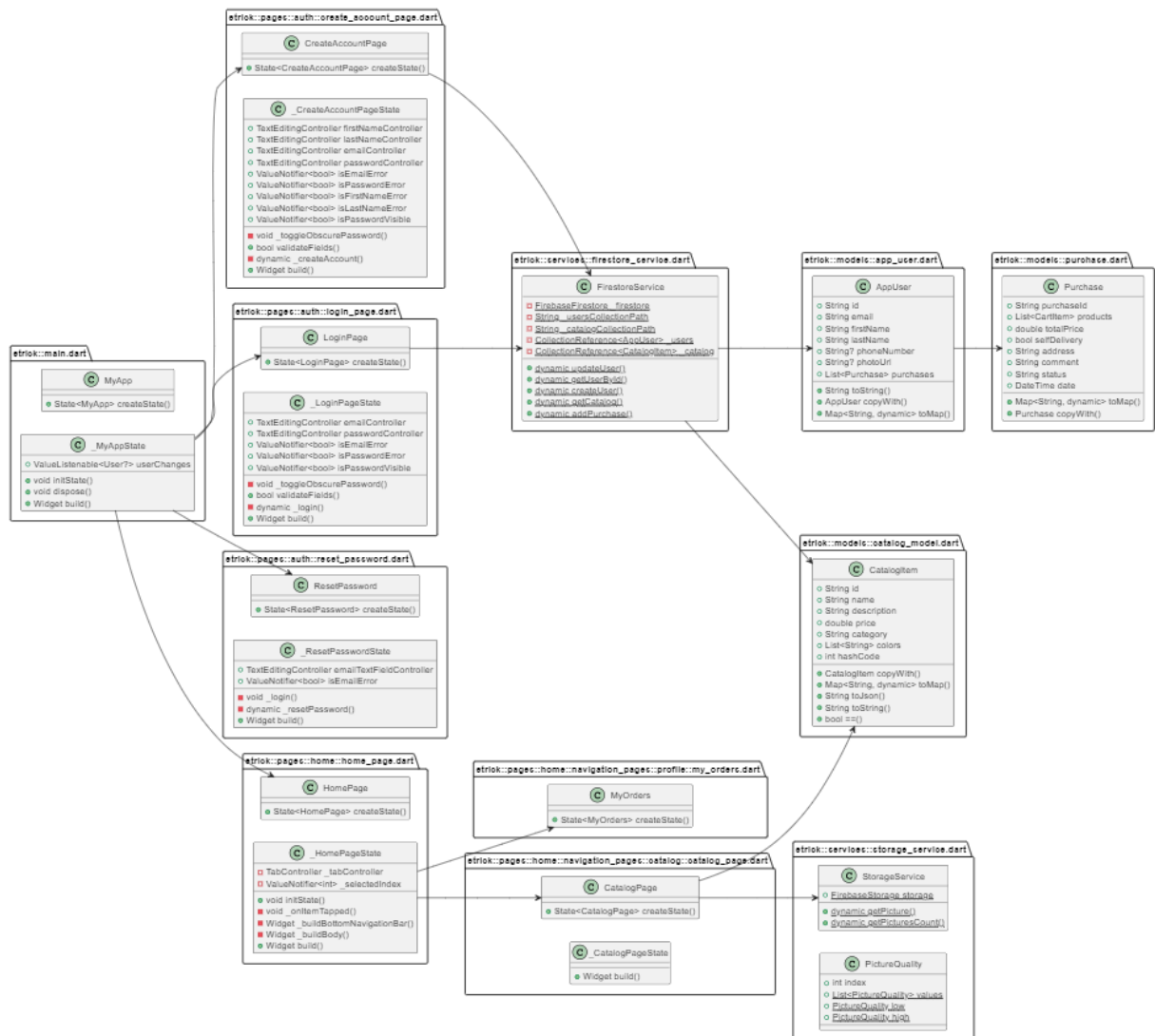


Рисунок 2.12 – Діаграма класів мобільного застосунку "eTrick" для продажу мобільних аксесуарів

Нижче представлена розшифровка основних класів та їх взаємозв'язків:

- MyApp: Цей клас представляє головний віджет додатка. В ньому є метод createState(), який створює об'єкт стану для віджета.
- MyAppState: Цей клас представляє об'єкт стану для віджета MyApp. Він має властивості, такі як userChanges, яке є ValueListenable і повідомляє про зміни у даних користувача. Він також має методи для ініціалізації та видалення стану, а також для побудови віджета.
- CreateAccountPage: Цей клас представляє сторінку для створення нового облікового запису користувача. Він має об'єкт стану _CreateAccountPageState, який відповідає за логіку та побудову користувацького інтерфейсу сторінки.
- CreateAccountPageState: Цей клас представляє об'єкт стану для сторінки CreateAccountPage. Він має властивості, такі як firstNameController, emailController і т.д., які є контролерами для обробки введення користувача. Він також має методи для перевірки полів форми та створення нового облікового запису.
- LoginPage: Цей клас представляє сторінку входу в систему. Він має об'єкт стану _LoginPageState, який відповідає за логіку та побудову користувацького інтерфейсу сторінки.
- LoginPageState: Цей клас представляє об'єкт стану для сторінки LoginPage. Він має властивості, такі як emailController, passwordController і т.д., які є контролерами для обробки введення користувача. Він також має методи для перевірки полів форми та виконання входу в систему.
- ResetPassword: Цей клас представляє сторінку скидання пароля. Він має об'єкт стану _ResetPasswordState, який відповідає за логіку та побудову користувацького інтерфейсу сторінки.
- ResetPasswordState: Цей клас представляє об'єкт стану для сторінки ResetPassword. Він має властивості, такі як emailTextFieldController, який є контролером для обробки введення електронної пошти користувача. Він також має методи для виконання скидання пароля.

- `HomePage`: Цей клас представляє головну сторінку додатка. Він має об'єкт стану `_HomePageState`, який відповідає за логіку та побудову користувацького інтерфейсу сторінки.
- `HomePageState`: Цей клас представляє об'єкт стану для сторінки `HomePage`. Він має приватні властивості, такі як `_tabController` і `_selectedIndex`, які використовуються для керування вкладками та вибором пунктів навігації. Він також має методи для ініціалізації, побудови нижньої панелі навігації та головного вмісту сторінки.
- `CatalogPage`: Цей клас представляє сторінку каталогу товарів. Він має об'єкт стану `_CatalogPageState`, який відповідає за логіку та побудову користувацького інтерфейсу сторінки.
- `AppUser`: Цей клас представляє користувача додатка з властивостями, такими як `id`, `email`, `firstName`, `lastName`, `phoneNumber`, `photoUrl` та `purchases`. Він також має методи для створення копії об'єкта, конвертації в мапу та представлення об'єкта у вигляді рядка.
- `CatalogItem`: Цей клас представляє товар в каталозі з властивостями, такими як `id`, `name`, `description`, `price`, `category` та `colors`. Він також має методи для створення копії об'єкта, конвертації в мапу, серіалізації у формат JSON та представлення об'єкта у вигляді рядка.
- `Purchase`: Цей клас представляє покупку з властивостями, такими як `purchaseId`, `products`, `totalPrice`, `selfDelivery`, `address`, `comment`, `status` та `date`. Він також має методи для створення копії об'єкта та конвертації в мапу.
- `FirestoreService`: Цей клас представляє сервіс для взаємодії з базою даних `Firestore`. Він має методи для оновлення користувача, отримання користувача за ідентифікатором, створення користувача, отримання каталогу товарів та додавання покупки.
- `StorageService`: Цей клас представляє сервіс для роботи з обліковими записами `Firebase Storage`. Він має методи для отримання зображень та отримання кількості зображень.

– PictureQuality: Цей клас представляє якість зображення. Він має властивість index та статичні властивості low та high, що представляють різні рівні якості зображення.

Наступним етапом розробки мобільного застосунку є розробка діаграми послідовності для мобільного застосунку "eTrick" для продажу мобільних аксесуарів. Беручи до уваги, таблиці і діаграми представлені, вище була розроблена динамічна модель використання, а саме певна кількість діаграм послідовності, які представлені на рисунках 2.13 – 2.18 .

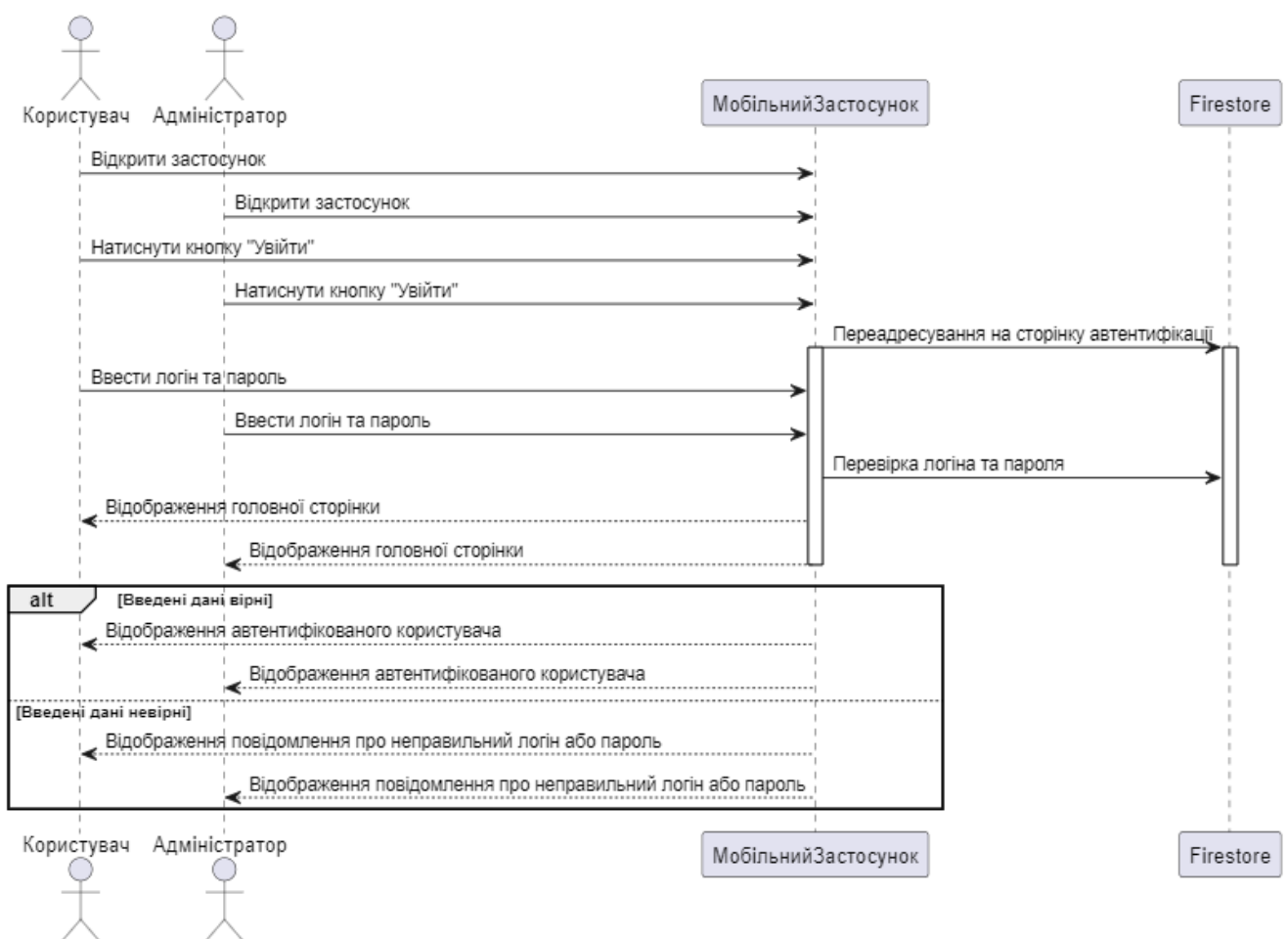


Рисунок 2.13 – Діаграма послідовності для прецеденту «Автентифікація»

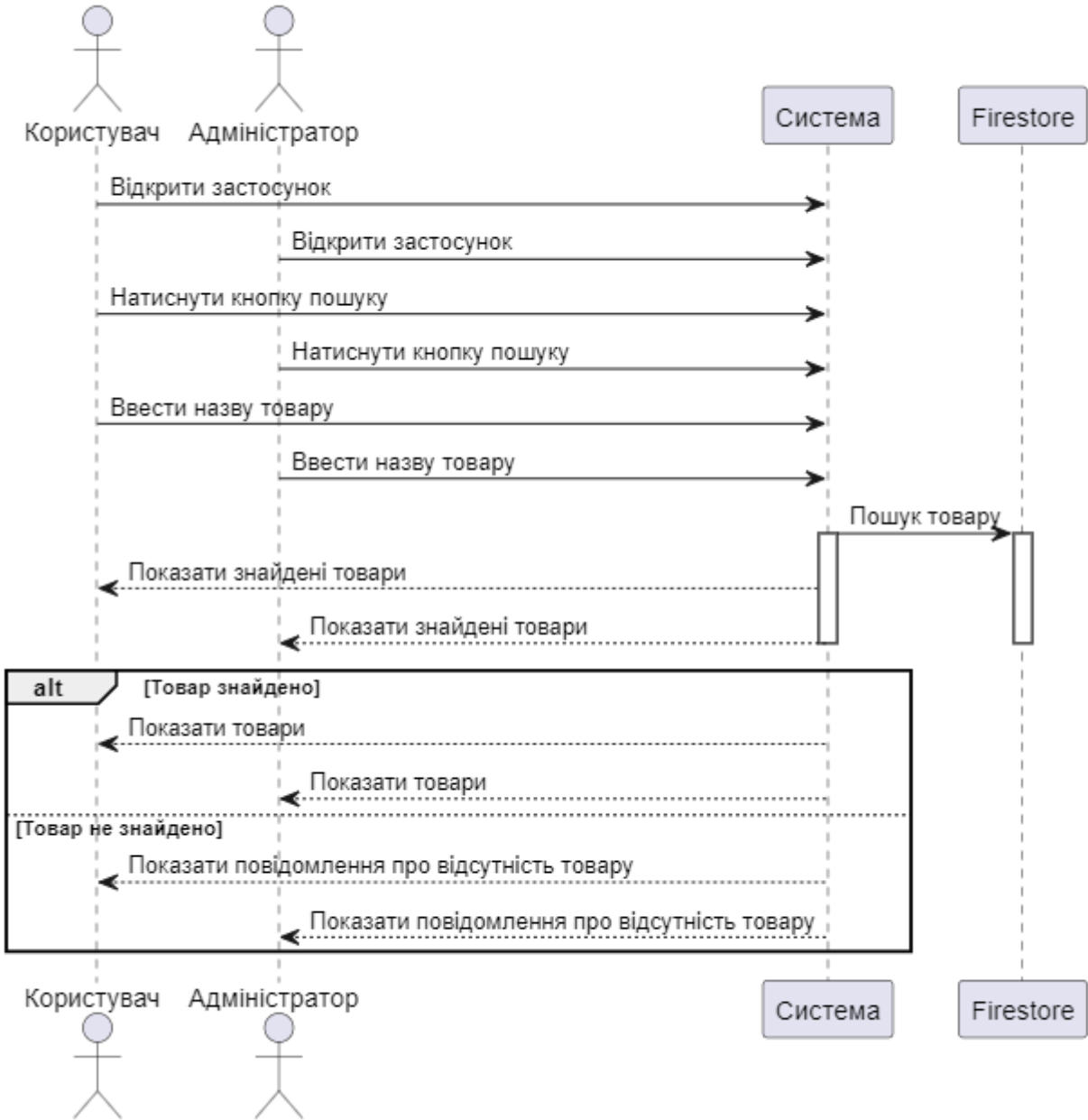


Рисунок 2.14 – Діаграма послідовності для прецеденту «Пошук товару»

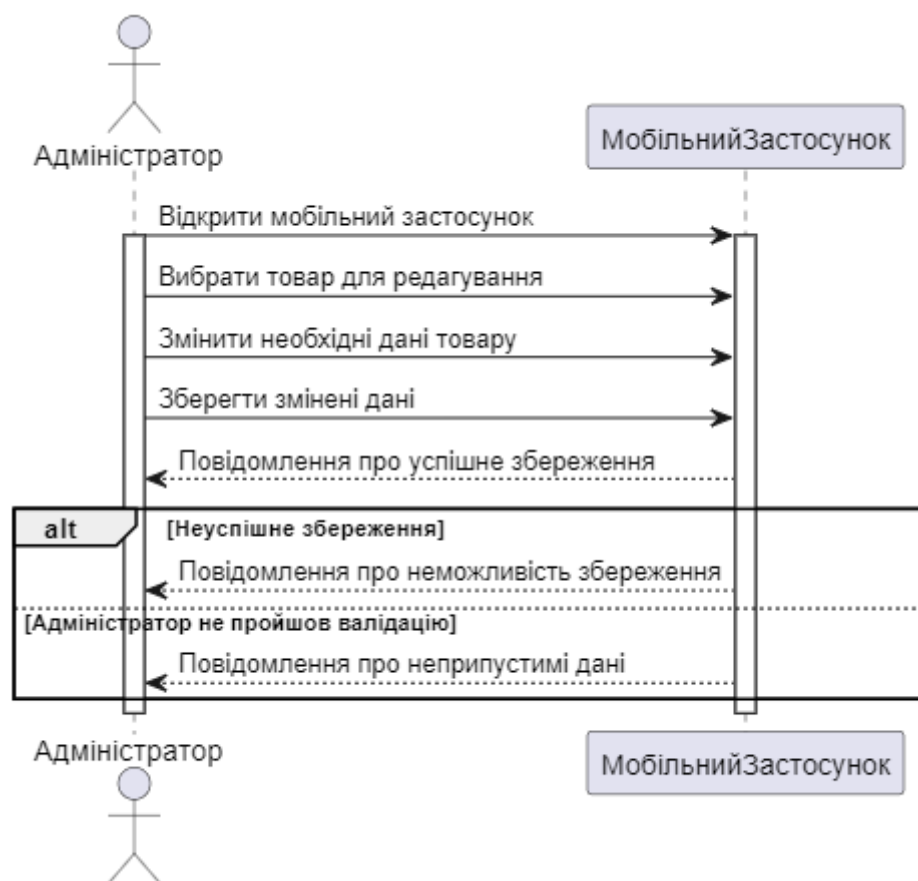


Рисунок 2.15 – Діаграма послідовності для прецеденту «Редагування інформації про товар»

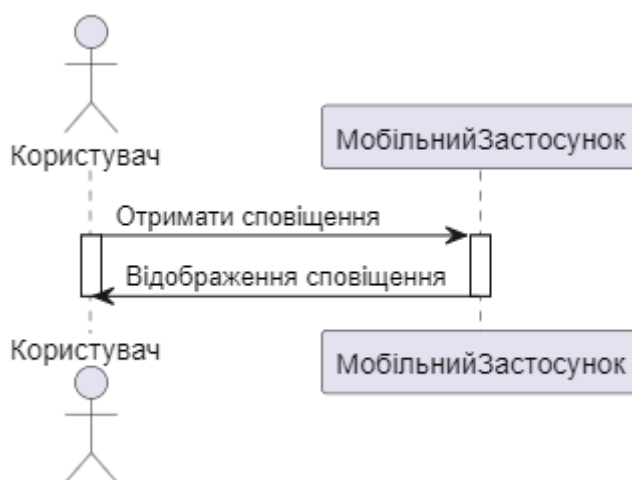


Рисунок 2.16 – Діаграма послідовності для прецеденту «Отримання сповіщень»

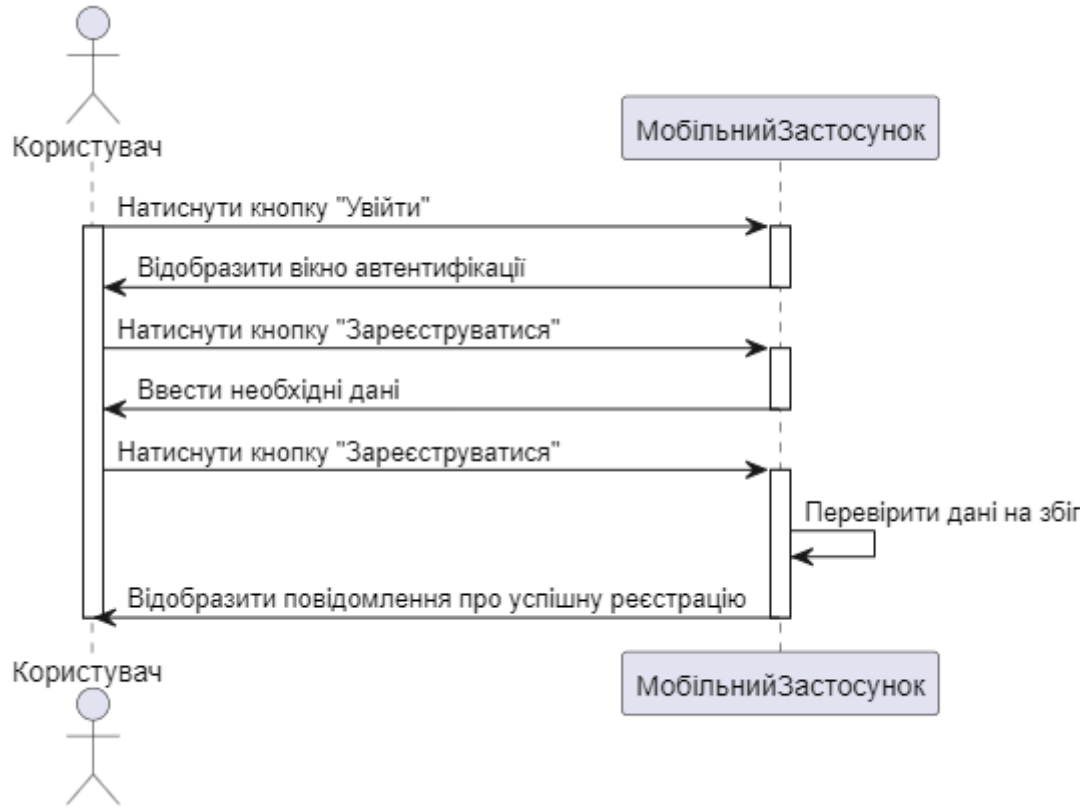


Рисунок 2.17 – Діаграма послідовності для прецеденту «Реєстрація»

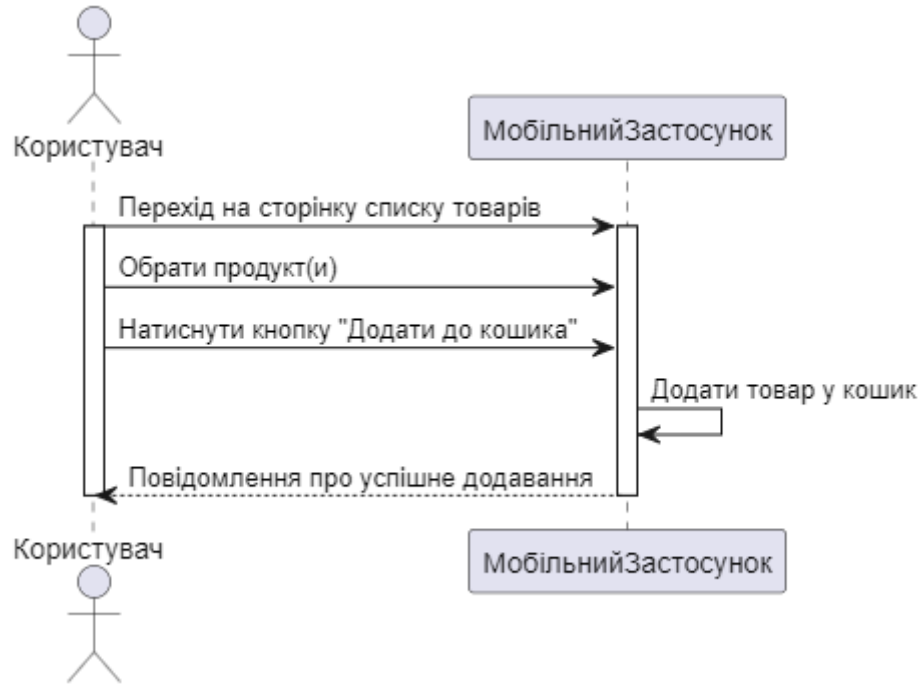


Рисунок 2.18 – Діаграма послідовності для прецеденту «Реєстрація»

Логічна модель даних використовується для опису структури та взаємозв'язків між даними в контексті певної системи або домену. Вона допомагає зрозуміти сутність даних, їх атрибути і взаємозв'язки між ними без прив'язки до конкретної технології збереження даних. Логічна модель даних для мобільного застосунку "eTrick" для продажу мобільних аксесуарів представлена нижче на рисунку 2.19.

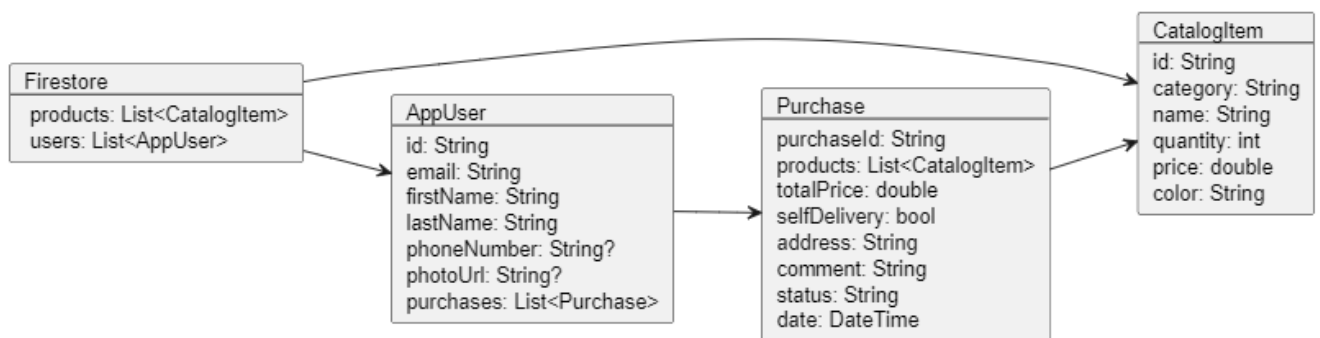


Рисунок 2.19 – Логічна модель даних мобільного застосунку "eTrick" для продажу мобільних аксесуарів

Логічній моделі даних є важливим інструментом для проектування бази даних, спрощує розуміння структури і взаємозв'язків даних, допомагає виявити проблеми та покращити ефективність системи.

2.3.3 Робочий проект мобільного застосунку "eTrick" для продажу мобільних аксесуарів

При розробці мобільного застосунку "eTrick" для продажу мобільних аксесуарів будуть використані наступні технології та інструменти:

- Visual Studio Code – легкий та потужний редактор коду, розроблений компанією Microsoft, буде використовуватися як середовище розробки для програмного забезпечення "eTrick". Він підтримує різні мови програмування та забезпечує широкі можливості для редагування, налагодження та розробки програмного коду [9].

- Flutter – інноваційний фреймворк для розробки мобільних додатків, який дозволяє створювати високоякісні кросплатформові додатки для iOS та Android. Використання Flutter дозволить швидко розробити інтуїтивно зрозумілий та естетично привабливий інтерфейс користувача для мобільного застосунку "eTrick" [1].

- Dart – мова програмування, яка використовується разом з Flutter для розробки функціональності "eTrick". Завдяки синтаксису, схожому на мови Java та JavaScript, Dart є легким для вивчення та використання. Використання Dart дозволить писати ефективний та надійний код для забезпечення плавної роботи застосунку [10].

- Firebase – комплексна платформа розробки застосунків, яка надає набір хмарних сервісів для зручного управління та розгортання мобільного застосунку "eTrick". Використання Firebase дозволить забезпечити автентифікацію користувачів, зберігання даних, реальний час спілкування з сервером та багато іншого [2].

- Firebase Authentication – сервіс, що надає можливість автентифікації та авторизації користувачів мобільного застосунку "eTrick". За допомогою Firebase Authentication можна забезпечити безпеку і захист особистих даних користувачів, а також використовувати різні методи автентифікації, такі як електронна пошта, соціальні мережі тощо.

- Firebase Cloud Firestore – гнучка й масштабована NoSQL база даних, що пропонується Firebase. Firebase Cloud Firestore дозволяє зберігати, синхронізувати та отримувати дані між мобільним застосунком та сервером в режимі реального часу.

- Firebase Cloud Functions – сервіс, який дозволяє створювати та розгортати функції в хмарному середовищі Firebase. Використання Firebase Cloud Functions дозволить виконувати певні операції на серверному боці, такі як обробка замовлень чи надсилення сповіщень.

- Firebase Hosting – сервіс для розгортання та хостингу мобільного застосунку "eTrick" в хмарному середовищі Firebase. Firebase Hosting дозволяє

зручно розгорнути застосунок, забезпечити його доступність для користувачів та швидку загрузку.

Використовуючи ці технології та інструменти, розробка мобільного застосунку "eTrick" для продажу мобільних аксесуарів стане ефективною та зручною, а користувачі отримають функціональний та зручний інтерфейс додатку.

Мобільний застосунок "eTrick" використовує дві різні колекції сховищ для відмінних компонентів системи. Firestore використовується для зберігання важливих даних, таких як інформація про продукти, користувачів та замовлення, що дозволяє ефективно управляти та отримувати доступ до цих даних. Завдяки швидкому доступу та можливості масштабування, Firestore забезпечує швидку та надійну роботу для компонентів, які використовують цю базу даних.

З іншого боку, використано Storage для зберігання та керування зображеннями, пов'язаними з продуктами в мобільному застосунку "eTrick". Storage забезпечує можливість зручно завантажувати та отримувати зображення продуктів, забезпечуючи швидкий доступ до них. Це дозволяє показувати високоякісні зображення користувачам без зайвих затримок.

Використання двох різних колекцій для зберігання даних – Firestore і Storage – допомагає оптимізувати роботу мобільного застосунку "eTrick". Firestore використовується для управління основними даними про продукти, користувачів та замовлення, забезпечуючи швидкий доступ та масштабованість. З іншого боку, Storage спеціалізується на зберіганні та отриманні зображень продуктів, забезпечуючи швидкий та ефективний доступ до них.

Це розділення даних дозволяє ефективно використовувати ресурси та знижує витрати. Кожне зберігання даних спрямоване на свої завдання і надає оптимальну продуктивність та ефективність для відповідних компонентів мобільного застосунку "eTrick".

Для мобільного застосунку "eTrick" база даних у Firebase має дві основні групи Firestore і Storage (див. рис. 2.20).

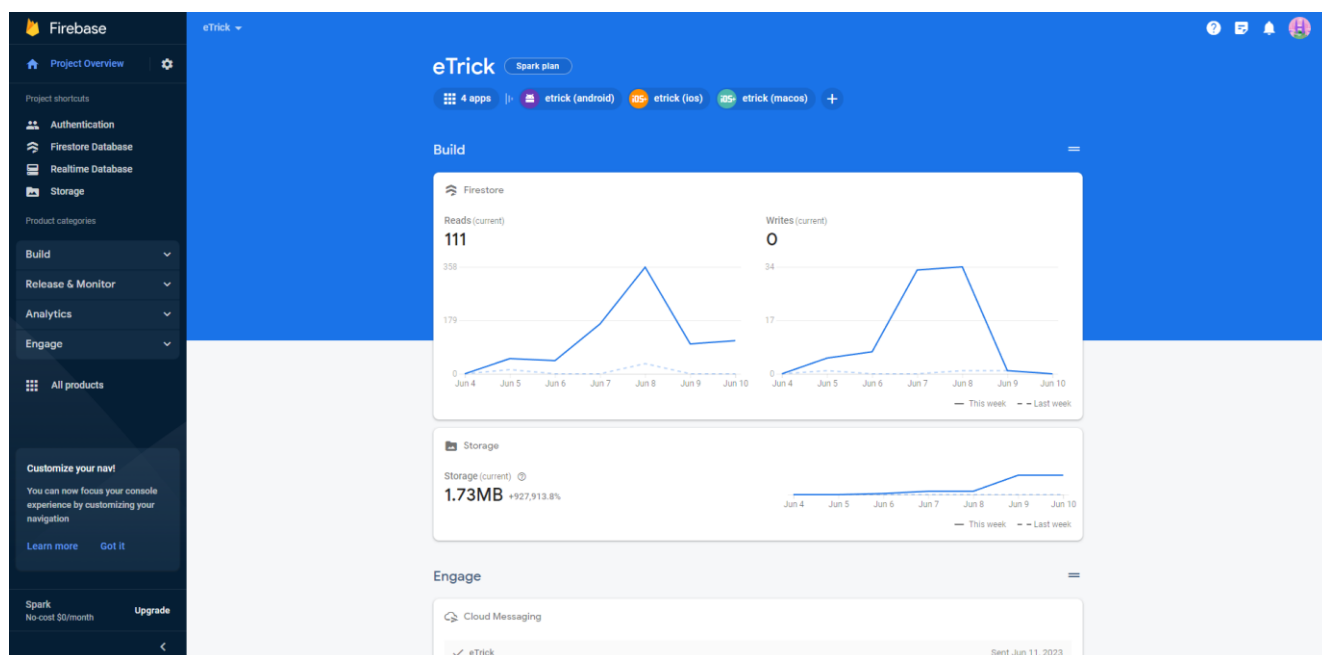


Рисунок 2.20 – Бази даних Firestore і Storage у Firebase

У базі даних Firestore містяться товари та зареєстровані користувачі. Структура catalog зображена нижче на малюнку 2.21.

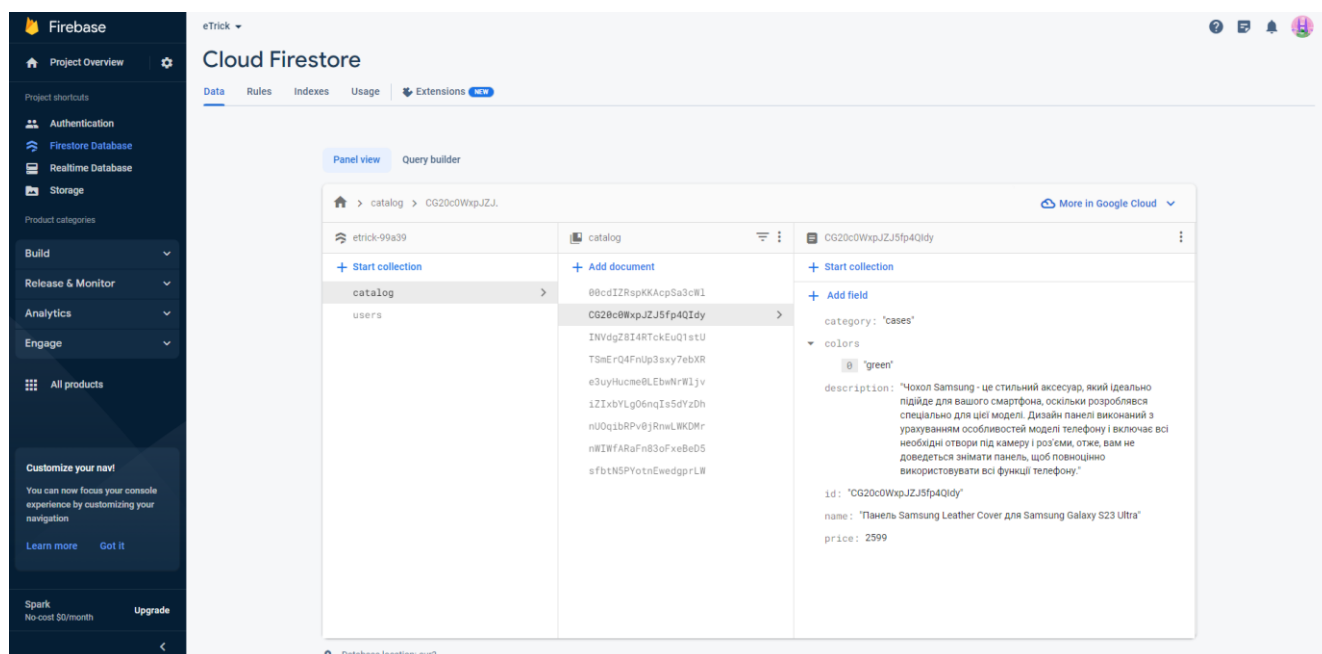


Рисунок 2.21 – Структура товарів у базі даних Firestore

Клас `CatalogItem` представляє елемент каталогу товарів. Він має наступні властивості:

- **id**: унікальний ідентифікатор товару. Це рядок, що ідентифікує конкретний товар в системі.
- **name**: назва товару. Це рядок, що описує назву товару.
- **description**: опис товару. Це рядок, що містить детальний опис характеристик та особливостей товару.
- **price**: ціна товару. Це десяткове число, що вказує на вартість товару.
- **category**: категорія товару. Це рядок, що визначає до якої категорії належить товар (наприклад, електроніка, одяг, аксесуари тощо).
- **colors**: список кольорів товару. Це список рядків, що містить доступні кольори для даного товару.

Цей клас дозволяє зберігати та отримувати інформацію про конкретний товар з каталогу, таку як його ідентифікатор, назва, опис, ціна, категорія та доступні кольори. Використовуючи об'єкти цього класу, можна ефективно управляти та представляти дані про товари в системі.

Дані про користувачів містяться у полі **users** (див. рис. 2.22).

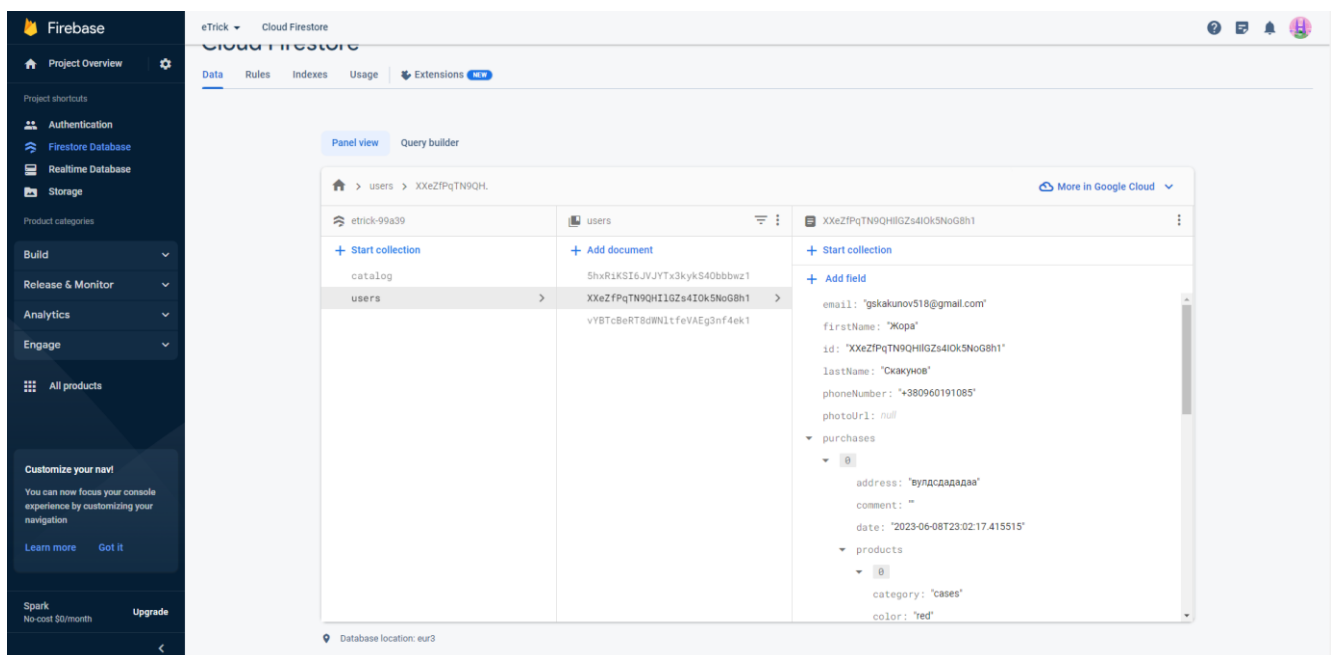


Рисунок 2.22 – Структура користувачів у базі даних Firestore

Клас `AppUser` представляє користувача додатку. Він має наступні властивості:

- `id`: унікальний ідентифікатор користувача. Це рядок, що ідентифікує конкретного користувача в системі.
- `email`: електронна адреса користувача. Це рядок, що містить поштову адресу користувача.
- `firstName`: ім'я користувача. Це рядок, що містить перше ім'я користувача.
- `lastName`: прізвище користувача. Це рядок, що містить прізвище користувача.
- `phoneNumber`: номер телефону користувача (необов'язково). Це рядок, що містить номер телефону користувача.
- `photoUrl`: URL-адреса фотографії користувача (необов'язково). Це рядок, що містить посилання на фотографію користувача.
- `purchases`: список покупок користувача. Це список об'єктів типу `Purchase`, який містить інформацію про покупки, зроблені користувачем.

Цей клас дозволяє зберігати та отримувати інформацію про користувача додатку, таку як його ідентифікатор, електронна адреса, ім'я, прізвище, номер телефону, фотографія та список покупок. Використовуючи об'єкти цього класу, можна ефективно управляти та представляти дані про користувачів в системі.

Для написання коду мобільного застосунку "eTrick" для продажу мобільних аксесуарів було обрано фреймворк Flutter. Далі наведено основні кроки для ініціалізації процесу розробки та налаштування мобільного застосунку:

- Flutter SDK повинен бути завантажений та налаштований для подальшої розробки.
- Створити новий проект Flutter у Visual Studio Code.
- Основні структури та файли будуть створені автоматично.
- У файл «`pubspec.yaml`» повинні бути вписані необхідні залежності для інтеграції з Firebase.

- Налаштування Firebase Authentication: Використовуючи Firebase Authentication, налаштувати механізм автентифікації для мобільного додатку "eTrick". Це дозволить користувачам створювати облікові записи, увійти до системи та забезпечити безпеку їх даних.

- Інтеграція з Firebase Firestore: Використовуючи Firebase Firestore, відбувається налаштування бази даних для зберігання та отримання даних для мобільного додатку "eTrick". Створюються необхідні колекції та документи для зберігання інформації про продукти, замовлення, користувачів та інше. Встановлюються правила доступу до даних, щоб забезпечити безпеку та обмеження доступу до важливої інформації.

- Розробка функціональності застосунку: Розроблюється функціональність мобільного додатку "eTrick" для перегляду каталогу аксесуарів, здійснення замовлень, оплати та отримання повідомлень про стан замовлення. Використовується Flutter фреймворк та мову програмування Dart для реалізації інтерфейсу користувача, взаємодії з Firebase.

Беручи до уваги кроки вище, процес розробки дозволяє створити та розгорнути мобільний додаток "eTrick" для продажу мобільних аксесуарів, забезпечуючи користувачам зручний і безпечний спосіб вибору, замовлення та оплати аксесуарів прямо з їх смартфона.

Для тестування мобільного застосунку "eTrick" для продажу мобільних аксесуарів було проведено низку тестів для перевірки відповідності до вимог:

- Перевірка коректності функції пошуку.
- Перевірка змісту та опису товарів.
- Перевірка функції додавання товару у кошик та коректність відображення товарів.
- Перевірка функції попереднього завантаження зображень.
- Перевірка сповіщень для користувача.
- Перевірка функції авторизації та створення нового користувача.

У таблиці 2.9 представлено тестування мобільного застосунку "eTrick".

Таблиця 2.9 – Тестування мобільного застосунку "eTrick".

Дія	Очікуваний результат	Результат
Пошук товару	Відображення товару, що відповідають критеріям пошуку	Товар успішно відображається
Перегляд детальної інформації про аксесуар	Відображення повної інформації про обраний аксесуар	Опис товару досить детальний та успішно відображається
Додавання аксесуарів до кошика	Додавання аксесуару до списку покупок в кошику	Товар успішно додається до кошику
Перегляд замовлення в кошику	Відображення списку аксесуарів та загальної суми в кошику	Товар коректно відображається у кошику
Вибір способу доставки	Можливість обрати спосіб доставки	Вибір способу доставки коректно працює
Отримання сповіщення про стан замовлення	Отримання сповіщення з інформацією про стан замовлення	Сповіщення успішно відображаються
Редагування списку товарів	Зміна інформації про аксесуари та їх характеристики	Редагування товару коректно працює
Додавання або видалення фото товару	Завантаження або видалення фотографій для обраного аксесуару	Додавання товару коректно працює
Редагування опису товару	Зміна опису та характеристик обраного аксесуару	Редагування опису товару коректно працює

3 РЕЗУЛЬТАТИ РОЗРОБКИ МОБІЛЬНОГО ЗАСТОСУНКУ "eTrick" ДЛЯ ПРОДАЖУ МОБІЛЬНИХ АКСЕСУАРІВ

Результатами кваліфікаційної роботи став мобільний застосунок "eTrick" для продажу мобільних аксесуарів. Сам мобільний застосунок повністю відповідає усім вимогам та працює відповідно. "eTrick" пройшов тестування та надає користувачам відповідний сервіс для покупки мобільних аксесуарів. Забезпечуючи користувачів швидким доступом до списку товарів та доступом до мобільного застосунку навіть при відсутності зв'язку Інтернет.

Нижче наведені основні результати після розробки мобільного застосунку "eTrick" для продажу мобільних аксесуарів:

- Мобільний застосунок: У процесі кваліфікаційної роботи було проведено розробку інтуїтивного інтерфейсу користувача та на його основі і було розроблено мобільний застосунок. Користувачі мають змогу оформити замовлення, переглядати товари та мати доступ до повного функціоналу без доступу до мережі Інтернет.

- Взаємодія з базою даних Firebase: Для збереження даних користувачів та контенту усього мобільного застосунку було використано базу даних Firebase. Зокрема було використано Firestore для даних про товар та даних користувачів, а для збереження зображень було використано Storage. Саме використання модулів описаних вище дало змогу досягти достатньої швидкості взаємодії з базою даних та збереження даних користувачів.

- Тестування: Мобільний застосунок був детально протестований. Всі відповідні та додаткові функції було перевірено та відсутність будь-яких інших була підтверджена.

Розмір мобільного застосунку "eTrick" для продажу мобільних аксесуарів:

Мобільний застосунок "eTrick" для продажу мобільних аксесуарів складається з 4279 рядків коду, написаних за допомогою мови Dart та фреймворку

Flutter. Розмір коду на дисковому просторі 220kB. Сам застосунок займає 22.3 MB.

Результати випробування мобільного застосунку

Мобільний застосунок "eTrick" для продажу мобільних аксесуарів було випробувано згідно з методикою випробовувань наведених у Додатку Д. Нижче приведені результати випробувань:

На рисунках 3.1 – 3.6 продемонстровано результати випробування мобільного застосунку "eTrick".

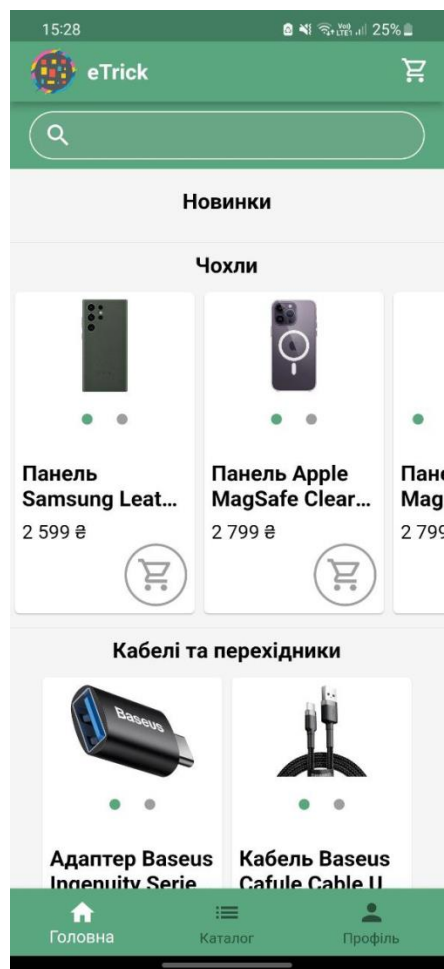


Рисунок 3.1 – Головна сторінка мобільного застосунку "eTrick"

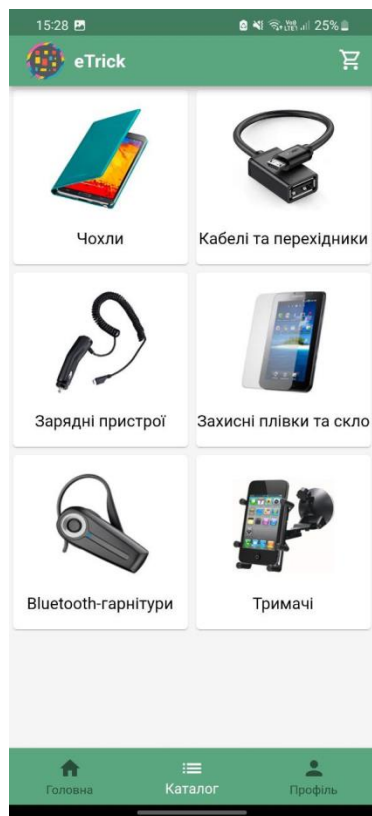


Рисунок 3.2 – Категорії товару мобільного застосунку "eTrick"

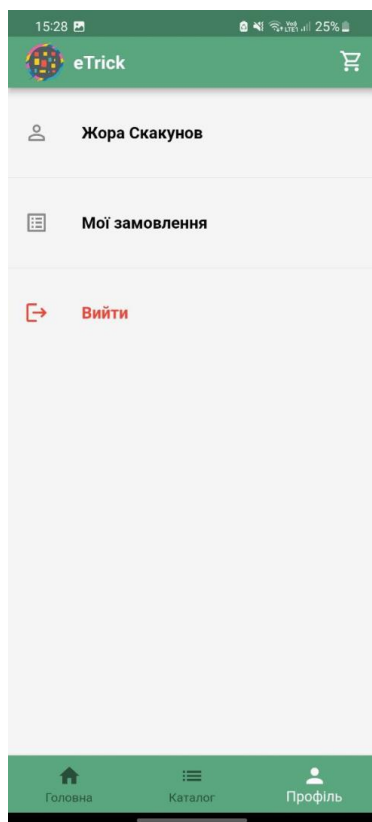


Рисунок 3.3 – Профіль користувача мобільного застосунку "eTrick"

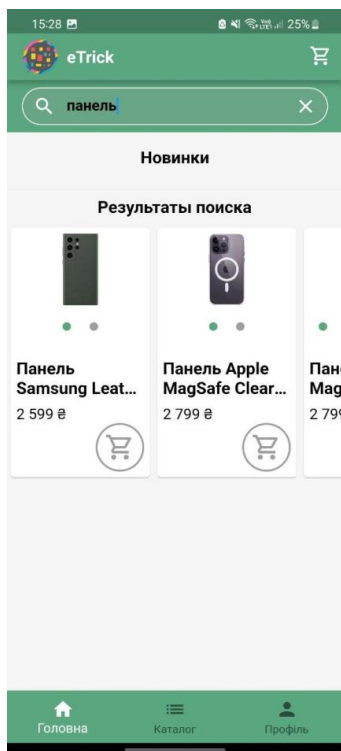


Рисунок 3.4 – Тестування функції пошук товару

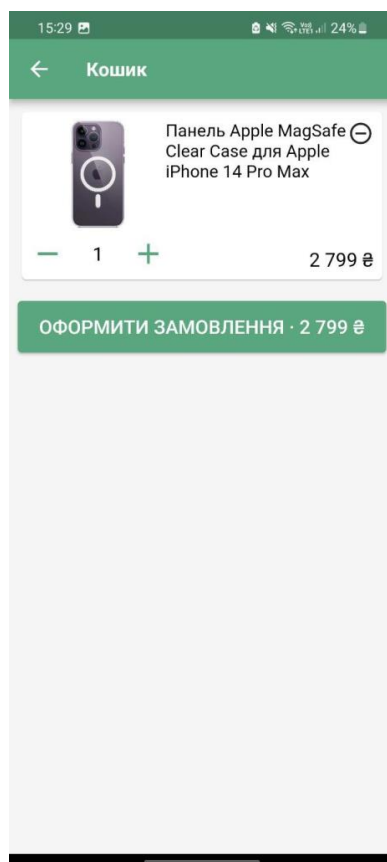


Рисунок 3.5 – Тестування функції додавання товару у кошик

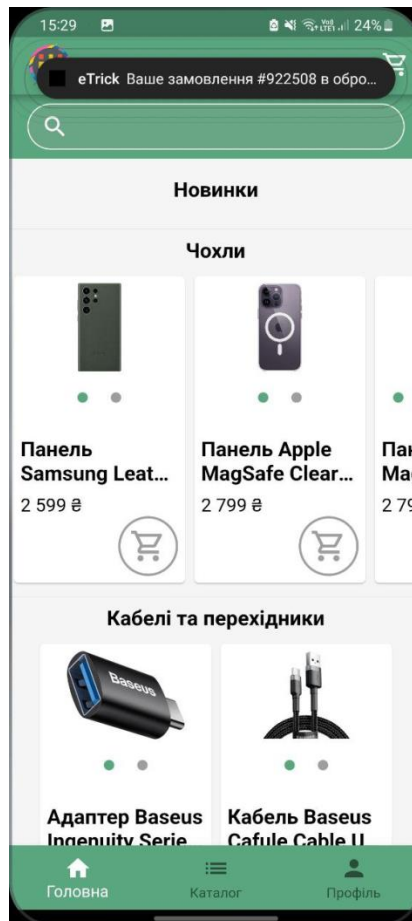


Рисунок 3.6 – Тестування функції отримання сповіщень

У результаті тестування мобільного застосунку "eTrick" для продажу мобільних аксесуарів було підтверджено повну функціональну відповідність, а саме:

- Коректне відображення усіх сторінок продуктів, головної, каталогу та сторінки профілю.
- Коректна робота функції передзавантаження зображень для роботи мобільного застосунку при відсутності мережі Інтернет.
- Коректне функціонування додавання товару у кошик.
- Правильна робота сповіщень, щодо статусу замовлення.

Аналізуючи вищесказане, можна зробити висновок, що було досягнуто мету кваліфікаційної роботи, а саме розробка мобільного застосунку "eTrick" для продажу мобільних аксесуарів.

4 ОХОРОНА ПРАЦІ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

За допомогою охорони праці вирішується наступні завдання:

- інженерно-технічне – направлено на зменшення ризиків при роботі, за допомогою впровадження нових технологій та заміни матеріалів на менш небезпечні;

- соціальне – направлено на відшкодування шкоди, яка була завдана під час трудового процесу, внаслідок нещасних випадків, та захисту прав працівника.

Усі права працівника на безпечні умови праці захищаються законом України “Про охорону праці” [11].

Охорона праці є важливим аспектом будь-якого робочого середовища і забезпечує захист працівників від шкідливих і небезпечних факторів, пов'язаних з їхньою трудовою діяльністю. Вона базується на системі правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів.

Інженерно-технічне завдання охорони праці спрямоване на запобігання та мінімізацію ризиків для працівників. Для цього впроваджуються нові технології, застосовуються безпечні матеріали, розробляються ергономічні робочі місця, встановлюються системи вентиляції та освітлення, проводяться навчання та тренінги з питань безпеки та здоров'я.

Соціальне завдання охорони праці полягає у відшкодуванні шкоди, яка була завдана працівникам у результаті нещасних випадків або професійних захворювань. Це включає медичне обслуговування, реабілітацію, компенсаційні виплати, страхування, соціальну підтримку та захист прав працівників.

Охорона праці регулюється законом України "Про охорону праці", який визначає права та обов'язки працівників і роботодавців у сфері охорони праці. Закон встановлює вимоги до організації робочого процесу, проведення оцінки ризиків, забезпечення безпеки та здоров'я працівників, надання медичної допомоги, розслідування нещасних випадків тощо.

Важливим елементом охорони праці є система нагляду та контролю, що забезпечує виконання вимог охорони праці і запобігає порушенням. Державні органи, які відповідають за охорону праці, здійснюють перевірки підприємств, надають консультації та рекомендації з питань безпеки та здоров'я працівників.

Отже, охорона праці є системою заходів, спрямованих на забезпечення безпеки, здоров'я та працездатності працівників під час трудової діяльності. Вона включає інженерно-технічні та соціальні заходи, регулюється законодавством та передбачає нагляд і контроль з боку відповідних органів.

4.1 Аналіз шкідливих та небезпечних факторів під час роботи з мобільними пристроями

Існують небезпечні та шкідливі фактори роботи з мобільним телефоном які потрібно знати. До головних можна віднести роботу з самим мобільним телефоном та робоче місце працівника [12].

Нижче наведено деякі з них:

1) Вплив на зір: Використання мобільних пристроїв може призводити до напруження очей, сухості, подразнення та втоми. Довготривале спрямоване уваги на екран може спричинити втому очей, головний біль та навіть проблеми зі зріом. Рекомендується регулярно робити перерви, виконувати вправи для очей та налаштовувати яскравість та контрастність екрану на комфортний рівень.

2) Постійний стрес: Спостереження за мобільними пристроями може призводити до постійного стресу, особливо в ситуаціях, коли потрібно бути постійно на зв'язку або доступним. Це може впливати на психічне здоров'я та загальний стан. Важливо встановити режими відпочинку, вимкнути сповіщення,

коли вони необхідні, і виробити звичку розслаблятися і відпочивати від мобільного пристрою.

3) Негативний вплив на позу: Використання мобільних пристроїв може приводити до поганої пози тіла, особливо коли людина схиляється вперед або незручно тримає пристрій. Це може призводити до болю в шії, спині та плечах. Важливо дотримуватися правильної пози, тримати пристрій на відстані від очей та регулярно робити перерви для розтягування та вправ для м'язів.

4) Вплив електромагнітного випромінювання: Мобільні пристрої випромінюють електромагнітне випромінювання, яке може бути шкідливим для здоров'я в довготривалій перспективі. Хоча наукові дослідження не дають однозначних доказів про шкідливість електромагнітного випромінювання на низьких рівнях, деякі люди можуть відчувати негативний вплив. Можна зменшити експозицію, використовуючи гарнітуру або вбудовані функції гучномовця, а також обмежуючи тривалість розмов та тримаючи пристрій подалі від тіла.

5) Залежність від мобільних пристроїв: Часте використання мобільних пристроїв може призводити до залежності, яка може негативно позначитися на психічному стані та міжособистових відносинах. Важливо обмежувати час, проведений з мобільним пристроєм, встановлювати правила для себе і приймати свідомі рішення щодо використання пристроїв.

Норми мікроклімату зазначені у санітарних нормах мікроклімату виробничих приміщень ДСН 3.3.6.042-99 [13].

Температура повітря може відрізнятись від норм залежно від пори року та інших умов. Щоб уникнути небажаних відхилень потрібно встановлювати допоміжні засоби регулювання мікроклімату: кондиціонери, обігрівачі тощо.

Найбільш прийнятними методами захисту від шуму є використання акустичних екранів і звукопоглинальних облицювань.

Акустичний екран є перешкодою для звукових хвиль, що знижує рівень звуку шляхом утворення акустичної тіні за екраном в зоні розташування робочого місця.

Ергономічна безпека праці досягається за допомогою нескладних правил. Спина людини повинна бути нахилена назад під кутом в декілька градусів для збільшення кута між тулубом і стегнами, посилення кровообігу і зменшення тиску на хребет. Руки розслаблені й вільно опущені уздовж боків, передпліччя і кисті розташовані паралельно підлозі. Стегна знаходяться під прямим кутом до тулуба. Коліна - під прямим кутом до стегон.

Спинка стільця повторює лінію вигину нижньої частини спини. Сидіння злегка нахилене вперед для перенесення тиску з хребта на стегна і ноги. Край подушки сидіння заломлений вниз для ослаблення тиску на стегна.

Максимальний час безперервної роботи з телефоном не повинен перевищувати 2-х годин в день. Через кожні 7 хвилин потрібно робити коротку перерву на 10-15 секунд. Можна подивитися у вікно (у далечінь) або на картину з приємним для очей пейзажем, яку необхідно повісити за комп'ютером. У картині повинні переважати неяскраві, заспокійливі зір кольори, такі, наприклад, як зелені, блакитні. Також потрібно робити декілька простих вправ для очей.

2) Вимоги безпеки після закінчення роботи:

2.1) Після закінчення роботи фахівець зобов'язаний: відключити від електромережі мобільний телефон, засоби оргтехніки та інше устаткування, крім обладнання, яке використовується цілодобово (апарати факсимільного зв'язку, мережеві сервери тощо); упорядкувати робоче місце, провітрити приміщення; привести себе у порядок, вимити руки й обличчя;

2.2) Зберегти інформацію в телефоні;

2.3) Прибрати робоче місце.

4.3 Методи нейтралізації або зменшення впливу негативних факторів під час роботи зі мобільними пристроями

Нейтралізація шкідливих та небезпечних факторів при роботі з мобільними пристроями може бути досягнута за допомогою наступних заходів:

1) Зменшення ефекту на зір:

- Регулярно робіть перерви під час роботи з мобільним пристроєм. Загалом рекомендується робити перерву кожні 20-30 хвилин і протягом 5-10 хвилин відводити погляд від екрану;

- Виконуйте вправи для очей, такі як погляд у далечину, обертання очей у різні напрямки і покладання теплих компресів на очі;

- Правильно налаштуйте яскравість, контрастність та розмір шрифту на своєму пристрої для забезпечення комфортного читання;

2) Менеджмент стресу:

- Встановіть часові обмеження для використання мобільних пристроїв і дотримуйтесь їх;

- Вимкніть сповіщення, коли вони необхідні, або встановіть режим "Не турбувати" під час важливих зустрічей або відпочинку;

- Займіться релаксаційними практиками, такими як медитація, глибоке дихання або йога, для зняття стресу і напруження;

3) Збереження правильної пози:

- Підтримуйте правильну позу тіла під час використання мобільного пристрою. Завжди стежте за положенням голови, шиї та спини;

- Використовуйте підставку або тримайте пристрій на такій висоті, щоб ваша голова була в природному положенні і ви могли дивитися прямо на екран без похилення голови;

4) Зменшення впливу електромагнітного випромінювання:

- Використовуйте гарнітуру або вбудовані функції гучномовця, щоб зменшити близький контакт пристрою з головою під час розмов;

- Обмежте тривалість розмов по мобільному телефону і використовуйте функцію високоякісного звуку (HD Voice) для зниження рівня експозиції;

5) Керування залежністю від мобільних пристроїв:

- Встановіть правила для себе щодо використання мобільних пристроїв, наприклад, обмеження часу, проведеного з пристроєм, або встановлення безпосередніх "безмобільних" періодів;

- Підтримуйте здорові звички, такі як виконання фізичних вправ, соціальні контакти та заняття хобі, які не пов'язані з мобільними пристроями;

Загалом, баланс і свідоме використання мобільних пристроїв, разом з регулярними перервами і здоровими звичками, допоможуть нейтралізувати більшість шкідливих та небезпечних факторів, пов'язаних з роботою з мобільними пристроями.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи була вирішена практична задача інженерії програмного забезпечення, яка характеризується комплексністю та невизначеністю умов, у ході її виконання був розроблений мобільний застосунок "eTrick" для продажу мобільних аксесуарів – зручна та інноваційна платформа, яка надає користувачам можливість здійснювати онлайн-покупки аксесуарів для їхніх мобільних телефонів. Мету роботи було досягнуто.

Аналізуючи розроблений застосунок, можна зробити наступні висновки:

- Застосунок "eTrick" має привабливий та інтуїтивно зрозумілий інтерфейс, що сприяє зручному користуванню та швидкому доступу до багатого асортименту аксесуарів. Важливим аспектом є його респонсивний дизайн, що дозволяє користувачам зручно використовувати застосунок на різних пристроях, включаючи смартфони та планшети.

- Функціональність системи пошуку та фільтрації дозволяє користувачам швидко знаходити необхідні товари, з урахуванням їхніх переваг та характеристик. Розширені можливості фільтрації, такі як виробник, ціновий діапазон, оцінки користувачів та інші параметри, допомагають збільшити точність пошуку та зробити процес вибору більш персоналізованим.

- Наявність функціональності кошика покупок дозволяє користувачам збирати необхідні товари перед остаточним замовленням. Вони можуть переглядати та змінювати свій кошик, додавати або видаляти товари, а також керувати кількістю замовлених товарів.

У процесі розробки мобільного застосунку були використані такі інструменти: Flutter, Dart та Firebase. Цей інструментарій допоміг забезпечити функціональний та зручний процес розробки мобільного застосунку. Після впровадження застосунком користуються люди для швидкого та інтуїтивно зрозумілого оформлення замовлення.

В ході кваліфікаційної роботи було виконано розділ з охорони праці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Flutter. Build apps for any screen. URL: <https://flutter.dev/> (дата звернення: 23.05.2023).
2. Developer documentation for Firebase. URL: <https://firebase.google.com/docs> (дата звернення: 23.05.2023).
3. Flutter documentation. URL: <https://flutter.dev/docs> (дата звернення: 23.05.2023).
4. Rozetka – інтернет-магазин «Розетка™». URL: <https://play.google.com/store/apps/details?id=ua.com.rozetka.shop> (дата звернення: 23.05.2023).
5. ЦИТРУС – інтернет-магазин, гаджети та аксесуари. URL: <https://play.google.com/store/apps/details?id=com.citrusapp&hl=ru&gl=US&pli=1> (дата звернення: 23.05.2023).
6. Діаграма розгортання – Призначення діаграми розгортання. URL: https://vuzlit.com/1009782/diagrama_rozgortannya (дата звернення: 12.06.2023).
7. Діаграма діяльності – Що таке діаграма діяльності. URL: <https://uk.theastrologypage.com/activity-diagram> (дата звернення: 12.06.2023).
8. Діаграма класів – Елементи UML. URL: <https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml-elements.html> (дата звернення: 12.06.2023).
9. Visual Studio Code. Code editing. URL: <https://code.visualstudio.com> (дата звернення: 14.06.2023).
10. Dart. Dart is a client-optimized language for fast apps on any platform. URL: <https://dart.dev> (дата звернення: 14.06.2023).
11. Про охорону праці: Закон України від 21.11.2002 р. № 229-IV. Дата оновлення: 04.02.2021. URL: <https://zakon.rada.gov.ua/laws/main/2694-12#Text>. (дата звернення: 23.05.2023).

12. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників: Закон України від 14.02.2018 № 207 Дата оновлення: 17.06.2015. URL: <https://zakon.rada.gov.ua/laws/show/z0508-18#n14> (дата звернення: 23.05.2023).

13. Санітарні норми мікроклімату виробничих приміщень (ДСН 3.3.6.042-99). URL: <https://zakon.rada.gov.ua/rada/show/va042282-99#Text> (дата звернення: 23.05.2023).

Додаток А – Технічне завдання на розробку мобільного застосунку "eTrick" для продажу мобільних аксесуарів

Вступ

Повна назва програмного продукту: мобільний застосунок для продажу мобільних аксесуарів "eTrick".

Сфера застосування:

Мобільний застосунок "eTrick" має широкий спектр застосувань у сферах, пов'язаних з електронною комерцією, маркетингом, тощо.

1. Підстави для розробки:

Підставою для розробки даного програмного забезпечення є наказ на затвердження тем кваліфікаційних робіт бакалаврів зі спеціальності 121 Інженерія програмного забезпечення в Національному університеті кораблебудування ім. адмірала Макарова.

2. Призначення розробки

1.1 Експлуатаційне призначення

Експлуатаційним призначенням мобільного застосунку є пошук та замовлення аксесуарів для мобільних телефонів.

1.2 Функціональне призначення

Користувачі можуть:

- Переглядати список товарів та їх характеристики;
- Отримувати детальну інформацію про будь-який товар представлений у застосунку;
- Збирати кошик товарів та мати можливість їх замовити;
- Мати можливість отримувати дані про доставку товару та сповіщення, щодо зміни статусу оброблення замовлення;
- Мати можливість отримувати сповіщення;

- Мати можливість отримувати інформацію про доставку.

Адміністратор може:

- Мати можливість редагувати список товарів;
- Мати можливість перевіряти та видаляти коментарі під товарами;
- Мати можливість додавати або видаляти фото товарів;
- Мати можливість редагувати описи будь-яких товарів;

3. Вимоги до програми або програмного виробу

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

Функції користувача:

- Автентифікація: Користувачі можуть мати власні акаунти для відстежування власних замовлень та отримування персоналізованих сповіщень.
- Пошук і фільтрацію продуктів: Користувачі можуть швидко знайти потрібні аксесуари, використовуючи функцію пошуку або застосовуючи фільтри за категорією, брендом, ціною тощо;
- Детальні описи продуктів: Кожен аксесуар має свій власний сторінку з описом, зображеннями, характеристиками, відгуками користувачів та рекомендаціями;
- Кошик і оформлення замовлення: Користувачі можуть додавати продукти до кошика, переглядати замовлення, вибирати способи доставки та оплати;
- Система оглядів і рейтингів: Користувачі можуть залишати відгуки та оцінки для продуктів, що дозволяє іншим покупцям зробити освідчений вибір;
- Оповіщення і відстеження замовлень: Користувачі можуть отримувати сповіщення про стан своїх замовлень, включаючи підтвердження покупки, статус доставки та інформацію про повернення;

3.1.2 Вимоги до організації вхідних та вихідних даних

- Вхідні дані:

- Автентифікація або реєстрація користувача: Користувач має надати актуальну електронну адресу та пароль;
- Дані для доставки товару: актуальна адреса або номер поштового відділення, зручний час доставки;
- Дані для оплати товару: номер кредитної або дебетової карти, акаунт у Google або Apple pay.

Вихідні дані:

- Застосунок показує зібрані у кошик товари;
- Застосунок відображає повний і актуальний список бажань;
- Застосунок зберігає Вподобані товари;
- Застосунок надає дані про доставку товару: статус, номер замовлення, сума до сплати, тощо;
- Застосунок відображає відгуки і рейтинг товару.

3.2 Вимоги до надійності

3.2.1 Вимоги до забезпечення надійного (сталого) функціонування програми.

Надійне (стійке) функціонування програми має бути забезпечене виконанням сукупності організаційно-технічних заходів, перелік яких подано нижче:

- захист від некоректних дій користувача(під час розпізнавання);
- встановлення Flutter Framework не нижче версії 2.18.6.

3.2.3 Відмови через некоректні дії оператора

Відмова можлива при наступних умовах:

- Висока навантаженість: Якщо програмне забезпечення стикається з надмірним навантаженням або несподівано великою кількістю запитів, це може спричинити відмову через перевантаження ресурсів;
- Непередбачені вхідні дані: Якщо програмне забезпечення отримує некоректні або непередбачені дані від користувачів, це може призвести до відмови або некоректної роботи програми.

3.3 Умови експлуатації

3.3.1 Кліматичні умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні відповідати вимогам, що пред'являються до технічних засобів в частині умов їх експлуатації.

3.3.2 Вимоги до чисельності та кваліфікації користувачів

Користувач повинен мати базові навички роботи з мобільним телефоном.

3.4 Вимоги до складу і параметрів технічних засобів

Система повинна задовольняти вказаним вимогам на мобільному пристрої наступної мінімальної комплекції:

- Процесор серії Exynos 990 Qualcomm Snapdragon 865;
- Оперативна пам'ять 2 ГБ;
- Дискового простору 100 МБ для встановлення програми;
- Роздільна здатність екрану 1280x800 пікселів;
- Інтернет-з'єднання: Застосунок вимагає доступу до Інтернету для завантаження інформації про продукти, оновлення даних та здійснення покупок. Доступ до Wi-Fi або мобільного Інтернету (3G, 4G, 5G) є необхідним.

3.5 Вимоги до інформаційної та програмної сумісності

Системні програмні засоби, які використовуються програмою, повинні бути представлені ліцензійною версією операційної системи Android 8.0 і вище.

Вимоги до захисту інформації та програм не пред'являються

3.6 Спеціальні вимоги

Програма повинна забезпечувати взаємодію з користувачем за допомогою графічного інтерфейсу користувача, розробленого відповідно до рекомендацій компанії-виробника операційної системи.

4. Вимоги до програмної документації

Склад програмної документації повинен включати в себе:

- технічне завдання;
- програму та методику випробувань;

- керівництво користувача;
- опис програми;
- код програми.

5. Техніко-економічні показники

У даній кваліфікаційній роботі техніко-економічні показники не розраховуються

6. Етапи роботи

Етапи роботи представлені у таблиці А.1

Таблиця А.1 – Етапи роботи

Номер	Назва етапів роботи	Терміни виконання
1.	Аналіз практичної задачі інженерії ПЗ	30.03.2023
2.	Аналіз вимог до ПЗ	05.04.2023
3.	Розробка архітектури ПЗ	20.04.2023
4.	Розробка проекту ПЗ	04.05.2023
5.	Реалізація та тестування ПЗ	13.05.2023

7. Порядок контролю та приймання

Контроль за аналізом та проектуванням кожної окремої частини програмного забезпечення на кожному етапі з урахуванням вимог, визначених у технічному завданні. Кожна стадія розробки повинна бути представлена в зазначені строки та узгоджена із замовником.

Прийом проводяться відповідно до програми і методики випробувань і документують за допомогою протоколу проведення випробувань. У разі знаходження помилок під час прийому програмного виробу складається акт про знайдені помилки, який підписуються представниками замовника і розробника і затверджуються керівником організації – замовника та організації – розробника. Розробник повинен на протязі не більше ніж 2 тижнів виправити зазначені помилки і оповістити замовника про повторне проведення перевірки.

Додаток Б – Вихідні тексти програмних модулів

main.dart – файл який є точкою входу в додаток Flutter який встановлює необхідні залежності, ініціалізує Firebase і локальні сповіщення, а також налаштовує провайдерів і маршрути для програми Flutter.

```
import 'package:etrick/models/app_user.dart';
import 'package:etrick/models/cart_model.dart';
import 'package:etrick/models/catalog_model.dart';
import 'package:etrick/providers/bottom_navigation_provider.dart';
import 'package:etrick/providers/notification_provider.dart';
import 'package:etrick/providers/search_provider.dart';
import 'package:etrick/providers/shared_preferences_provider.dart';
import 'package:etrick/providers/theme_provider.dart';
import 'package:etrick/services/app_router.dart';
import 'package:etrick/services/auth_service.dart';
import 'package:etrick/app_theme.dart';
import 'package:etrick/services/utils.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:flutter_local_notifications/flutter_local_notifications.dart';
import 'package:permission_handler/permission_handler.dart';
import 'package:provider/provider.dart';
import 'package:shared_preferences/shared_preferences.dart';

import 'constants.dart';
import 'firebase_options.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  var instance = await SharedPreferences.getInstance();

  final flutterLocalNotificationsPlugin = FlutterLocalNotificationsPlugin();
  const initializationSettingsAndroid =
    AndroidInitializationSettings('@mipmap/ic_launcher');
```

```

const initializationSettings = InitializationSettings(
  android: initializationSettingsAndroid,
);
await flutterLocalNotificationsPlugin.initialize(initializationSettings);
PermissionStatus status = await Permission.notification.request();
Utils.log('Permission status: $status');
runApp(
  MultiProvider(
    providers: [
      Provider(
        create: (_) => SharedPreferencesProvider(instance),
      ),
      Provider(
        create: (_) => NotificationProvider(
          flutterLocalNotificationsPlugin,
        ),
      ),
      Provider(
        create: (_) => AuthService(),
      ),
      ChangeNotifierProvider(
        create: (_) => AppThemeProvider(),
      ),
      ChangeNotifierProvider(
        create: (_) => AppUserModel(),
      ),
      ChangeNotifierProvider(
        create: (_) => CatalogModel(items: []),
      ),
      ChangeNotifierProvider(
        create: (_) => SearchProvider(),
      ),
      ChangeNotifierProxyProvider<CatalogModel, CartModel>(
        create: (_) => CartModel(),
        update: (context, catalog, cart) {
          if (cart == null) {
            throw ArgumentError.notNull('cart');
          }
          cart.catalog = catalog;
          cart.loadCart(context);
          return cart;
        },
      ),
      ChangeNotifierProvider(
        create: (_) => BottomNavigationProvider(),

```

```

        ),
      ],
      child: const MyApp(),
    ),
  );
}

class MyApp extends StatefulWidget {
  const MyApp({super.key});

  @override
  State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  late ValueListenable<User?> userChanges;
  @override
  void initState() {
    final authService = context.read<AuthService>();
    authService.initialize();
    userChanges =
      authService.authStateChanges.toValueNotifier(authService.user);
    super.initState();
    if (userChanges.value != null && userChanges.value!.uid.isNotEmpty) {
      Utils.fetchData(context);
    }
  }
}

@override
Widget build(BuildContext context) {
  final router = AppRouter(userChanges).router;
  return Consumer<AppThemeProvider>(
    builder: (context, value, child) => MaterialApp.router(
      title: Constants.appName,
      theme: AppTheme.lightTheme,
      routeInformationProvider: router.routeInformationProvider,
      routeInformationParser: router.routeInformationParser,
      routerDelegate: router.routerDelegate,
      debugShowCheckedModeBanner: false,
    ),
  );
}
}

```

create_account_page.dart – це віджет Flutter, який представляє собою сторінку для створення облікового запису користувача в додатку.

```
import 'package:etrick/app_theme.dart';
import 'package:etrick/constants.dart';
import 'package:etrick/models/app_user.dart';
import 'package:etrick/models/catalog_model.dart';
import 'package:etrick/services/auth_service.dart';
import 'package:etrick/services/firestore_service.dart';
import 'package:etrick/services/storage_service.dart';
import 'package:etrick/widgets/logo_widget.dart';
import 'package:flutter/material.dart';
import 'package:go_router/go_router.dart';
import 'package:provider/provider.dart';

class CreateAccountPage extends StatefulWidget {
  const CreateAccountPage({super.key});

  @override
  State<CreateAccountPage> createState() => _CreateAccountPageState();
}

class _CreateAccountPageState extends State<CreateAccountPage> {
  final firstNameController = TextEditingController();
  final lastNameController = TextEditingController();
  final emailController = TextEditingController();
  final passwordController = TextEditingController();

  final isEmailError = ValueNotifier<bool>(false);
  final isPasswordError = ValueNotifier<bool>(false);
  final isFirstNameError = ValueNotifier<bool>(false);
  final isLastNameError = ValueNotifier<bool>(false);
  final isPasswordVisible = ValueNotifier<bool>(false);

  void _toggleObscurePassword() {
    isPasswordVisible.value = !isPasswordVisible.value;
  }

  bool validateFields() {
    isEmailError.value = false;
    isPasswordError.value = false;
    isFirstNameError.value = false;
    isLastNameError.value = false;

    final auth = context.read<AuthService>();
```

```

if (!auth.isFirstNameValid(firstNameController.text.trim())) {
    isFirstNameError.value = true;
    return false;
}
if (!auth.isLastNameValid(lastNameController.text.trim())) {
    isLastNameError.value = true;
    return false;
}
if (!auth.isEmailValid(emailController.text.trim())) {
    isEmailError.value = true;
    return false;
}
if (!auth.isPasswordValid(passwordController.text)) {
    isPasswordError.value = true;
    return false;
}
return true;
}

Future<void> _createAccount() async {
    if (!validateFields()) {
        ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(
                content: Text('Будь ласка, введіть коректні дані'),
            ),
        );
        return;
    }

    final String firstName = firstNameController.text.trim();
    final String lastName = lastNameController.text.trim();
    final String email = emailController.text.trim();
    final String password = passwordController.text;

    final appUserModel = context.read<AppUserModel>();
    final auth = context.read<AuthService>();
    final catalog = context.read<CatalogModel>();

    final result = await context.read<AuthService>().createAccount(
        AppUser(
            firstName: firstName,
            lastName: lastName,
            email: email,
            id: "",
        ),
    ),

```

```

        password,
    );

    bool success = result.keys.first;
    String e = result.values.first;
    if (!success && mounted) {
        ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(
                content: Text(auth.getFirebaseAuthErrorMessage(e)),
            ),
        );
    }
    appUserModel.user = await FirestoreService.getUserById(auth.user!.uid);
    catalog.items = await FirestoreService.getCatalog();
    await StorageService.downloadCatalogPictures(catalog, catalog.items);
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: const Text('Створити обліковий запис'),
            centerTitle: true,
        ),
        body: SafeArea(
            child: ListView(
                padding: const EdgeInsets.symmetric(
                    horizontal: 16,
                    vertical: 40,
                ),
                children: [
                    const LogoWidget(),
                    const SizedBox(height: 32.0),
                    ValueListenableBuilder(
                        valueListenable: isFirstNameError,
                        builder: (context, value, child) => TextField(
                            controller: firstNameController,
                            decoration: InputDecoration(
                                errorText:
                                    value ? 'Будь ласка, введіть правильне ім\'я' : null,
                                labelText: 'Ім\'я',
                            ),
                        ),
                    ),
                    const SizedBox(height: 8.0),
                ],
            ),
        ),
    );
}

```

```

ValueListenableBuilder(
  valueListenable: isLastNameError,
  builder: (context, value, child) => TextField(
    controller: lastNameController,
    decoration: InputDecoration(
      errorText:
        value ? 'Будь ласка, введіть правильне прізвище' : null,
      labelText: 'Прізвище',
    ),
  ),
),
const SizedBox(height: 8.0),
ValueListenableBuilder(
  valueListenable: isEmailError,
  builder: (context, value, child) => TextField(
    controller: emailController,
    decoration: InputDecoration(
      errorText:
        value ? 'Будь ласка, введіть правильний email' : null,
      labelText: 'Email',
    ),
  ),
),
const SizedBox(height: 8.0),
ValueListenableBuilder(
  valueListenable: isPasswordError,
  builder: (context, passwordError, child) =>
    ValueListenableBuilder(
      valueListenable: isPasswordVisible,
      builder: (context, visible, child) => TextField(
        controller: passwordController,
        obscureText: !visible,
        decoration: InputDecoration(
          errorText: passwordError
            ? 'Будь ласка, введіть пароль (мін. 8 символів)'
            : null,
          labelText: 'Пароль',
          suffixIcon: IconButton(
            onPressed: _toggleObscurePassword,
            icon: Icon(
              visible ? Icons.visibility : Icons.visibility_off,
            ),
          ),
        ),
      ),
    ),
),

```

```

    ),
  ),
  const SizedBox(height: 24.0),
  ElevatedButton(
    onPressed: _createAccount,
    child: const Text('Створити обліковий запис'),
  ),
  const SizedBox(height: 16.0),
  TextButton(
    onPressed: () {
      context.go(Constants.loginLoc);
    },
    child: const Text(
      'Увійти',
      style: TextStyle(
        color: AppTheme.primaryColor,
        decoration: TextDecoration.underline,
      ),
    ),
  ),
],
),
);
}
}

```

login_page.dart – віджет Flutter під назвою LoginPage, який представляє собою сторінку для входу користувача в додаток.

```

import 'package:etrick/app_theme.dart';
import 'package:etrick/constants.dart';
import 'package:etrick/models/app_user.dart';
import 'package:etrick/models/catalog_model.dart';
import 'package:etrick/services/auth_service.dart';
import 'package:etrick/services/firestore_service.dart';
import 'package:etrick/services/storage_service.dart';
import 'package:etrick/widgets/logo_widget.dart';
import 'package:flutter/material.dart';
import 'package:go_router/go_router.dart';
import 'package:provider/provider.dart';

class LoginPage extends StatefulWidget {
  const LoginPage({super.key});

```

```

@override
State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();

  final isEmailError = ValueNotifier<bool>(false);
  final isPasswordError = ValueNotifier<bool>(false);
  final isPasswordVisible = ValueNotifier<bool>(false);

  void _toggleObscurePassword() {
    isPasswordVisible.value = !isPasswordVisible.value;
  }

  bool validateFields() {
    isEmailError.value = false;
    isPasswordError.value = false;

    final auth = context.read<AuthService>();
    if (!auth.isEmailValid(emailController.text.trim())) {
      isEmailError.value = true;
      return false;
    }
    if (!auth.isPasswordValid(passwordController.text)) {
      isPasswordError.value = true;
      return false;
    }
    return true;
  }

  Future<void> _login() async {
    if (!validateFields()) {
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
          content: Text('Будь ласка, введіть коректні дані'),
        ),
      );
      return;
    }

    final String email = emailController.text.trim();
    final String password = passwordController.text;
    final appUserModel = context.read<AppUserModel>();
  }

```

```

final auth = context.read<AuthService>();
final catalog = context.read<CatalogModel>();
final result = await auth.login(email, password);

bool success = result.keys.first;
String e = result.values.first;
if (!success && mounted) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
      content: Text(auth.getFirebaseAuthErrorMessage(e)),
    ),
  );
}

appUserModel.user = await FirestoreService.getUserById(auth.user!.uid);
catalog.items = await FirestoreService.getCatalog();
await StorageService.downloadCatalogPictures(catalog, catalog.items);
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Увійти'),
    ),
    body: SafeArea(
      child: ListView(
        padding: const EdgeInsets.symmetric(
          horizontal: 16,
          vertical: 40,
        ),
        children: [
          const LogoWidget(),
          const SizedBox(height: 32.0),
          ValueListenableBuilder<bool>(
            valueListenable: isEmailError,
            builder: (context, value, child) {
              return TextField(
                controller: emailController,
                decoration: InputDecoration(
                  labelText: 'Електронна пошта',
                  errorText: value ? 'Некоректна пошта' : null,
                ),
              );
            },
          ),
        ],
      ),
    ),
  );
}

```

```

),
const SizedBox(height: 8.0),
ValueListenableBuilder<bool>(
  valueListenable: isPasswordError,
  builder: (context, error, child) {
    return ValueListenableBuilder(
      valueListenable: isPasswordVisible,
      builder: (context, visible, child) => TextField(
        controller: passwordController,
        obscureText: !visible,
        decoration: InputDecoration(
          labelText: 'Пароль',
          errorText: error ? 'Некоректний пароль' : null,
          suffixIcon: IconButton(
            icon: Icon(
              visible ? Icons.visibility : Icons.visibility_off),
            onPressed: _toggleObscurePassword,
          ),
        ),
      ),
    ),
  ),
);
},
),
const SizedBox(height: 24.0),
ElevatedButton(
  onPressed: _login,
  child: const Text('Увійти'),
),
const SizedBox(height: 16.0),
TextButton(
  onPressed: () {
    if (context.canPop()) {
      context.pop();
    } else {
      context.go(Constants.createAccountLoc);
    }
  },
  child: const Text(
    'Зареєструватися',
    style: TextStyle(
      color: AppTheme.primaryColor,
      decoration: TextDecoration.underline,
    ),
  ),
),
),
),

```

```

const SizedBox(height: 32.0),
Align(
  alignment: Alignment.centerRight,
  child: RichText(
    text: TextSpan(
      style: Theme.of(context).textTheme.titleLarge!.copyWith(
        fontWeight: FontWeight.normal,
      ),
    children: [
      const TextSpan(
        text: 'Забули пароль?',
      ),
      const WidgetSpan(
        child: Padding(
          padding: EdgeInsets.only(left: 4),
        ),
      ),
      WidgetSpan(
        child: MouseRegion(
          cursor: SystemMouseCursors.click,
          child: GestureDetector(
            onTap: () {
              context.go(Constants.resetPasswordLoc);
            },
            child: Text(
              'Відновити тут',
              style: Theme.of(context)
                .textTheme
                .titleLarge!
                .copyWith(
                  fontWeight: FontWeight.bold,
                  color: AppTheme.primaryColor,
                  decoration: TextDecoration.underline,
                ),
            ),
          ),
        ),
      ),
    ],
  ),
),

```

```
);
}
}
```

home_page.dart – код який включає клас HomePage, який представляє головну сторінку домашнього екрану програми.

```
import 'package:etrick/models/catalog_model.dart';
import 'package:etrick/models/tab_item.dart';
import 'package:etrick/pages/home/navigation_pages/catalog/catalog_page.dart';
import
'package:etrick/pages/home/navigation_pages/home_main/home_main_page.dart';
import 'package:etrick/pages/home/navigation_pages/profile/profile_page.dart';
import 'package:etrick/services/auth_service.dart';
import 'package:etrick/services/utls.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';

import 'home_app_bar.dart';

final List<TabItem> tabBar = [
  TabItem(
    index: 0,
    title: 'Головна',
    icon: const Icon(Icons.home),
    page: const HomeMainPage(),
  ),
  TabItem(
    index: 1,
    title: 'Каталог',
    icon: const Icon(Icons.list),
    page: const CatalogPage(),
  ),
  TabItem(
    index: 2,
    title: 'Профіль',
    icon: const Icon(Icons.person),
    page: const ProfilePage(),
  ),
];

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
```

```

State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage>
  with SingleTickerProviderStateMixin {
  late TabController _tabController;
  final ValueNotifier<int> _selectedIndex = ValueNotifier<int>(0);

  @override
  void initState() {
    _tabController = TabController(
      length: tabBar.length,
      vsync: this,
      initialIndex: _selectedIndex.value,
    );
    _tabController.addListener(() {
      _selectedIndex.value = _tabController.index;
    });
    super.initState();
  }

  void _onItemTapped(int index) {
    _selectedIndex.value = index;
    _tabController.index = index;
  }

  Widget _buildBottomNavigationBar() {
    return ValueListenableBuilder<int>(
      valueListenable: _selectedIndex,
      builder: (context, value, child) {
        return BottomNavigationBar(
          items: tabBar
            .map(
              (e) => BottomNavigationBarItem(
                icon: e.icon,
                label: e.title,
              ),
            )
            .toList(),
          currentIndex: value,
          onTap: _onItemTapped,
        );
      },
    );
  }
}

```

```

Widget _buildBody() {
  return SafeArea(
    child: TabBarView(
      physics: const NeverScrollableScrollPhysics(),
      controller: _tabController,
      children: tabBar.map((e) => e.page).toList(),
    ),
  );
}

@override
Widget build(BuildContext context) {
  final auth = context.watch<AuthService>();
  final catalog = context.watch<CatalogModel>();
  Utils.log(auth.user!.uid);
  Utils.log(catalog.items.length.toString());
  return Scaffold(
    appBar: const HomeAppBar(),
    body: _buildBody(),
    bottomNavigationBar: _buildBottomNavigationBar(),
  );
}
}

```

storage_service.dart – код який включає клас StorageService, який надає методи для отримання та завантаження зображень зі сховища Firebase Storage.

```

import 'package:etrick/constants.dart';
import 'package:etrick/models/catalog_model.dart';
import 'package:etrick/services/utils.dart';
import 'package:firebase_storage/firebase_storage.dart';

```

```

enum PictureQuality { low, high }

class StorageService {
  static final storage = FirebaseStorage.instance;
  static Future<String> getPicture({
    required CatalogItem item,
    required int pictureId,

```

```

        String? color,
        PictureQuality quality = PictureQuality.high,
    )) {
        color ??= item.colors.first;
        String path =
            'catalog/${item.category}/${item.id}/${color}/${quality.name}/${pictureId  +
1}.jpg';
        Utils.log('[getPicture] $path');
        return storage.ref().child(path).getDownloadURL();
    }

```

```

static Future<int> getPicturesCount({
    required CatalogItem item,
    PictureQuality quality = PictureQuality.high,
    String? color,
}) async {
    color ??= item.colors.first;
    String path = 'catalog/${item.category}/${item.id}/${color}/${quality.name}';
    Utils.log('[getPicturesCount] $path');
    final Reference folderRef = storage.ref().child(path);
    final ListResult result = await folderRef.listAll();

    return result.items.length;
}

```

```

static downloadCatalogPictures(
    CatalogModel catalog,
    List<CatalogItem> items, [
    List<String>? categories,
]) async {

```

```

categories ??= Constants.categories.keys.toList();
List<CatalogItem> categoryItems =
    items.where((item) => categories!.contains(item.category)).toList();
for (var item in categoryItems) {
    List<Pictures> pictures = [];
    await Future.forEach(item.colors, (color) async {
        int idLength = await getPicturesCount(item: item, color: color);
        List<String> urls = [];
        for (int i = 0; i < idLength; i++) {
            urls.add(await getPicture(item: item, pictureId: i, color: color));
        }
        pictures = [
            ...pictures,
            Pictures(color: color, urls: urls),
        ];
    });
    catalog.saveCatalogItem(item, pictures);
}
}
}

```

Додаток В – Опис мобільного застосунку "eTrick" для продажу мобільних аксесуарів

1. Загальні відомості

1.1 Позначення і найменування програми

Назва програми: «eTrick». Мобільний застосунок "eTrick" для продажу мобільних аксесуарів. Застосунок надає змогу користувачам зручно і швидко замовляти мобільні аксесуари. Завдяки мобільному застосунку, користувачі мають повноцінний доступ до всього функціоналу мобільного застосунку навіть при відсутності мережі Інтернет. Мобільний застосунок "eTrick" надає зручний та інтуїтивно зрозумілий інтерфейс для користувачів будь-яких категорій.

1.2 Програмне забезпечення, необхідне для функціонування програми

Система повинна задовольняти вказаним вимогам на мобільному пристрої наступної мінімальної комплекції:

- Процесор серії Exynos 990 Qualcomm Snapdragon 865;
- Оперативна пам'ять 2 ГБ;
- Дискового простору 100 МБ для встановлення програми;
- Роздільна здатність екрану 1280x800 пікселів;

Інтернет-з'єднання: Застосунок вимагає доступу до Інтернету для завантаження інформації про продукти, оновлення даних та здійснення покупок. Доступ до Wi-Fi або мобільного Інтернету (3G, 4G, 5G) є необхідним.

Мобільний застосунок має розміщуватися виключно на мобільному телефоні з Android операційною системою.

1.3 Мови програмування

Мова програмування, використана в розробці програми "eTrick" це Dart. Код програми використовується з Flutter, який є фреймворком для створення красивих і високопродуктивних додатків для платформ, таких як Android, iOS, веб і інші.

2. Функціональне призначення

- Пошук і фільтрація продуктів.
- Детальні описи продуктів.
- Кошик і оформлення замовлення.
- Оповіщення і відстеження замовлень.

3. Опис логічної структури

Програма "eTrick" має наступну логічну структуру:

Мобільний застосунок:

- Модуль авторизації та аутентифікації: Дозволяє модераторам та адміністраторам авторизуватися в системі за допомогою свого електронного листа та пароля.

- Модуль керування мобільним застосунком: Дозволяє керувати усім мобільним застосунком

- Модуль керування зображеннями: Дозволяє керувати системою переднього завантаження зображень

База даних:

- Таблиця «Користувачі»: Зберігає дані про користувачів, зокрема їхні ідентифікатори, email, ім'я та список покупок.

- Таблиця «Товари»: Зберігає всю необхідну інформацію про товари.

- Таблиця «Зображення» : Зберігає зображення високого і низького розширення

3.2 Алгоритм програми

Алгоритм роботи мобільного застосунку:

- Користувач авторизується завдяки своїй пошті та пароллю.
- Користувач має доступ до будь-якого товару.
- Користувач має змогу оформити замовлення і отримувати сповіщення, щодо його статусу.

- Адміністратор авторизується завдяки своїй пошті та пароллю.

- Адміністратор має доступ до будь-якого товару.

- Адміністратор має змогу редагувати будь-яку інформацію щодо товару.

3.3 Використовувані методи

Програмне забезпечення "eTrick" розроблено мовою програмування Dart, яка є об'єктно-орієнтованою.

- Метод завантаження товарів: Реалізація функціональності для завантаження товарів з бази даних.

– Метод обробки даних: Реалізація функцій для обробки отриманих даних про товар. Цей метод включає додавання у корзину, перегляд інформації про товар.

- Метод взаємодії з базою даних: Реалізація функцій для обробки замовлень та редагування списку товарів

– Метод відображення інтерфейсу користувача: Реалізація функціональності для відображення інтерфейсу мобільного застосунку. Цей метод включає в себе відображення товарів, списків та профілю користувача.

4. Програмні засоби розробки мобільного застосунку

У розробці програмного забезпечення "eTrick" були використані наступні програмні засоби:

- Використання фреймворку Flutter разом із мовою програмування Dart для створення мобільного застосунку.

– Використання мови програмування Dart для розробки компонентів інтерфейсу користувача, обробки взаємодії з користувачем та зв'язку з базою даних.

- Використання Firebase Firestore для зберігання, редагування та отримання даних користувача мобільного застосунку та отримання товарів.

– Використання Firebase Authentication для автентифікації та авторизації користувачів мобільного застосунку.

- Використання Firebase Storage для зберігання та отримання зображень товарів.

- Використання Visual Studio Code як середовища розробки.

5. Виклик та завантаження

Виклик та завантаження програмного забезпечення «eTrick» відбувається запуском застосунку у ОС Android.

6. Вхідні дані

Дані можна вводити в систему за допомогою різних пристроїв введення, наприклад, таких як сенсорний екран мобільного пристрою. Введення даних може здійснюватися шляхом ручного введення або вибору зі списків, що надаються.

7. Вихідні дані

Дані можна виводити з системи за допомогою пристроїв виведення, наприклад, таких як екран мобільного пристрою. Це дозволяє користувачу переглядати інформацію, яка була оброблена системою.

Додаток Г – Інструкція користувача мобільного застосунку "eTrick" для продажу мобільних аксесуарів

Використовуючи цю інструкцію, користувач зможе правильно взаємодіяти з мобільним застосунком "eTrick" для продажу мобільних аксесуарів.

Інструкція користувача для мобільного застосунку "eTrick" для продажу мобільних аксесуарів:

1. Завантаження та встановлення:

- Відкрийте магазин програм на своєму мобільному пристрої.
- У полі пошуку введіть "eTrick".
- Знайдіть застосунок "eTrick" у списку результатів пошуку.
- Натисніть на кнопку "Встановити" або "Завантажити" для початку завантаження та встановлення застосунку.
- Після завершення встановлення натисніть на значок "eTrick" на головному екрані або у списку програм, щоб запустити застосунок (див. рис. Г.1).

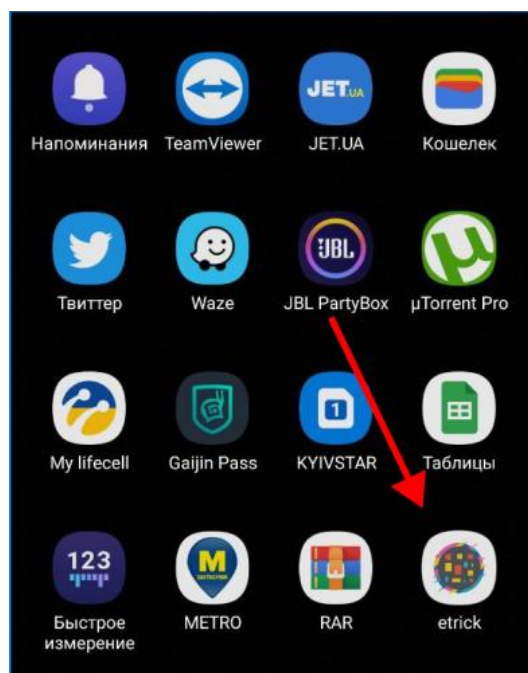


Рисунок Г.1 – Значок мобільного застосунку "eTrick" для продажу мобільних аксесуарів для телефонів

2. Реєстрація або вхід в обліковий запис:

- При першому запуску ви побачите екран реєстрації або входу в обліковий запис (див. рис Г.2).

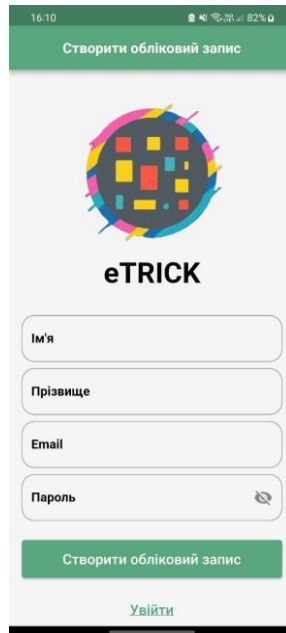


Рисунок Г.2 – Сторінка створення облікового запису

- Якщо у вас вже є обліковий запис, введіть свої дані (електронну пошту та пароль) і натисніть "Увійти"(див. рис Г.3).

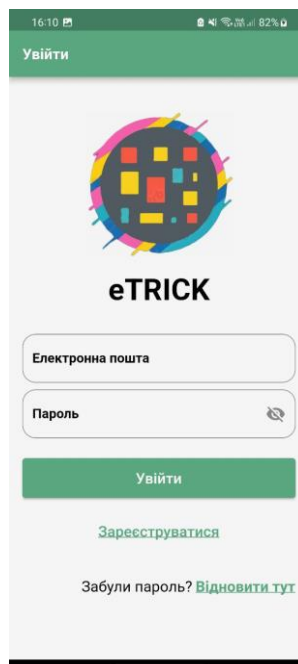


Рисунок Г.3 – Сторінка для входу в існуючий обліковий запис

– Якщо у вас немає облікового запису, натисніть на посилання "Зареєструватися" і слідуйте інструкціям для створення нового облікового запису.

3. Оформлення замовлення:

- Ви маєте змогу додати будь-який товар у кошик.
- Натисніть на рисунок кошика зверху справа.
- Натисніть кнопку «Оформити замовлення» (див. рис. Г.4).

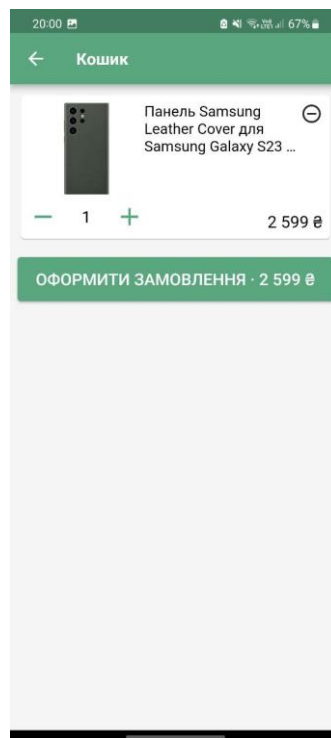


Рисунок Г.4 – Сторінка Кошику мобільного застосунку "eTrick" для продажу мобільних аксесуарів для телефонів

- Заповніть усі поля необхідними даними та підтвердіть замовлення.

Додаток Д – Програма та методика випробувань мобільного застосунку "eTrick" для продажу мобільних аксесуарів

1. Об'єкт випробування

Об'єктом випробування є мобільний застосунок "eTrick" для продажу аксесуарів для мобільних телефонів.

2. Мета випробування

Випробування має на меті виявлення можливих недоліків, помилок та проблем, які можуть вплинути на коректну роботу мобільного застосунку. Це процес, який допомагає перевірити функціональність, стабільність та надійність застосунку, забезпечуючи його оптимальну якість.

Конкретні цілі випробування включають:

- Перевірку коректності відображення усіх сторінок.
- Перевірку перегляду списку товарів.
- Перевірка функції пошук товару.
- Перевірку функції кошику.
- Перевірку оформлення замовлення.
- Перевірка функції попереднього завантаження зображень.
- Перевірку додавання, редагування та видалення товару адміністратором.

3. Вимоги до програмного забезпечення

При проведенні випробувань мобільного застосунку "eTrick" функціональні характеристики підлягають перевірці на відповідність вимогам, викладеним у пункті «Вимоги до функціональних характеристик» технічного завдання.

4. Вимоги до програмної документації

Програмна документація повинна включати в себе наступні документи:

- 1) Технічне завдання
- 2) Текст програми.
- 3) Опис програми.
- 4) Програма та методика випробувань.

5) Інструкція користувача.

5. Засоби і порядок випробувань

5.1 Технічні засоби, що використовуються під час випробувань

- CPU: Exynos 990
- RAM: 8 GB
- Дисковий простір: 128 GB

5.2 Програмні засоби, що використовуються під час випробувань

- Android 13
- One UI 5.1
- Flutter 3.10.4
- Dart SDK 3.0.3

5.3 Порядок проведення випробувань

Процес випробування мобільного застосунку "eTrick" включає два основних етапи: перевірку відповідності програмним вимогам та перевірку функціональних характеристик програмного забезпечення. Ці етапи дозволяють переконатись, що застосунок відповідає вимогам, виконує необхідні функції та працює належним чином перед його випуском на ринок.

6 Методи випробувань

6.1 Методика проведення перевірки вимог до програмної документації

Під час перевірки вимог до програмної документації було проведено оцінку наявності та відповідності необхідних документів. Виконано перевірку відповідності змісту і формату документів вимогам. Також було перевірено відповідність змісту документів вимогам, включаючи наявність необхідних даних і опис програми, а також відповідність програми і методики тестування вимогам до тестування. Оцінювалась структура та оформлення документів, включаючи заголовки, нумерацію сторінок, зміст і читабельність тексту. Перевірялась відповідність граматичним та орфографічним правилам. Важливо було також переконатись, що документи відповідають стандартам документації і мають зрозумілу структуру.

6.2 Методика проведення перевірки вимог до функціональних характеристик програмного продукту

Під час випробування мобільного застосунку "eTrick" для продажу мобільних аксесуарів були проведені наступні тести для перевірки вимог до функціональних характеристик:

- Перевірка функції пошуку продуктів.
- Перевірка правильності та детальності опису продуктів.
- Перевірка роботи кошику і оформлення замовлення.
- Перевірка роботи оповіщення і відстеження замовлень.

У таблиці Д.1 представлено перелік випробувань.

Таблиця Д.1 – Перелік випробувань основних функціональних можливостей

Дія	Очікуваний результат	Результат
Пошук товару	Відображення товару, що відповідають критеріям пошуку	Товар успішно відображається
Перегляд детальної інформації про аксесуар	Відображення повної інформації про обраний аксесуар	Опис товару досить детальний та успішно відображається
Додавання аксесуарів до кошика	Додавання аксесуару до списку покупок в кошику	Товар успішно додається до кошику
Перегляд замовлення в кошику	Відображення списку аксесуарів та загальної суми в кошику	Товар коректно відображається у кошику
Вибір способу доставки	Можливість обрати спосіб доставки	Вибір способу доставки коректно працює
Отримання сповіщення про стан замовлення	Отримання сповіщення з інформацією про стан замовлення	Сповіщення успішно відображаються
Редагування списку товарів	Зміна інформації про аксесуари та їх характеристики	Редагування товару коректно працює
Додавання або видалення фото товару	Завантаження або видалення фотографій для обраного аксесуару	Додавання товару коректно працює
Редагування опису товару	Зміна опису та характеристик обраного аксесуару	Редагування опису товару коректно працює

В результаті проведення випробувань мобільного застосунку "eTrick" для продажу аксесуарів для мобільних телефонів було перевірено його функціональні характеристики. Виходячи з результатів випробування, можна зробити висновок, що мобільний застосунок "eTrick" повністю відповідає функціональним характеристикам, та пройшов усі випробовування і дав очікуваний результат.