

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами

Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

—  проф. Приходько С.Б.

«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

на тему: Розробка вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Виконав: студент групи 4157ст

—  Воробець М. І.
(підпис, ПІБ)

Керівник роботи:

— доцент, к.ф.-м.н.
(посада, науковий ступень вчене звання)

—  Латанська Л. О.
(підпис, ПІБ)

Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

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

«ЗАТВЕРДЖУЮ»
 Гарант освітньої програми
 _____ доц. Макарова Л.М.
 (підпис)
 « 08 » ____ 04 ____ 2025 р.

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

Студенту _____ Воробцю Миколі Івановичу _____
 (Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Керівник роботи Латанська Л. О.

Затверджені наказом ректора №153 від 08.04.2025 р.

2. Термін подання роботи: 02.06.2025 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; Проєкт програмного забезпечення; Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, опис програми, програма та методика випробувань програмного забезпечення, інструкція користувача, текст програми).

5. Перелік презентаційних матеріалів _____

Титульний слайд; Мета та завдання кваліфікаційної роботи; Аналіз вимог до програмного забезпечення (Призначення вебзастосунку); Аналіз вимог до програмного забезпечення (Діаграма варіантів використання); Архітектура програмного забезпечення (Діаграма взаємодії шаблону проєктування та діаграма розгортання модельованої системи); Архітектура програмного забезпечення (Діаграма компонентів); Діаграми діяльності для варіантів використання; Концептуальна модель даних; Макети користувацького інтерфейсу; Діаграма класів; Діаграми послідовності; Логічна модель даних; Засоби розробки; Результати розробки; Висновки; Фінальний 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 08.04.2025 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	31.03.2025	ПП*
2.	Аналіз практичної задачі інженерії ПЗ	07.04.2025	ПП*
3.	Аналіз вимог до ПЗ	14.04.2025	ПП*
4.	Розробка архітектури ПЗ	21.04.2025	
5.	Розробка проекту ПЗ	05.05.2025	
6.	Реалізація та тестування ПЗ	12.05.2025	
7.	Підготовка розділу Результати розробки ПЗ	16.05.2025	
8.	Підготовка розділу з охорони праці	19.05.2025	ПП*
9.	Підготовка розділу ВИСНОВКИ	23.05.2025	
10.	Оформлення списку використаних джерел та додатків	26.05.2025	
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.)	02.06.2025	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
12.	Підготовка презентації та доповіді	06.06.2025	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	09.06.2025	
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	16.06.2025	

* - за результатами переддипломної практики (ПП), яка була з 03.02.2025 до 23.03.2025 р.

Студент


(підпис)

Воробець М. І.

(ПІБ)

Керівник роботи


(підпис)

Латанська Л. О.

(ПІБ)

АНОТАЦІЯ

Дана кваліфікаційна робота присвячена розв'язанню практичної задачі інженерії програмного забезпечення на тему «Розробка вебзастосунку для інтернет-магазину смартфонів «Smartafon»».

У роботі представлено аналіз практичної задачі інженерії програмного забезпечення, аналіз існуючих аналогів, обґрунтування необхідності власної розробки, постановку задачі, аналіз вимог, розробку архітектури, розробку проєкту, результати розробки програмного забезпечення, розділ з охорони праці, а також додатки.

Робота виконана на 111 сторінках, містить 22 рисунки, 33 таблиці, 5 додатків та список використаних джерел з 8 найменувань.

ABSTRACT

This document is a report of a qualification work, that aimed to solve a practical problem of software engineering on the topic «Development of a web application for the «Smartafon» online smartphone store».

The report presents an analysis of the practical task of software engineering, an analysis of existing analogs, justification for the necessity of one's own software development, task formulation, requirements analysis, architecture design, software project development, results of the software development, a section on occupational safety, and appendices.

The document contains 111 pages of typewritten text, which contains 22 figures, 33 tables, 5 appendices and a list of used sources of 8 titles.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБЗАСТОСУНКУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ СМАРТФОНІВ «SMARTAFON»	9
1.1 Аналіз практичної задачі інженерії програмного забезпечення.....	9
1.2 Аналіз існуючих аналогів	10
1.3 Постановка задачі на розробку вебзастосунку для інтернет-магазину смартфонів «Smartafon».....	14
2 Розробка вебзастосунку для інтернет-магазину смартфонів «Smartafon».....	18
2.1 Аналіз вимог до вебзастосунку для інтернет-магазину смартфонів «Smartafon»	18
2.1.1 Розробка моделі варіантів використання вебзастосунку для інтернет-магазину смартфонів «Smartafon»	18
2.1.2 Розробка специфікацій прецедентів вебзастосунку для інтернет-магазину смартфонів «Smartafon».....	21
2.2 Розробка архітектури вебзастосунку для інтернет-магазину смартфонів «Smartafon»	30
2.3 Проєкт вебзастосунку для інтернет-магазину смартфонів «Smartafon» ...	34
2.3.1 Ескізний проєкт вебзастосунку для інтернет-магазину смартфонів «Smartafon»	34
2.3.2 Технічний проєкт вебзастосунку для інтернет-магазину смартфонів «Smartafon»	41
2.3.3 Робочий проєкт вебзастосунку для інтернет-магазину смартфонів «Smartafon»	49
3 Результати розробки вебзастосунку для інтернет-магазину смартфонів «Smartafon»	69

4	ОХОРОНА ПРАЦІ.....	70
4.1	Основні законодавчі і нормативні документи, які регламентують охорону праці в Україні	70
4.2	Структура управління питаннями охорони праці на підприємстві	71
4.3	Забезпечення техніки безпеки та охорона праці для конкретних посад або при виконанні конкретних завдань. Розробка заходів по зменшенню дії шкідливих факторів для конкретних посад або при виконанні конкретних завдань.....	71
	ВИСНОВКИ.....	73
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	74
	ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ.....	75
	ДОДАТОК Б – ОПИС ВЕБЗАСТОСУНКУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ СМАРТФОНІВ «SMARTAFON»	82
	ДОДАТОК В – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ВЕБЗАСТОСУНКУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ СМАРТФОНІВ «SMARTAFON»	86
	ДОДАТОК Г – ІНСТРУКЦІЯ КОРИСТУВАЧА.....	89
	ДОДАТОК Д – ТЕКСТ ВЕБЗАСТОСУНКУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ СМАРТФОНІВ «SMARTAFON»	97

ВСТУП

Дана кваліфікаційна робота присвячена вирішенню практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробці вебзастосунку для інтернет-магазину смартфонів «Smartafon», який є ініціативною розробкою.

Комплексність та невизначеність умов полягають в тому, що вимоги можуть бути нечіткими, або недостатньо визначеними, в необхідності розробки зручного та інтуїтивно зрозумілого інтерфейсу для користувачів, в забезпеченні функціоналу для адміністраторів, а також в забезпеченні легкої масштабованості при зміні вимог до функціоналу та її продуктивності.

Оскільки ринок інтернет-магазинів смартфонів переповнений, перед початком розробки вебзастосунку ми маємо можливість зробити аналіз існуючих інтернет-магазинів для виявлення успішних рішень, недоліків і відмінностей в їх роботі.

В результаті виконання аналізу існуючих інтернет-магазинів [1-3], було виявлено їхні недоліки, такі як: дещо перевантажений дизайн сайту, візуальні проблеми з графічною частиною, відсутність повної інформації про товар та обмежену доступність цієї інформації. Ці недоліки негативно впливають на якість застосунку та комфорт його використання. Тому було вирішено розробити власний вебзастосунок для інтернет-магазину смартфонів «Smartafon», який не матиме цих недоліків.

Метою кваліфікаційної роботи є розв'язання практичної задачі інженерії програмного забезпечення, а саме розробка вебзастосунку для інтернет-магазину смартфонів «Smartafon», який забезпечить користувачам можливість швидко та легко отримувати всю необхідну інформацію про смартфони, переглядати відгуки, здійснювати замовлення та керувати ними, але не оплачувати їх, це буде дороблено на фінальному етапі створення бізнесу. Також вебзастосунок передбачає функціонал, який дозволить адміністраторам ефективно керувати каталогом

смартфонів, оновлювати інформацію, здійснювати модерування відгуків та контролювати процеси обробки замовлень. Особливу увагу буде приділено вирішенню функціональних, графічних та дизайнерських недоліків, які присутні в існуючих аналогах, що дозволить уникнути негативного впливу цих недоліків на якість застосунку та комфорт його використання.

Необхідно розв'язати такі завдання, щоб досягти поставленої мети:

- провести аналіз практичної задачі інженерії програмного забезпечення;
- провести аналіз існуючих аналогів та обґрунтувати необхідність розробки власного вебзастосунку;
- розробити постановку задачі та технічне завдання;
- виконати аналіз вимог;
- розробити архітектуру вебзастосунку;
- розробити вебзастосунок для інтернет-магазину смартфонів «Smartafon»;
- провести тестування та розробити документацію до розробленого вебзастосунку;
- розробити розділ охорони праці.

Об'єктом кваліфікаційної роботи є процес розробки вебзастосунку для інтернет-магазину смартфонів «Smartafon».

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБЗАСТОСУНКУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ СМАРТФОНІВ «SMARTAFON»

1.1 Аналіз практичної задачі інженерії програмного забезпечення

Спілкування – еволюційний процес, який, з часом, став можливим завдяки винаходам радіотелефону, провідного телефону, рації, а з розвитком часу - мобільних девайсів. В нашій сучасності без телефонів не уявляємо вже звичайного життя, але технології часто міняються, з'являються нові, і, відповідно, людина, через деякий час, вимушена купувати новий телефон. Інколи не вистачає часу, щоб самостійно приїхати у магазин, обрати певний телефон та купити його, на це треба час та ресурси на дорогу. Саме через це, для зручності, були створені інтернет-магазини, в тому числі й для продажу смартфонів.

Інтернет-магазин – місце, де відбувається продаж товарів або послуг споживачеві, використовуючи спеціальний вебзастосунок, на якому розміщується інформація про товар, виконується замовлення й оплата товару. Довгострокова інформація зазвичай зберігається в базі даних.

В інтернет-магазинах, що спеціалізуються на продажу смартфонів, представлені моделі з різними характеристиками (колір, процесор, кількість пам'яті, операційна система, різновид камери, виробник, ємність батареї, розміри екрана, кількість слотів для SIM-Карт, та їх стан).

Вебзастосунок для інтернет-магазину смартфонів «Smartafon» буде спеціалізованим застосунком, що призначений для спрощення та автоматизації процесів продажу смартфонів та який охоплюватиме основні функції, які необхідні для повноцінного функціонування інтернет-магазину, а саме: управління каталогом смартфонів (додавання, редагування, видалення смартфонів),

можливість додавати відгуки до смартфона, збереження смартфонів в кошику, створення замовлень, формування звітів.

Розробка вебзастосунку відбуватиметься з урахуванням додаткових вимог, що виходять за межі основної функціональності, а саме: валідація даних (перевірка правильності введених даних користувачем), інтуїтивно-зрозумілий та адаптивний інтерфейс, резервне копіювання бази даних та легка масштабованість при зміні вимог до функціоналу та продуктивності вебзастосунку.

Для розробки вебзастосунку для інтернет-магазину смартфонів «Smartafon» буде використані наступні засоби розробки:

- PHP: Мова програмування, яка була обрана для реалізації серверної частини вебзастосунку;
- Laravel: PHP-фреймворк для легкої та ефективної розробки серверної логіки, маршрутизації, валідації форм і організації MVC-архітектури;
- MySQL: Реляційна система керування базами даних для забезпечування збереження інформації про смартфони, користувачів, замовлення, статистики та іншої необхідної інформації;
- Bootstrap: Фреймворк для швидкої розробки адаптивного інтерфейсу, що забезпечує уніфіковане оформлення елементів інтерфейсу та гнучку сіткову систему для побудови макета сторінки.

Сервер вебзастосунку може бути запущений на виділеному сервері хостингу із встановленим сервером Apache чи Nginx, з підтримкою PHP та доступом до MySQL-сервера, який може бути запущений на тому ж самому сервері.

1.2 Аналіз існуючих аналогів

Перед початком розробки вебзастосунку проведемо аналіз існуючих аналогів, які надають можливість придбати смартфони. Метою аналізу є здатність визначитись з функціоналом власного вебзастосунку та виявити успішні рішення, недоліки і відмінності в роботі існуючих аналогів.

Було розглянуто інтернет-магазин «Світ смартфонів» [1]. Головну сторінку

представлено на рисунку 1.1.

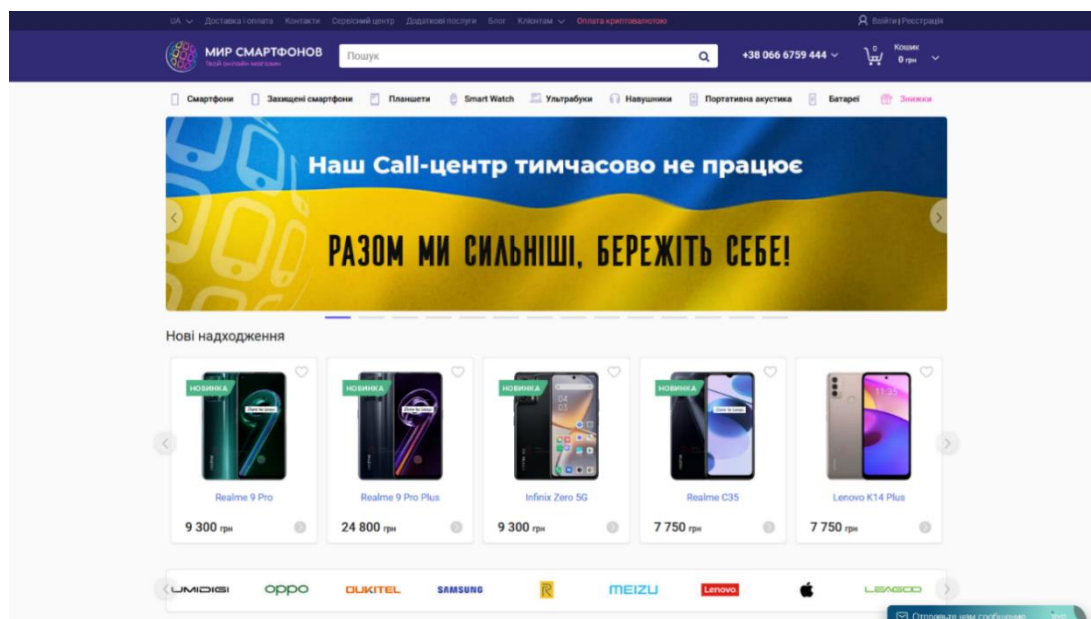


Рисунок 1.1 – Головна сторінка інтернет-магазину «Світ смартфонів»

Світ смартфонів – інтернет-магазин смартфонів з великим вибором телефонів, які були випущені з початку двохтисячних років до сьогодні та аксесуарами до цих телефонів.

Переваги:

- багато критеріїв для фільтрування товару;
- присутній опис товару;
- візуально-приємний інтерфейс;
- присутня можливість вибору додаткових послуг.

Недоліки:

- характеристика товару не повна;
- нефункціональне оформлення футера (Відсутній доступ до важливих сторінок інтернет-магазину).

Також було розглянуто інтернет-магазин «TOUCH» [2]. Головну сторінку представлено на рисунку 1.2.

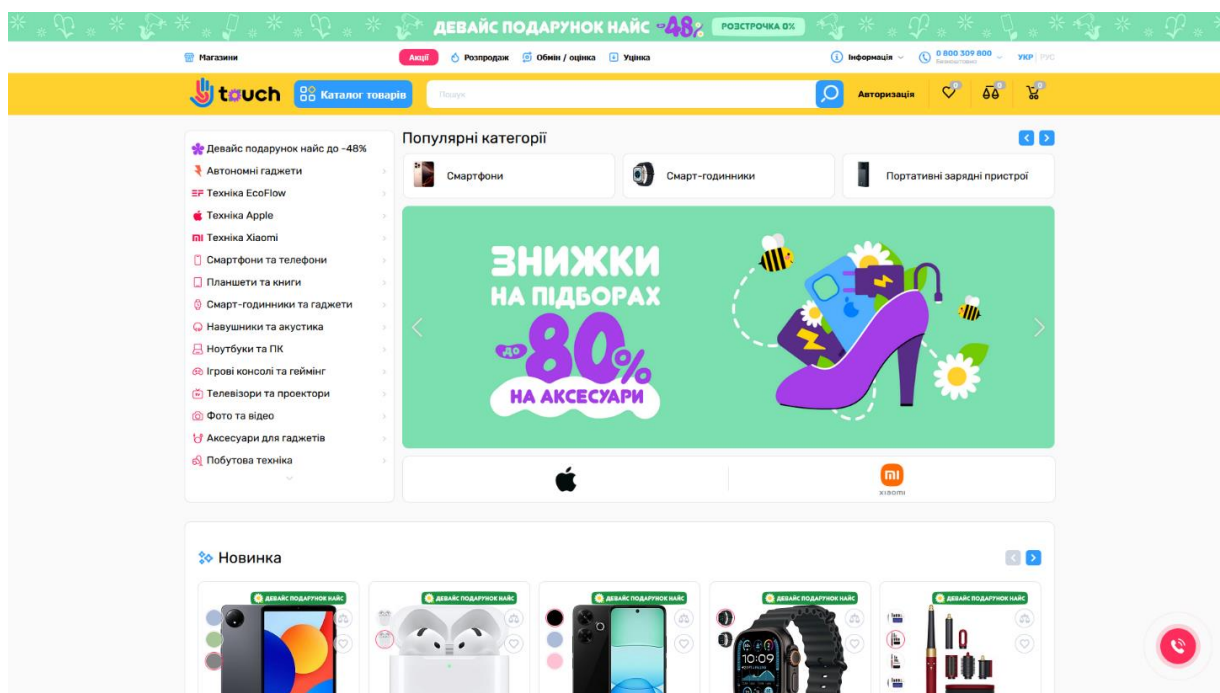


Рисунок 1.2 – Головна сторінка інтернет-магазину «TOUCH»

TOUCH – один із відомих інтернет-магазинів смартфонів та аксесуарів з широким асортиментом.

Переваги:

- багато критеріїв для фільтрування товару;
- присутній розширений опис товару;
- присутня можливість вибору варіантів товару;
- можливість брати техніку в кредит.

Недоліки:

- характеристика товару не повна;
- незрозуміло-підозрілі умови кредитування.

Останнім було розглянуто інтернет-магазин «Smartfon» [3]. Головну сторінку представлено на рисунку 1.3.

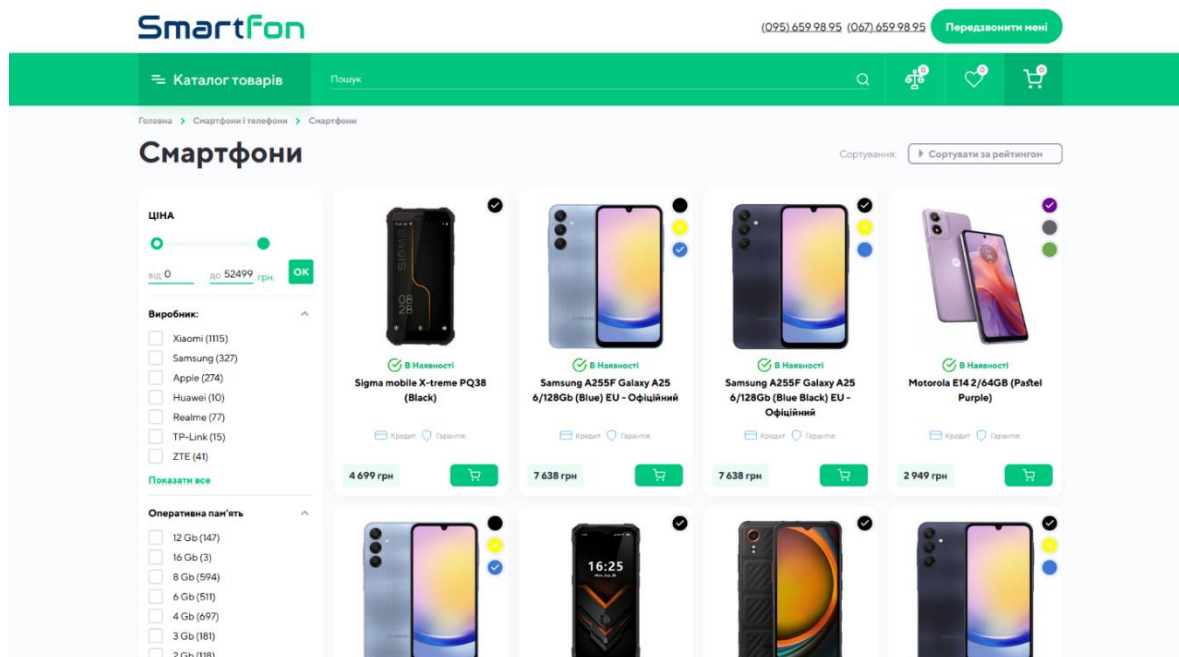


Рисунок 1.3 – Головна сторінка інтернет-магазину «Smartfon»

Smartfon – це, як зазначено на сайті, недорогий онлайн-маркет смартфонів та аксесуарів для них, від топових брендів.

Переваги:

- багато критеріїв для фільтрування товару;
- присутній розширений опис товару;
- присутня можливість вибору додаткових послуг;
- можливість брати техніку в кредит.

Недоліки:

- дещо переповнений дизайн сайту;
- часті візуальні проблеми з інтерфейсом.

Виконавши аналіз існуючих аналогів, було виявлено їхні недоліки, а саме: дещо перевантажений дизайн сайту, візуальні проблеми з графічною частиною, відсутність повної інформації про товар та обмежену доступність цієї інформації. Ці недоліки впливають на якість застосунку та комфорт його використання. Тому було поставлено за мету розробити власний вебзастосунок для інтернет-магазину смартфонів «Smartafon», який не матиме цих недоліків.

1.3 Постановка задачі на розробку вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Виходячи з аналізу предметної галузі та аналізу існуючих аналогічних готових рішень, прийняли рішення розробити власний вебзастосунок для інтернет-магазину смартфонів, який буде задовольняти наступним вимогам:

Вимоги до складу виконуваних функцій

Функції вебзастосунку для користувача:

- реєстрація (Вхідні дані: Прізвище та ім'я користувача; електронна пошта користувача; пароль користувача; номер телефону користувача. Результат: створення облікового запису користувача, який дозволяє доступ до персоналізованих функцій вебзастосунку та відображення повідомлення про створення облікового запису);
- авторизація (Вхідні дані: адреса електронної пошти користувача та пароль користувача; або тільки номер телефону користувача. Результат: доступ до особистого кабінету користувача з можливістю перегляду історії замовлень, редагування профілю тощо; відображення повідомлення про вхід в обліковий запис);
- відображення списку смартфонів (Вхідні дані: натиснення кнопки «Каталог». Результат: відображення списку смартфонів з неповною інформацією про смартфон, а саме назва смартфона, зображення смартфона, опис смартфона, ціна смартфона, оцінка смартфона, що дозволяє користувачеві ознайомитися з асортиментом);
- відображення інформації про смартфон (Вхідні дані: ID смартфона. Результат: відображення всієї інформації про смартфон, що дозволяє користувачеві ознайомитися з смартфоном);
- додавання відгуку (Вхідні дані: смартфон; текст відгуку; текст розділу «Переваги»; текст розділу «Недоліки»; рейтинг; дата та час створення відгуку; ID користувача. Результат: додавання відгуку до смартфона);

- відображення списку відгуків (Вхідні дані: ID смартфона. Результат: відображення списку відгуків, написаних іншими користувачами (ID відгуку; смартфон; текст розділу «Переваги»; текст розділу «Недоліки»; текст відгуку; користувач, який написав; дата написання));
- видалення відгуку (Вхідні дані: натиснення кнопки «Видалити»; ID відгуку. Результат: видалення відгуку до смартфона);
- створення замовлення смартфона (Вхідні дані: смартфон; кількість смартфонів; адреса доставки; вартість замовлення; дата та час створення замовлення; спосіб оплати; клієнт. Результат: формування замовлення з підтвердженням та деталями, яке обробляється програмним забезпеченням);
- відображення списку замовлень (Вхідні дані: натиснення кнопки «Мої замовлення». Результат: відображення списку замовлень з такими даними: ID замовлення; назва смартфона; кількість смартфонів; дата та час створення; спосіб доставки; адреса доставки; одержувач замовлення; вартість замовлення; спосіб оплати);
- скасування замовлень (Вхідні дані: ID замовлення. Результат: скасування замовлення, створеного користувачем).

Функції вебзастосунку для адміністратора:

- авторизація (Вхідні дані: адреса електронної пошти адміністратора; пароль адміністратора. Результат: доступ до адмін-панелі для управління вебзастосунком; відображення повідомлення про вхід в обліковий запис);
- додавання, редагування смартфона (Вхідні дані: натиснення посилання «Додати смартфон»/«Редагувати смартфон»; ID смартфона; артикул; код моделі; назва; колір; процесор; кількість оперативної пам'яті; кількість основної пам'яті; операційна система; камера; фронтальна камера; виробник; роздільна якість екрана; діагональ екрана; ємність батареї; ціна; акційна ціна; кількість слотів для SIM-карт; стан; зображення, тип слотів для SIM-карт. Результат: додавання або редагування даних смартфона ; відображення повідомлення про статус виконання дії);

- видалення смартфона (Вхідні дані: ID смартфона. Результат: видалення даних смартфона. Відображення повідомлення про статус виконання дії);
- видалення відгуку (Вхідні дані: натиснення кнопки «Видалити відгук»; ID відгуку. Результат: видалення відгуку для забезпечення якості та безпечності);
- запит на звіт продажів (Вхідні дані: натиснення кнопки «Звіт продажів за тиждень/місяць/квартал». Результат: звіт продажів (загальна кількість замовлень; кількість виконаних замовлень; кількість скасованих замовлень; виручка; дата початку звіту; дата кінця звіту));
- відображення списку замовлень (Вхідні дані: натиснення посилання «Замовлення». Результат: відображення списку замовлень з такими даними: ID замовлення; назва смартфонів; кількість смартфонів; дата та час створення; спосіб доставки; адреса доставки; одержувач замовлення; вартість замовлення; спосіб оплати);
- відображення списку відгуків (Вхідні дані: натиснення посилання «Відгуки»; ID смартфона. Результат: відображення списку відгуків написаних іншими користувачами (ID відгуку; смартфон; оцінка; текст розділу «Переваги»; текст розділу «Недоліки»; користувач, який написав; дата написання));
- відображення списку смартфонів (Вхідні дані: натиснення посилання «Смартфон». Результат: відображення списку смартфонів з інформацією про смартфон, а саме назва смартфона, зображення смартфона, характеристики смартфона, ціна смартфона, оцінка смартфона).

Вимоги до організації вхідних і вихідних даних

Вхідні дані – це інформація, що вводиться вручну (залежно від типу поля введення). Дані можуть бути: числовими (наприклад, ціни), текстовими (наприклад, назви моделей), або датами. Також можуть бути вибілковими списками (наприклад, список брендів смартфонів, тип операційної системи, тощо), які полегшують процес введення.

Вихідною інформацією є відображення результатів операцій у формі або повідомлення, а також сформовані звіти. Вихідними даними для звітів буде файл

Excel-таблиць, в якому буде міститись вся потрібна інформація.

Вимоги до складу й параметрів технічних засобів

Система повинна задовольняти вказаним мінімальним вимогам на сервері, а саме:

- процесор: 4 ядра 1.8 ГГц;
- оперативна пам'ять: 8 ГБ;
- пам'ять: 100 ГБ вільного місця на диску.

Користувачський комп'ютер повинен задовольняти таким мінімальним вимогам:

- процесор: 2 ядра 1.6 ГГц;
- оперативна пам'ять: 2 ГБ;
- доступ до інтернету.

Вимоги до інформаційної й програмної сумісності

Мінімальні вимоги інформаційної й програмної сумісності сервера:

- ОС: Windows 10 або Linux Ubuntu 20.10;
- PHP 8.2;
- phpMyAdmin 5.2;
- Система керування базами даних MySQL 8.0.

Мінімальні вимоги інформаційної й програмної сумісності комп'ютера користувача:

- ОС: Windows 7 або Linux Ubuntu 10;
- Браузер: Google Chrome 79, Mozilla Firefox 71, Microsoft Edge 79, Opera 65 або Safari 13.

Повна постановка задачі сформульована в технічному завданні, яке знаходиться в додатку А.

2 РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ СМАРТФОНІВ «SMARTAFON»

Для проєктування вебзастосунку використовується мова моделювання UML (Unified Modeling Language). UML – це уніфікована мова моделювання, яка використовується для проєктування та документування програмних систем, що допомагає описати структуру та поведінку програми за допомогою відповідних діаграм. Для побудови моделей було використано програмне забезпечення Rational Rose.

2.1 Аналіз вимог до вебзастосунку для інтернет-магазину смартфонів «Smartafon»

2.1.1 Розробка моделі варіантів використання вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Розробка моделі варіантів використання вебзастосунку для інтернет-магазину смартфонів «Smartafon» включає в себе такі етапи: виявлення акторів та їх опис; визначення та опис варіантів використання; побудову діаграми варіантів використання; розробка специфікацій варіантів використання.

Вебзастосунок для інтернет-магазину смартфонів «Smartafon» матиме два актори, а саме Адміністратор та Користувач. Список акторів та їх короткий опис представлено в таблиці 2.1.

Таблиця 2.1 – Виявлені актори

Актор	Короткий опис
Адміністратор	Користувач, що має повний доступ. Керує записами про смартфони, замовленнями, відгуками. Має доступ для формування звітів.
Користувач	Користувач, що має обмежений доступ. Має права переглядати існуючі смартфони, додавати відгуки, робити замовлення.

Визначені варіанти використання та їх опис представлено в таблиці 2.2

Таблиця 2.2 – Виявлені варіанти використання

Код	Актор	Назва	Формулювання
1	2	3	4
A1	Адміністратор	Авторизація	Варіант використання, що забезпечує адміністратора можливістю авторизуватись в свій обліковий запис.
A2	Адміністратор	Перегляд усіх замовлень	Варіант використання, що забезпечує адміністратора можливістю переглядати усі замовлення.
A3	Адміністратор	Робота з смартфонами	Варіант використання, що забезпечує адміністратора можливістю додати новий смартфон, змінювати інформацію вже існуючого смартфона та видалити існуючий смартфон.
A4	Адміністратор	Видалення відгуків	Варіант використання, що забезпечує адміністратора можливістю видаляти відгуки.
U1	Користувач	Авторизація	Варіант використання, що забезпечує користувача можливістю авторизуватись в свій обліковий запис.
U2	Користувач	Реєстрація	Варіант використання, що забезпечує користувача можливістю зареєструвати обліковий запис.
U3	Користувач	Перегляд існуючих смартфонів	Варіант використання, що забезпечує користувача можливістю переглядати існуючі смартфони у вебзастосунку.
U4	Користувач	Перегляд інформації про смартфон	Варіант використання, що забезпечує користувача можливістю переглядати інформацію про смартфон.
U5	Користувач	Створення замовлень	Варіант використання, що забезпечує користувача можливістю створити замовлення з потрібним йому смартфоном.
U6	Користувач	Перегляд замовлень	Варіант використання, що забезпечує користувача можливістю переглядати створені ним замовлення.

Кінець таблиці 2.2

1	2	3	4
U7	Користувач	Видалення замовлень	Варіант використання, що забезпечує користувача можливістю скасовувати замовлення, яке він створив.
U8	Користувач	Розміщення відгуків	Варіант використання, що забезпечує користувача можливістю розмістити відгук до смартфона.
U9	Користувач	Відбір смартфонів за характеристикою та ціною	Варіант використання, що забезпечує користувача можливістю використовувати фільтр і відбирати смартфони за характеристикою та ціною.
U10	Користувач	Обрання способу оплати	Варіант використання, що забезпечує користувача можливістю обрати спосіб оплати при створенні нового замовлення.
U11	Користувач	Редагування інформації про себе	Варіант використання, що забезпечує користувача можливістю редагувати інформацію про себе.
U12	Користувач	Пошук смартфонів у вебзастосунку	Варіант використання, що забезпечує користувача можливістю виконувати пошук смартфонів у вебзастосунку.
U13	Користувач	Видалення відгуків	Варіант використання, що забезпечує користувача можливістю видаляти відгуки.

Діаграма варіантів використання – діаграма, яка відображає відношення між акторами та їх варіантами використання в програмному забезпеченні, тобто хто користується та які дії він може виконувати.

Діаграму варіантів використання для інтернет-магазину смартфонів представлена на рисунку 2.1

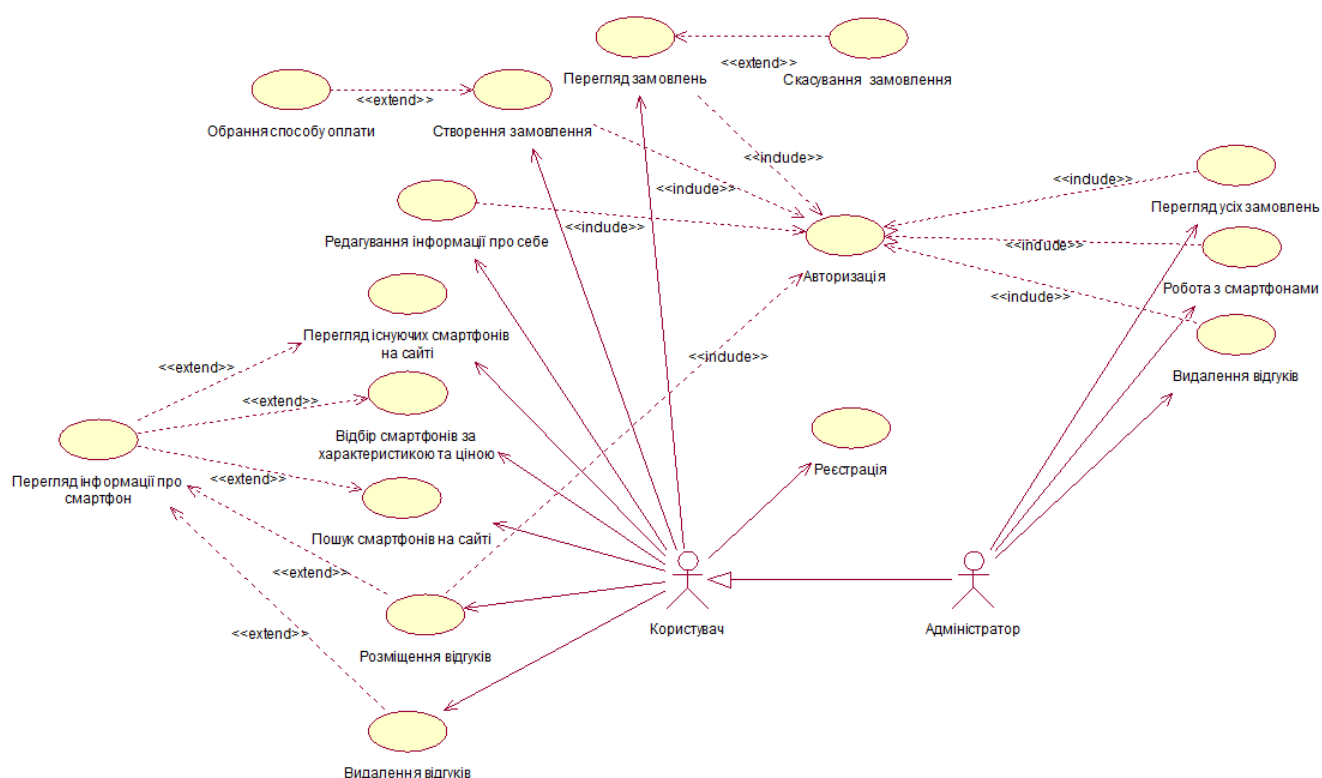


Рисунок 2.1 – Діаграма варіантів використання вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Представлена діаграма варіантів використання відображає визначені ролі та основні сценарії взаємодії користувачів із вебзастосунком, що дає змогу краще зрозуміти логіку його роботи та полегшує процес проєктування архітектури вебзастосунку.

2.1.2 Розробка специфікацій прецедентів вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Для конкретизації варіантів використання створимо їх специфікації. Специфікації варіантів використання представлені в таблицях 2.3-2.18

Таблиця 2.3 – Специфікація прецеденту «Авторизація»

Складова	Опис
Короткий опис	Варіант використання, що забезпечує користувача можливістю авторизуватись в свій обліковий запис.
Дійові особи прецеденту	Користувач, Адміністратор
Передумови	Не визначені
Основний потік подій	1. Користувач ініціює авторизацію. 2. Вебзастосунок надає форму авторизації. 3. Користувач вводить адресу електронної пошти/номер телефону та пароль, які вводив при реєстрації облікового запису в задану форму. 4. Вебзастосунок перевіряє вірність введення даних та допускає до облікового запису.
Альтернативний потік подій	1. Якщо користувач вводить незареєстровану адресу електронної пошти/номер телефону, то вебзастосунок відображає повідомлення про це та пропонує повторити введення або завершити варіант використання. 2. Якщо користувач вводить пароль, що не співпадає з паролем, що вводив при реєстрації облікового запису, то вебзастосунок відображає повідомлення про це та пропонує повторити введення або завершити варіант використання.
Постумови	При успішній авторизації, вебзастосунок авторизує користувача та відкриває головну сторінку.

Таблиця 2.4 – Специфікація прецеденту «Реєстрація»

Складова	Опис
1	2
Короткий опис	Варіант використання, що забезпечує користувача можливістю зареєструвати обліковий запис.
Дійові особи прецеденту	Користувач
Передумови	Не визначені
Основний потік подій	1. Користувач ініціює реєстрацію. 2. Вебзастосунок надає форму. 3. Користувач заповнює форму реєстрації. 4 Вебзастосунок перевіряє валідність введених даних і створює новий обліковий запис.

Кінець таблиці 2.4

1	2
Альтернативний потік подій	1. Якщо користувач вводить вже зареєстровану адресу електронної пошти/номер телефону, то вебзастосунок відображає повідомлення про це та пропонує повторити введення або завершити варіант використання. 2. Якщо користувач вводить адрес електронної пошти/номер телефону не правильного формату, то вебзастосунок відображає повідомлення про це та пропонує повторити введення або завершити варіант використання. 2. Якщо користувач вводить пароль, що має довжину, що менше за 8 символів, то вебзастосунок відображає повідомлення про це та пропонує повторити введення або завершити варіант використання
Постумови	При успішній реєстрації, вебзастосунок авторизує користувача в новостворений обліковий запис та відкриває головну сторінку.

Таблиця 2.5 – Специфікація прецеденту «Перегляд існуючих смартфонів»

Складова	Опис
Короткий опис	Варіант використання, що забезпечує користувача можливістю переглядати існуючі смартфони у вебзастосунку.
Дійові особи прецеденту	Користувач
Передумови	Не визначені
Основний потік подій	1. Користувач ініціює перегляд існуючих смартфонів у вебзастосунку. 2. Вебзастосунок виводить список існуючих смартфонів. 3. Користувач переглядає список існуючих смартфонів.
Альтернативний потік подій	Відсутній.
Постумови	Після успішного виконання прецеденту, користувач отримує можливість використовувати фільтр та пошук смартфонів

Таблиця 2.6 – Специфікація прецеденту «Відбір смартфонів за характеристикою та ціною»

Складова	Опис
1	2
Короткий опис	Варіант використання, що забезпечує користувача можливістю використовувати фільтр і відбирати смартфони за характеристикою та ціною.
Дійові особи прецеденту	Користувач
Передумови	Не визначені.

Кінець таблиці 2.6

1	2
Основний потік подій	1. Користувач ініціює відбір смартфонів за характеристикою та ціною. 2. Вебзастосунок надає форму для фільтрації смартфонів. 3. Користувач вибирає потрібні йому характеристики та ціну у формі. 4. Вебзастосунок надає список смартфонів за обраними користувачем характеристиками. 5. Користувач переглядає список смартфонів.
Альтернативний потік подій	1. Якщо смартфонів із вибраними характеристиками немає, вебзастосунок відображає повідомлення про це.
Постумови	Не визначені.

Таблиця 2.7 – Специфікація прецеденту «Перегляд інформації про смартфон»

Складова	Опис
Короткий опис	Варіант використання, що забезпечує користувача можливістю переглядати інформацію про смартфон.
Дійові особи прецеденту	Користувач
Передумови	Користувач повинен ініціювати перегляд існуючих смартфонів.
Основний потік подій	1. Користувач обирає смартфон. 2. Користувач ініціює перегляд інформації про смартфон. 3. Вебзастосунок надає сторінку з детальною інформацією про смартфон. 4. Користувач переглядає інформацію про смартфон.
Альтернативний потік подій	Відсутній.
Постумови	Не визначені.

Таблиця 2.8 – Специфікація прецеденту «Розміщення відгуків»

Складова	Опис
1	2
Короткий опис	Варіант використання, що забезпечує користувача можливістю розмістити відгук до смартфона.
Дійові особи прецеденту	Користувач
Передумови	Користувач повинен увійти у свій обліковий запис та ініціювати перегляд інформації про смартфон.

Кінець таблиці 2.8

1	2
Основний потік подій	1. Користувач обирає смартфон. 2. Користувач ініціює перегляд інформації про смартфон. 3. Користувач ініціює розміщування відгуків. 4. Вебзастосунок надає форму. 5. Користувач пише відгук у формі та надсилає її. 6. Вебзастосунок зберігає дані в базі даних та оновлює список відгуків до смартфона.
Альтернативний потік подій	1. Якщо користувач не заповнив обов'язкові поля та намагається відіслати форму, вебзастосунок відображає повідомлення про це.
Постумови	Не визначенні.

Таблиця 2.9 – Специфікація прецеденту «Редагування інформації про себе»

Складова	Опис
Короткий опис	Варіант використання, що забезпечує користувача можливістю редагувати інформацію про себе.
Дійові особи прецеденту	Користувач
Передумови	Користувач повинен увійти у свій обліковий запис.
Основний потік подій	1. Користувач ініціює редагування інформації про себе. 2. Вебзастосунок надає форму. 3. Користувач редагує потрібну йому інформацію у формі та надсилає її. 4. Вебзастосунок зберігає дані в базі даних.
Альтернативний потік подій	Відсутній.
Постумови	Не визначені.

Таблиця 2.10 – Специфікація прецеденту «Створення замовлення»

Складова	Опис
1	2
Короткий опис	Варіант використання, що забезпечує користувача можливістю створити замовлення з потрібним йому смартфоном.
Дійові особи прецеденту	Користувач
Передумови	Користувач повинен увійти у свій обліковий запис.

Кінець таблиці 2.10

1	2
Основний потік подій	1. Користувач ініціює вибір смартфона. 2. Вебзастосунок надає форму з детальним описом смартфона. 3. Користувач ініціює додавання смартфона до кошику. 4. Вебзастосунок надає повідомлення про додавання смартфона до кошику. 5. Користувач ініціює створення замовлення. 6. Вебзастосунок надає форму для створення замовлення. 7. Користувач вводить усю потрібну інформацію у надану форму та робить замовлення. 8. Вебзастосунок зберігає інформацію в базі даних і створює замовлення
Альтернативний потік подій	1. Якщо користувач не ввів потрібні дані для створення замовлення та намагається створити замовлення, вебзастосунок виводить повідомлення про це.
Постумови	Якщо створення замовлення було успішним, вебзастосунок відобразить повідомлення про вдале створення замовлення. Користувач отримає можливість переглянути створене ним замовлення в списку замовлень, що знаходиться на сторінці облікового запису.

Таблиця 2.11 – Специфікація прецеденту «Видалення замовлення»

Складова	Опис
Короткий опис	Варіант використання, що забезпечує користувача можливістю скасовувати замовлення, яке він створив.
Дійові особи прецеденту	Користувач
Передумови	Користувач повинен увійти у свій обліковий запис.
Основний потік подій	1. Користувач ініціює перегляд замовлень. 2. Вебзастосунок надає список замовлень користувача. 3. Користувач обирає активне замовлення та ініціює скасування замовлення. 4. Вебзастосунок надає форму з повідомленням про підтвердження скасування замовлення. 5. Користувач підтверджує скасування замовлення. 6. Вебзастосунок оновлює інформацію в базі даних і скасовує замовлення.
Альтернативний потік подій	Відсутній.
Постумови	Не визначені.

Таблиця 2.12 – Специфікація прецеденту «Обрання способу оплати»

Складова	Опис
Короткий опис	Варіант використання, що забезпечує користувача можливістю обрати спосіб оплати при створенні нового замовлення.
Дійові особи прецеденту	Користувач
Передумови	Користувач повинен почати створювати замовлення.
Основний потік подій	1. Користувач ініціює вибір способу оплати. 2. Вебзастосунок надає форму з списком способів оплати. 3. Користувач обирає спосіб оплати. 4. Вебзастосунок зберігає зміни.
Альтернативний потік подій	Відсутній.
Постумови	Не визначені.

Таблиця 2.13 – Специфікація прецеденту «Перегляд усіх замовлень»

Складова	Опис
Короткий опис	Варіант використання, що забезпечує адміністратора можливістю переглядати усі замовлення.
Дійові особи прецеденту	Адміністратор
Передумови	Адміністратор повинен увійти у свій обліковий запис.
Основний потік подій	1. Адміністратор ініціює перегляд усіх замовлень. 2. Вебзастосунок надає список з усіма замовленнями. 3. Адміністратор переглядає список замовлень.
Альтернативний потік подій	Відсутній.
Постумови	Не визначені.

Таблиця 2.14 – Специфікація прецеденту «Видалення відгуків»

Складова	Опис
1	2
Короткий опис	Варіант використання, що забезпечує адміністратора можливістю видаляти відгуки.
Дійові особи прецеденту	Адміністратор
Передумови	Адміністратор повинен увійти у свій обліковий запис.

Кінець таблиці 2.14

1	2
Основний потік подій	1. Адміністратор ініціює управління відгуками. 2. Вебзастосунок надає форму. 3. Адміністратор видаляє відгук. 4. Вебзастосунок надає повідомлення з підтвердженням видалення відгуку. 5. Адміністратор підтверджує видалення відгуку. 4. Вебзастосунок видаляє відгук та зберігає зміни в базі даних.
Альтернативний потік подій	1. Якщо адміністратор не підтверджує видалення, вебзастосунок не видаляє відгук.
Постумови	Не визначені.

Таблиця 2.15 – Специфікація прецеденту «Робота з смартфонами»

Складова	Опис
Короткий опис	Варіант використання, що забезпечує адміністратора можливістю додати новий смартфон, змінювати інформацію вже існуючого смартфона та видалити існуючий смартфон.
Дійові особи прецеденту	Адміністратор
Передумови	Адміністратор повинен увійти у свій обліковий запис.
Основний потік подій	Додавання нового смартфона 1 Адміністратор ініціює додавання нового смартфона. 2 Вебзастосунок надає форму. 3 Адміністратор вводить всю потрібну інформацію про смартфон у форму та відправляє її. 4 Вебзастосунок зберігає зміни в базі даних.
Альтернативний потік подій	Редагування смартфона: 1 Адміністратор ініціює редагування інформації про смартфон. 2 Вебзастосунок надає форму, яка заповнена даними про смартфон з бази даних. 3 Адміністратор редагує потрібну інформацію у формі. 4 Вебзастосунок зберігає зміни в базі даних. Видалення смартфона: 1 Адміністратор ініціює видалення смартфона. 2 Вебзастосунок надає форму з повідомленням про підтвердження видалення смартфона. 3 Адміністратор підтверджує видалення смартфона. 4 Вебзастосунок видаляє смартфон та зберігає зміни в базі даних.
Постумови	При успішному завершенні прецеденту, вебзастосунок відкриває адміністративну сторінку з списком смартфонів, що вже були додані.

Таблиця 2.16 – Специфікація прецеденту «Пошук смартфонів»

Складова	Опис
Короткий опис	Варіант використання, що забезпечує користувача можливістю виконувати пошук смартфонів у вебзастосунку.
Дійові особи прецеденту	Користувач
Передумови	Не визначені.
Основний потік подій	1. Користувач ініціює пошук смартфона. 2. Вебзастосунок активує форму пошуку. 3. Користувач вводить пошуковий запит в форму та надсилає. 4. Вебзастосунок надає список смартфонів за пошуком 5. Користувач переглядає наданий список смартфонів.
Альтернативний потік подій	1. Якщо смартфони, що відповідають пошуковому запиту, відсутні, вебзастосунок виведе повідомлення про це.
Постумови	Не визначені.

Таблиця 2.17 – Специфікація прецеденту «Перегляд замовлень»

Складова	Опис
Короткий опис	Варіант використання, що забезпечує користувача можливістю переглядати створені ним замовлення.
Дійові особи прецеденту	Користувач
Передумови	Користувач повинен увійти у свій обліковий запис.
Основний потік подій	1. Користувач ініціює перегляд замовлень. 2. Вебзастосунок надає форму зі списком всіх замовлень користувача. 3. Користувач переглядає список всіх своїх замовлень.
Альтернативний потік подій	Відсутній.
Постумови	Не визначені.

Таблиця 2.18 – Специфікація прецеденту «Видалення відгуків»

Складова	Опис
1	2
Короткий опис	Варіант використання, що забезпечує користувача можливістю видалити відгук до смартфона.
Дійові особи прецеденту	Користувач
Передумови	Користувач повинен увійти у свій обліковий запис та ініціювати перегляд інформації про смартфон.

Кінець таблиці 2.18

1	2
Основний потік подій	1. Користувач обирає смартфон. 2. Користувач ініціює перегляд інформації про смартфон. 3. Користувач ініціює видалення відгуку. 4. Вебзастосунок надає форму з повідомленням про підтвердження видалення відгуку. 5. Користувач підтверджує видалення. 6. Вебзастосунок оновлює інформацію в базі даних і видаляє відгук.
Альтернативний потік подій	1. Якщо користувач не підтверджує видалення, вебзастосунок не видаляє відгук.
Постумови	Не визначені.

У наведених таблицях подано та описано короткий зміст прецедентів, дійові особи прецедентів, передумови для виконання прецедентів, основний та альтернативні потоки прецедентів, а також постумови прецедентів.

2.2 Розробка архітектури вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Виходячи з вимог до вебзастосунку, які наведені в технічному завданні, було визначено, що доцільно обрати архітектурний стиль «Клієнт-Сервер» з шаблоном проектування «MVC (Model View Controller)».

Архітектура «Клієнт-Сервер» - це модель розподіленої обробки даних, у якій функції розділені між двома типами учасників: клієнтами та серверами. Вона є основою для багатьох сучасних систем, таких як веб-додатки, бази даних, мережеві сервіси тощо.

Завданням сервера є обробка запитів клієнтів, забезпечуючи доступ до даних та послуг. Він зберігає інформацію, контролює її цілісність і виконує основну частину обчислень.

Завдання клієнта полягає в ініціації запитів до сервера, щоб отримати необхідні дані або ресурси. Він працює за принципом запит-відповідь: надсилає запит, отримує результат і далі обробляє його відповідно до задачі.

MVC (Model View Controller) це шаблон проєктування, який використовується для розробки програмного забезпечення з чітким розподілом обов'язків між трьома компонентами:

- Model (Модель): Відповідає за бізнес-логіку та управління даними, отримує, зберігає, обробляє дані, а також взаємодіє з базами даних, проте не знає нічого про відображення чи введення даних;
- View (Подання/Вид/Представлення): Відповідає за відображення інформації користувачеві. Отримує дані від моделі і відображає їх у зручному вигляді і не містить логіки бізнес-процесів;
- Controller (Контролер): Відповідає за отримання вводу від користувача (наприклад, через кліки, форми чи команди). Обробляє ці дані, передає їх до моделі і визначає, яке подання має бути оновлене.

Діаграму взаємодії шаблону проєктування «MVC» представлено на рисунку 2.2

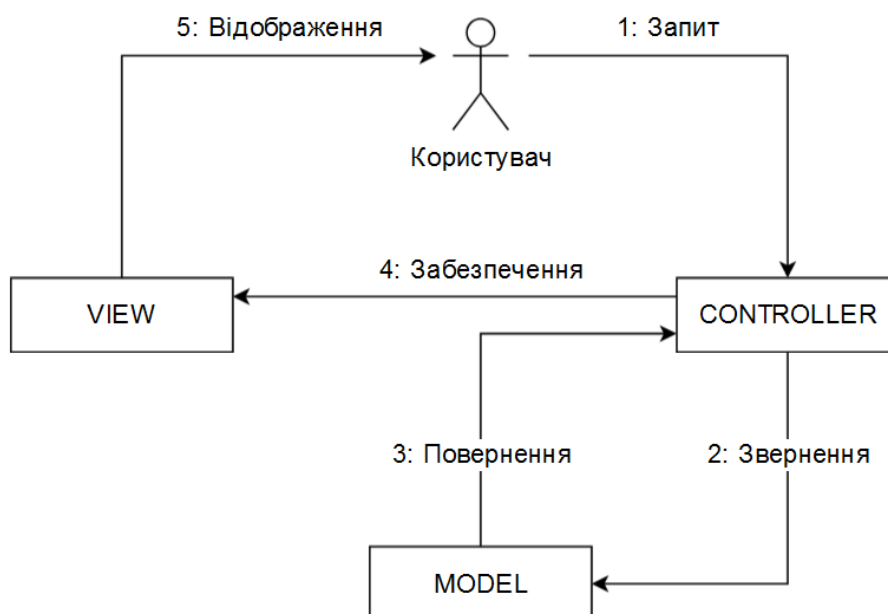


Рисунок 2.2 – Діаграма взаємодії шаблону проєктування «MVC (Model View Controller)»

Потік даних шаблону проєктування виглядає наступним чином:

- 1) Користувач взаємодіє з Контролером через інтерфейс;
- 2) Контролер отримує введені дані, обробляє їх та передає Моделі;
- 3) Модель виконує бізнес-логіку (Наприклад взаємодіє з базою даних) і оновлює свої дані.
- 4) Модель повідомляє Подання про зміну даних.
- 5) Подання отримує оновлені дані від Моделі та оновлює інтерфейс користувача.

Перевагами шаблону MVC є:

- Чіткий поділ обов'язків: Кожен компонент виконує свою окрему роль, що спрощує розуміння та підтримку коду.
- Можливість повторного використання: Модель та Подання не залежать одне від одного, тому Подання можна змінювати або використовувати повторно, не змінюючи Модель, або можна використовувати різні подання для одного й того ж контролера та моделі;
- Масштабованість: Чітка структура дозволяє легко додавати новий функціонал, не порушуючи існуючий код.

MVC використовують у проєктах, де важливо розділити логіку програми, представлення та управління даними для спрощення розробки та підтримки. Це актуально для великих і середніх програм, таких як веб-додатки, мобільні або настільні системи, де модульність та масштабованість є критично важливими. Завдяки MVC зручно додавати нові функції, тестувати окремі компоненти і адаптувати програму до змін без глобального переписування коду

Діаграма компонентів вебзастосунку для інтернет-магазину смартфонів «Smartafon» з використанням архітектурного шаблону MVC представлена на рисунку 2.3.

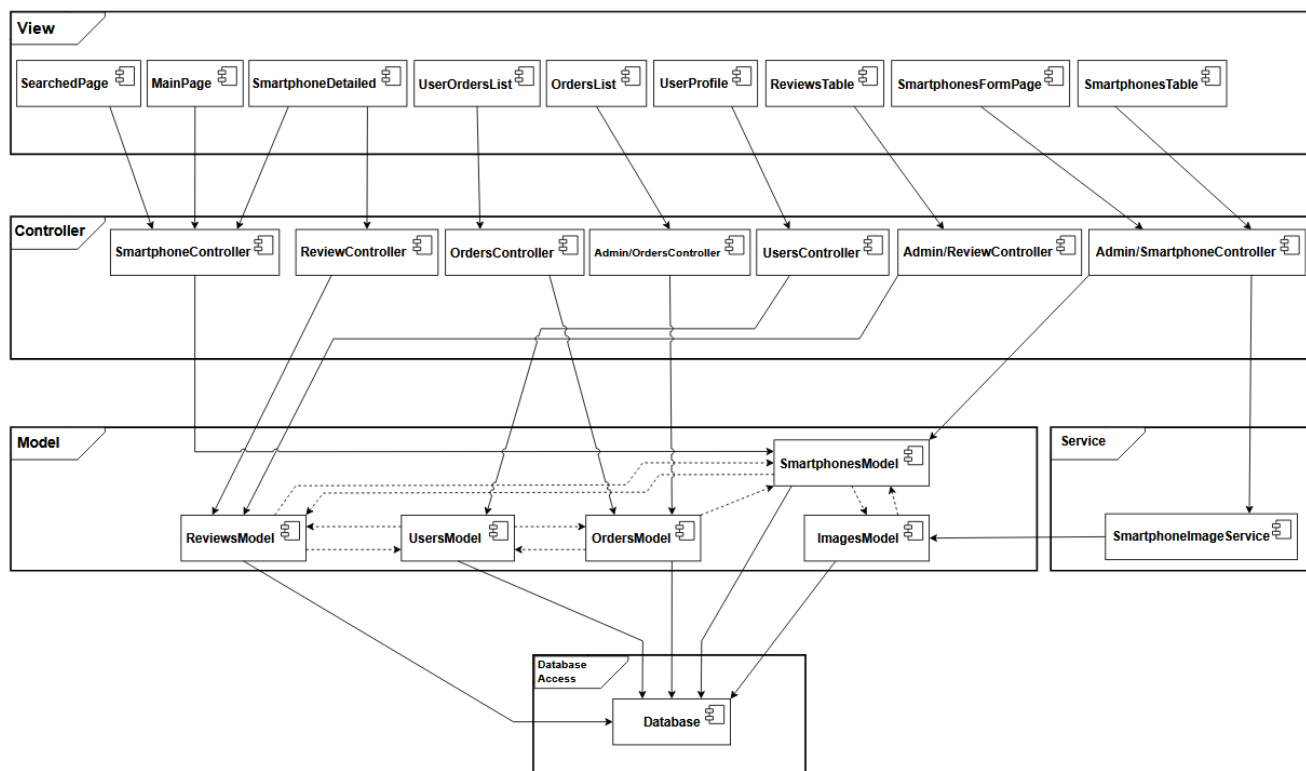


Рисунок 2.3 – Діаграма компонентів вебзастосунку для інтернет-магазину смартфонів «Smartafon» з використанням архітектурного шаблону MVC

Ця діаграма відображає основні компоненти програмного забезпечення та їх взаємодії, що дає чітке уявлення про структуру застосунку, де кожен компонент має свою чітко визначену роль, і як вони взаємодіють між собою.

Діаграму розгортання модельованої системи вебзастосунку для інтернет-магазину «Smartafon» представлено на рисунку 2.4

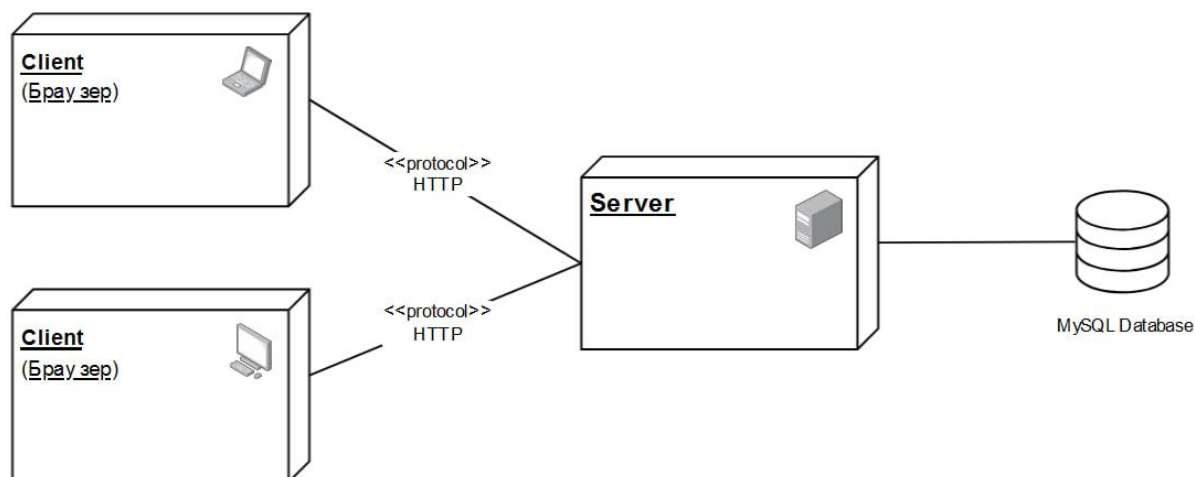


Рисунок 2.4 – Діаграма розгортання модельованої системи вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Діаграма розгортання – це діаграма, що відображає, як програмна система розміщується на пристроях. На ній відображаються вузли (сервери, клієнти тощо), компоненти програми, що працюють на цих вузлах, а також зв'язки між ними під час виконання системи.

2.3 Проєкт вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Проєктування вебзастосунку для інтернет-магазину смартфонів «Smartafon» здійснювалося поетапно та охоплювало ескізне, технічне та робоче проєктування.

2.3.1 Ескізний проєкт вебзастосунку для інтернет-магазину смартфонів «Smartafon»

З метою більш детального розуміння варіантів використання, наведених у таблицях 2.3–2.18, для деяких з них було побудовано діаграми діяльності, розроблено проєкт користувацького інтерфейсу, а також створено концептуальні моделі даних і програмної системи.

Розробка діаграми діяльності для певних варіантів використання вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Діаграма діяльності дозволяє моделювати послідовності бізнес-процесів або дій, реалізованих методами класів. Зазначені послідовності можуть являти собою альтернативні галузей процесу обробки даних або галузям, які можуть виконуватися паралельно. Діаграми діяльності є аналогом блок-схеми будь-якого алгоритму.

Діаграми діяльності для варіантів використання «Розміщення відгуків» та «Редагування інформації про себе» представлено на рисунках 2.5 та 2.6.

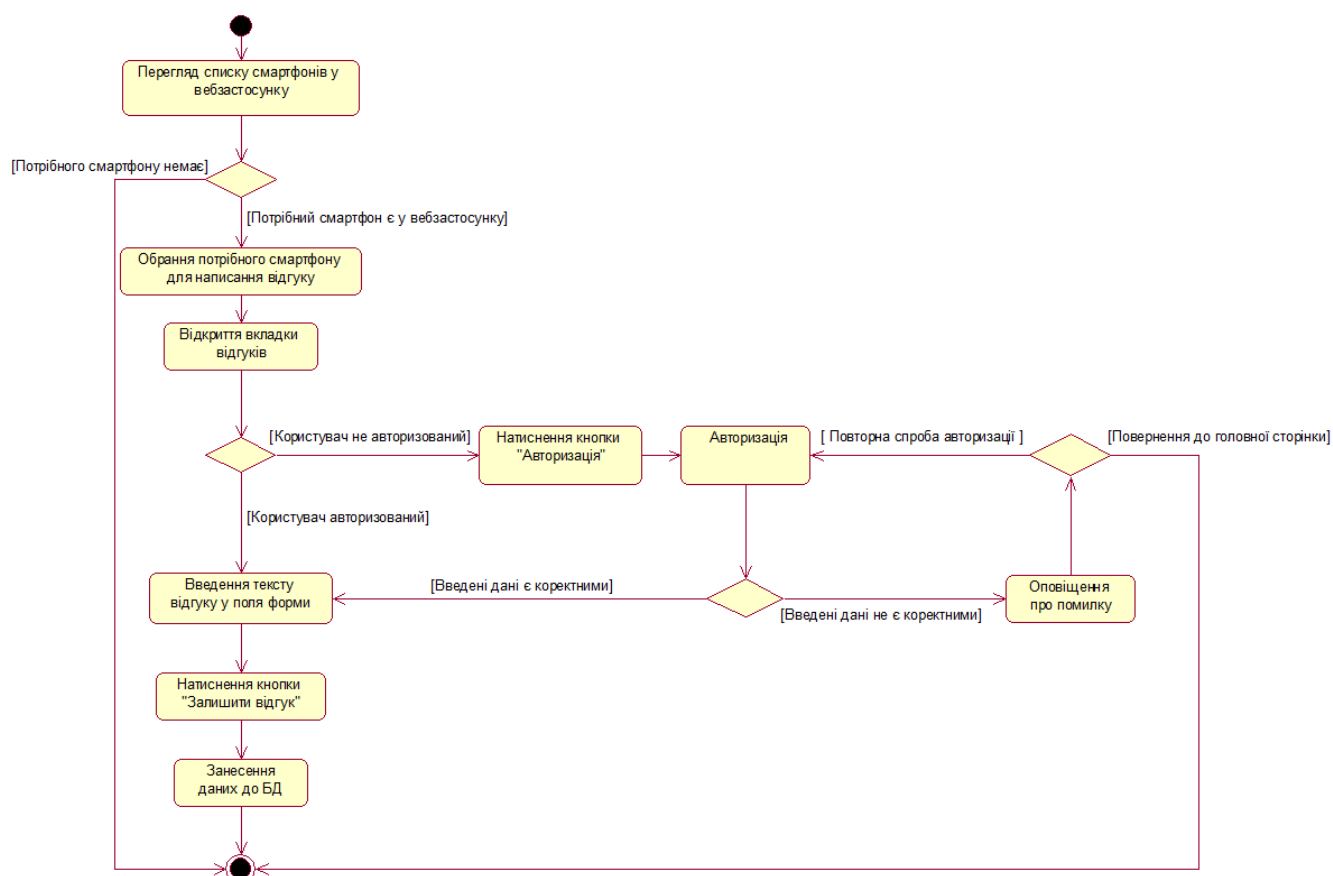


Рисунок 2.5 – Діаграма діяльності для прецеденту «Розміщення відгуків»

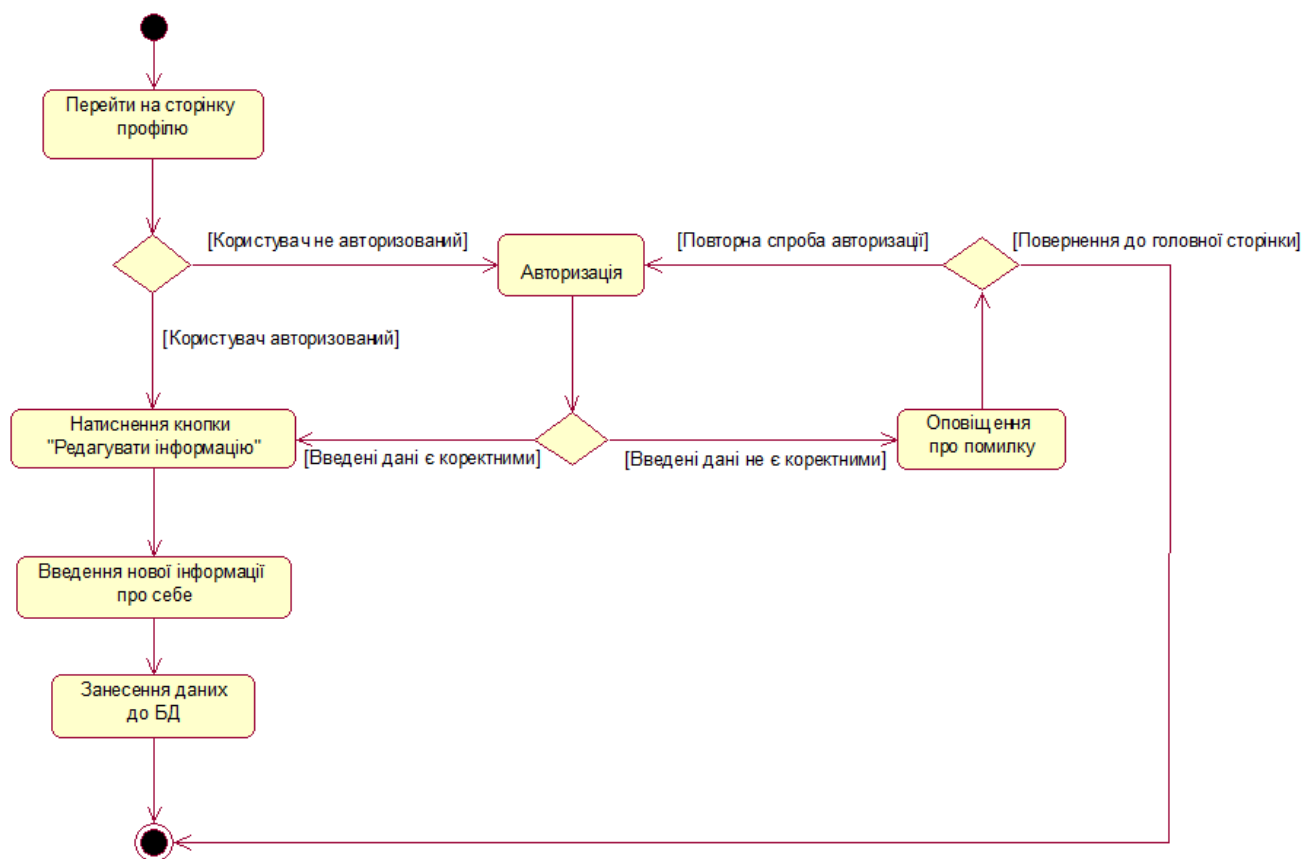


Рисунок 2.6 – Діаграма діяльності для прецеденту «Редагування інформації про себе»

Розробка концептуальної моделі даних вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Концептуальна модель даних описує які сутності існують у системі, їх атрибути та як вони пов'язані між собою без залежності від будь-якого рішення та без заглиблення у технічні аспекти реалізації.

Концептуальна модель даних розроблена засобами ER діаграми за нотацією Чена.

ER-модель – це інструмент для опису концептуальних схем за допомогою узагальнених блоків. Ця модель дозволяє систематично описувати та визначати певну предметну область.

Сутності та їх атрибути представлено в таблиці 2.19.

Таблиця 2.19 – Сутності та їх атрибути

Сутність	Атрибути
Відгуки	Код; Текст; Переваги; Недоліки; Рейтинг; Дата та Час створення; Користувач; Смартфон;
Користувач	Код; Прізвище; Ім'я; Пошта; Пароль; Телефон;
Замовлення	Код; Кількість смартфонів; Адреса доставки; Дата та час створення; Вартість замовлення; Статус замовлення; Користувач; Спосіб оплати;
Спосіб оплати	Код; Назва способу оплати;
Смартфон	Код; Артикул; Назва; Колір; Процесор; Кількість оперативної пам'яті; Кількість основної пам'яті; операційна система; Камера; Бренд; Кількість слотів для SIM-карт; Ціна; Ємність батареї; Діагональ екрана; Роздільна якість екрана; Стан; Слаг; Код моделі; Тип слотів для SIM-Карт; Акційна ціна; Кількість;
Зображення	Код; Назва файлу; Шлях до файлу; Смартфон;
Смартфон_Замовлення	Код; Смартфон; Замовлення;

Зв'язки між сутностями представлено в таблиці 2.20

Таблиця 2.20 – Сутності та їх зв'язки

Сутності	Зв'язок
Користувач – Відгуки	Один до багатьох
Користувач – Замовлення	Один до багатьох
Замовлення – Спосіб оплати	Один до багатьох
Смартфон – Відгуки	Один до багатьох
Смартфон – Зображення	Один до багатьох
Смартфон – Смартфон_Замовлення	Один до багатьох
Замовлення - Смартфон_Замовлення	Один до багатьох

ER діаграму вебзастосунку представлено на рисунку 2.7

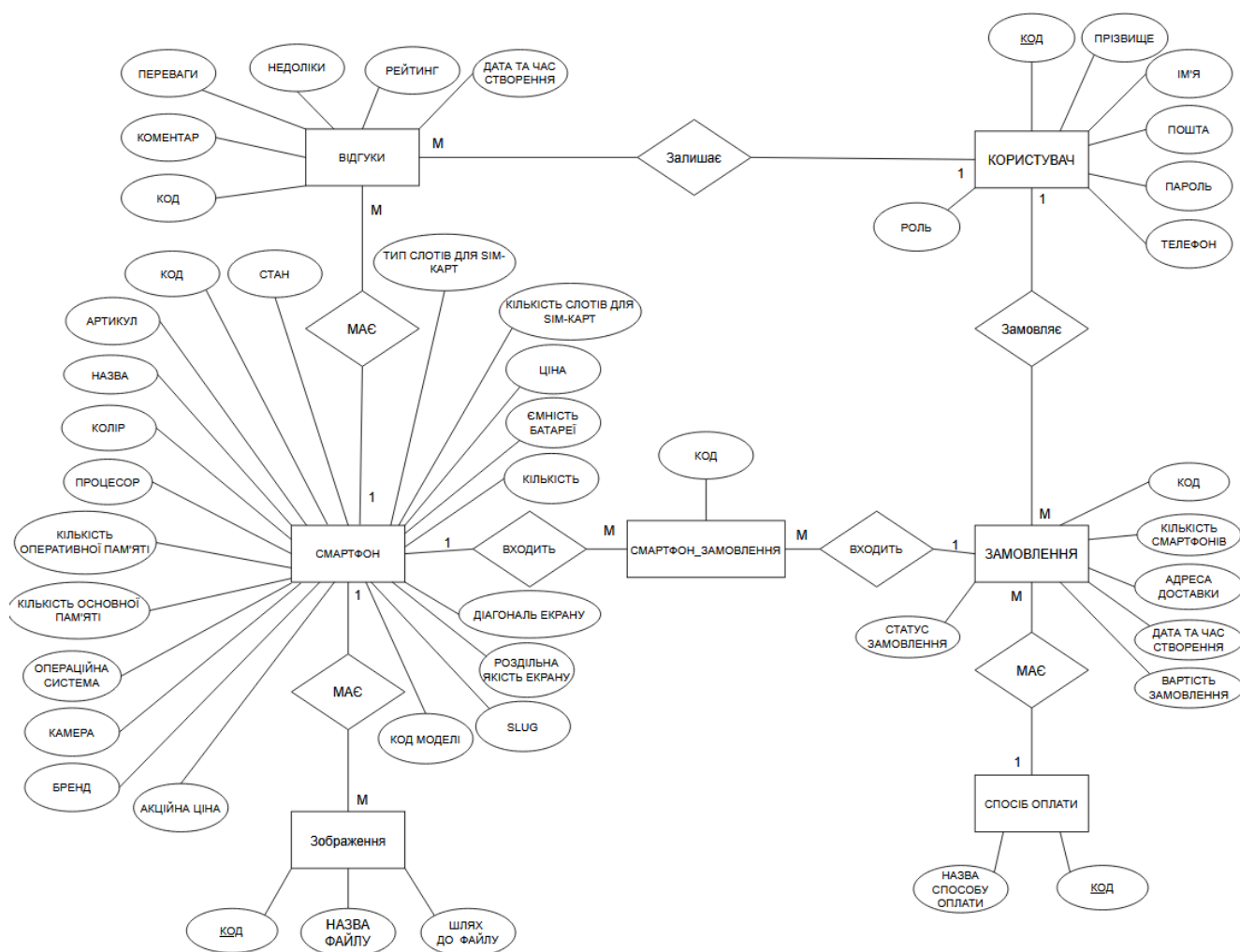


Рисунок 2.7 – Концептуальна модель даних вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Розробка користувацького інтерфейсу вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Інтерфейс користувача – це система, яка дозволяє взаємодіяти користувачеві з електронними пристроями чи програмними додатками. Основна мета інтерфейсу користувача – забезпечити ефективну взаємодію між користувачем і системою, дозволяючи йому виконувати різноманітні завдання та отримувати доступ до функціоналу програми чи пристрою.

Макети сторінок та форм вебзастосунку для інтернет-магазину смартфонів «Smartafon» представлені на рисунках 2.8-2.11.

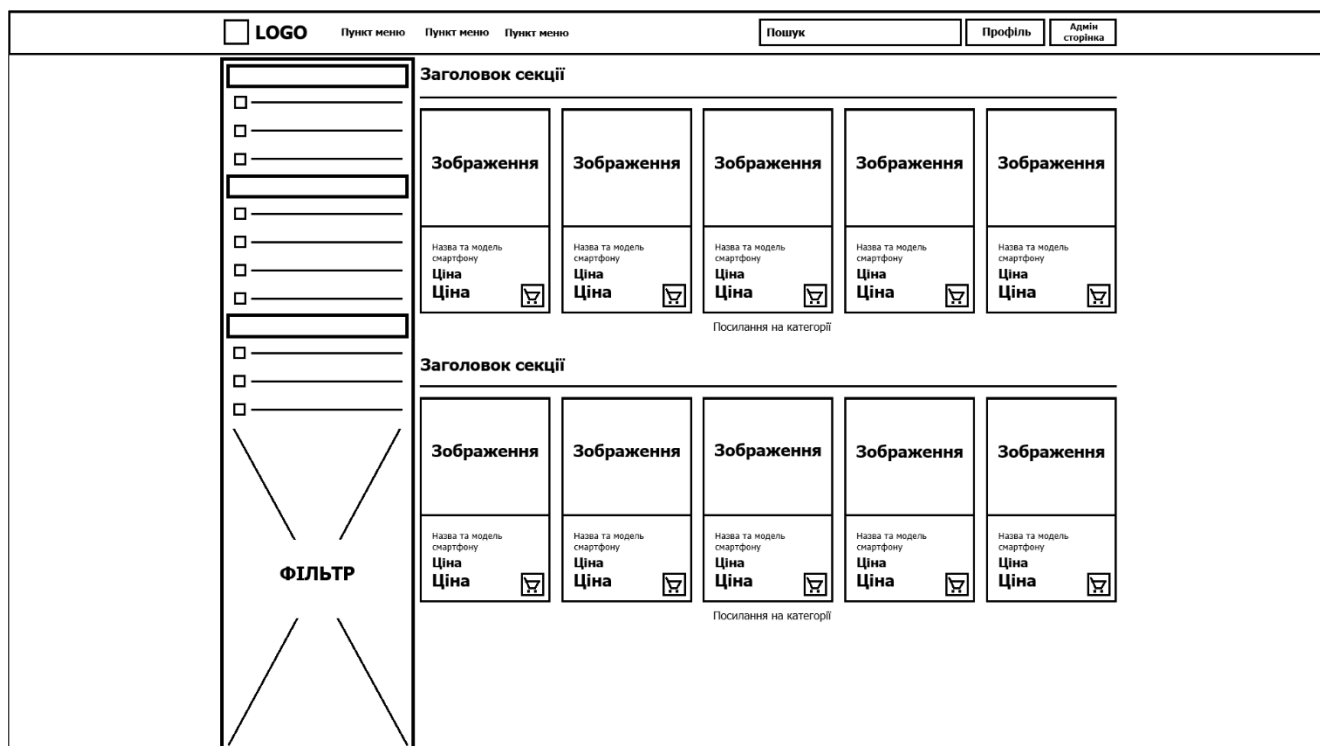


Рисунок 2.8 – Макет головної сторінки

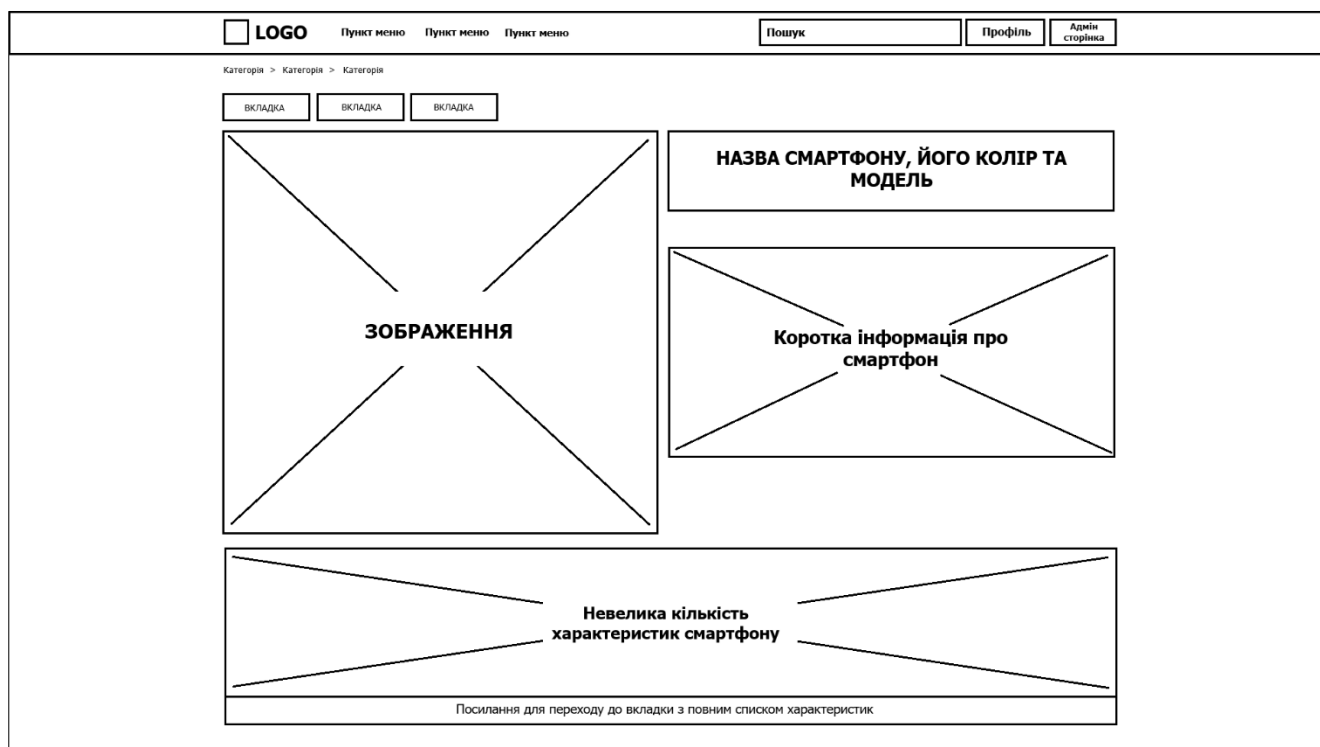


Рисунок 2.9 –Макет сторінки з детальною інформацією про смартфон

LOGO Пункт меню Пункт меню Пункт меню Пошук Профіль Адмін сторінка

Посилання для переходу до вкладки з повним списком характеристик

Секція з статистикою відгуків про товар

ІМ'Я КОРИСТУВАЧА ДАТА НАПИСАННЯ

Оцінка

Текст коментаря від користувача

Переваги
Текст переваг, що вказав користувач

Недоліки
Текст недоліків, що вказав користувач

ІМ'Я КОРИСТУВАЧА ДАТА НАПИСАННЯ

Оцінка

Текст коментаря від користувача

Переваги
Текст переваг, що вказав користувач

Недоліки
Текст недоліків, що вказав користувач

ІМ'Я КОРИСТУВАЧА ДАТА НАПИСАННЯ

Оцінка

Текст коментаря від користувача

Переваги
Текст переваг, що вказав користувач

Недоліки
Текст недоліків, що вказав користувач

Посилання для переходу до вкладки з повним списком відгуків та формою для написання власного відгуку

Рисунок 2.10 – Макет секції з відгуками на сторінці з детальною інформацією про смартфон

LOGO Пункт меню Пункт меню Пункт меню Пошук Профіль Адмін сторінка

Реєстрація

Поле для введення прізвища
Місце для виведення помилок

Поле для введення ім'я
Місце для виведення помилок

Поле для введення номеру телефона
Місце для виведення помилок

Поле для введення електронної пошти
Місце для виведення помилок

Поле для введення паролю
Місце для виведення помилок

Поле для повторного введення паролю
Місце для виведення помилок

Кнопка реєстрації

Авторизуватись

Рисунок 2.11 – Макет сторінки реєстрації

Представлені макети користувацького інтерфейсу візуалізують структуру

майбутнього вебзастосунку інтернет-магазину смартфонів, демонструючи розташування основних елементів інтерфейсу, що дозволяє оцінити зручність користування та логіку взаємодії з функціоналом вебзастосунку.

2.3.2 Технічний проєкт вебзастосунку для інтернет-магазину смартфонів «Smartafon»

На основі ескізного проєкту вебзастосунку для інтернет-магазину смартфонів «Smartafon» було розроблено технічний проєкт, в межах якого побудовано статичну та динамічну моделі, що представлені у вигляді діаграми класів та діаграм послідовностей відповідно.

Побудова статичної моделі вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Статична модель представлена у вигляді діаграми класів.

Діаграма класів – це статична модель, яка демонструє класи системи, їхні властивості (атрибути), операції (методи), а також зв'язки між класами, що характеризують структуру програмної системи.

Діаграму класів вебзастосунку для інтернет-магазину смартфонів «Smartafon» представлено на рисунку 2.12.

Специфікації класів вебзастосунку для інтернет-магазину смартфонів «Smartafon» представлені в таблицях 2.21-2.33

Характеристика	Опис
Відповідальність	Сервіс для роботи з зображеннями для смартфонів: додавання, оновлення, видалення
Атрибути	Відсутні
Методи	addImages() – Зберігання зображення на диску та їх дані в таблиці; updateImages() – Оновлення даних про зображення в таблиці, зберігання нових зображень на диску та видалення старих. deleteImages() – Видалення зображення на диску та їх дані в таблиці.

Таблиця 2.22 – Специфікація класу ImagesModel

Характеристика	Опис
Відповідальність	Модель зображень смартфонів
Атрибути	fillable – Список атрибутів бази даних, які дозволено передавати при масовому заповненні;
Методи	smartphone() – зв'язок з моделлю SmartphonesModel

Таблиця 2.23 – Специфікація класу Admin/SmartphoneController

Характеристика	Опис
Відповідальність	Контролер для адміністрування смартфонів у панелі адміністратора.
Атрибути	SmartphoneImageService \$imageService – Сервіс для роботи з зображеннями для смартфонів;
Методи	index() – Відображення сторінки з списком всіх смартфонів; createPage() – Відображення сторінки з формою для додавання нового смартфона; store() – Збереження даних нового смартфона; editPage() – Відображення сторінки з формою для редагування вже існуючого смартфона; update() – Оновлення даних про смартфон; destroy() – Видалення смартфона.

Таблиця 2.24 – Специфікація класу SmartphoneController

Характеристика	Опис
Відповідальність	Контролер для публічної частини відображення смартфонів користувачу.
Атрибути	Відсутні
Методи	index() – Відображення сторінки смартфона, де наявна детальна інформація, зображення смартфона та відгуки; homePage() – Відображення головної сторінки з невеликими списками смартфонів; goodsPage() – Відображення сторінки з повним списком смартфонів; getFilterItems() – Отримання категорій фільтру; applySorting() – Застосування сортування; applyFilters() – Застосування фільтрів.

Таблиця 2.25 – Специфікація класу SmartphonesModel

Характеристика	Опис
Відповідальність	Модель смартфонів, що представляє таблицю smartphones бази даних.
Атрибути	fillable – Список атрибутів бази даних, які дозволено передавати при масовому заповненні.
Методи	reviews() – Зв'язок з моделлю ReviewsModel; images() – Зв'язок з моделлю ImagesModel

Таблиця 2.26 – Специфікація класу Admin/ReviewController

Характеристика	Опис
Відповідальність	Контролер для адміністрування відгуків до смартфонів у панелі адміністратора.
Атрибути	Відсутні
Методи	index() – Відображення сторінки з списком всіх відгуків; destroy() – Видалення відгуку.

Таблиця 2.27 – Специфікація класу ReviewController

Характеристика	Опис
Відповідальність	Контролер для обробки створення, оновлення та видалення відгуків.
Атрибути	Відсутні
Методи	store() – Зберігання відгуку користувача; destroy() – Видалення відгуку користувача.

Таблиця 2.28 – Специфікація класу ReviewsModel

Характеристика	Опис
Відповідальність	Модель відгуків, що представляє таблицю reviews бази даних.
Атрибути	fillable – Список атрибутів бази даних, які дозволено передавати при масовому заповненні.
Методи	author() – Зв'язок з моделлю UsersModels; smartphones() – Зв'язок з моделлю SmartphonesModel.

Таблиця 2.29 – Специфікація класу Admin/OrdersController

Характеристика	Опис
Відповідальність	Контролер для адміністрування замовлень у панелі адміністратора.
Атрибути	Відсутні
Методи	index() – Відображення сторінки з списком замовлень; updateStatus() – Оновлення статусу замовлення.

Таблиця 2.30 – Специфікація класу OrdersController

Характеристика	Опис
Відповідальність	Контролер для обробки перегляду, створення та скасування замовлень.
Атрибути	Відсутні
Методи	ordersPage() – Відображення сторінки з списком замовлень користувача; store() – Створення замовлення; destroy() – Скасування замовлення; isActiveOrder() – Перевірка чи активне замовлення;

Таблиця 2.31 – Специфікація класу OrdersModel

Характеристика	Опис
Відповідальність	Модель замовлень, що представляє таблицю orders бази даних.
Атрибути	fillable – Список атрибутів бази даних, які дозволено передавати при масовому заповненні.
Методи	smartphones() – Зв'язок з моделлю SmartphonesModels; user() – Зв'язок з моделлю UsersModels;

Таблиця 2.32 – Специфікація класу UsersController

Характеристика	Опис
1	2
Відповідальність	Контролер для керування інформацією про користувача, реєстрації та авторизації користувача.
Атрибути	Відсутні

Кінець таблиці 2.32

1	2
Методи	logout() – Вихід з облікового запису; editPage() – Сторінка редагування інформації про користувача; edit() – Оновлення інформації про користувача; passwordUpdate() – Оновлення паролю користувача; login() – Авторизація в обліковий запис; loginPage() – Сторінка авторизації register() – Реєстрація облікового запису; registerPage() – Сторінка реєстрації облікового запису.

Таблиця 2.33 – Специфікація класу UserModel

Характеристика	Опис
Відповідальність	Модель користувачів, що представляє таблицю users бази даних.
Атрибути	fillable – Список атрибутів бази даних, які дозволено передавати при масовому заповненні.
Методи	casts() – Автоматична обробка значень полів моделі при запису чи зчитуванні даних; reviews() – Зв'язок з моделлю ReviewsModels; orders() – Зв'язок з моделлю OrdersModels.

Побудова динамічної моделі вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Динамічна модель представлена у вигляді діаграм послідовності.

Діаграма послідовності – використовуються для моделювання логіки сценаріїв використання, показуючи інформацію, що передається між об'єктами в системі під час виконання сценарію. Діаграма показує, як взаємодіють об'єкти, що використовуються у сценарії, але не те, як вони пов'язані один з одним. Діаграми послідовності також часто використовуються для того, щоб показати взаємодію компонентів інтерфейсу користувача або компонентів програмного забезпечення [4].

Діаграми послідовності для прецедентів «Видалення замовлення» та «Розміщення відгуків» представлені на рисунках 2.13 та 2.14.

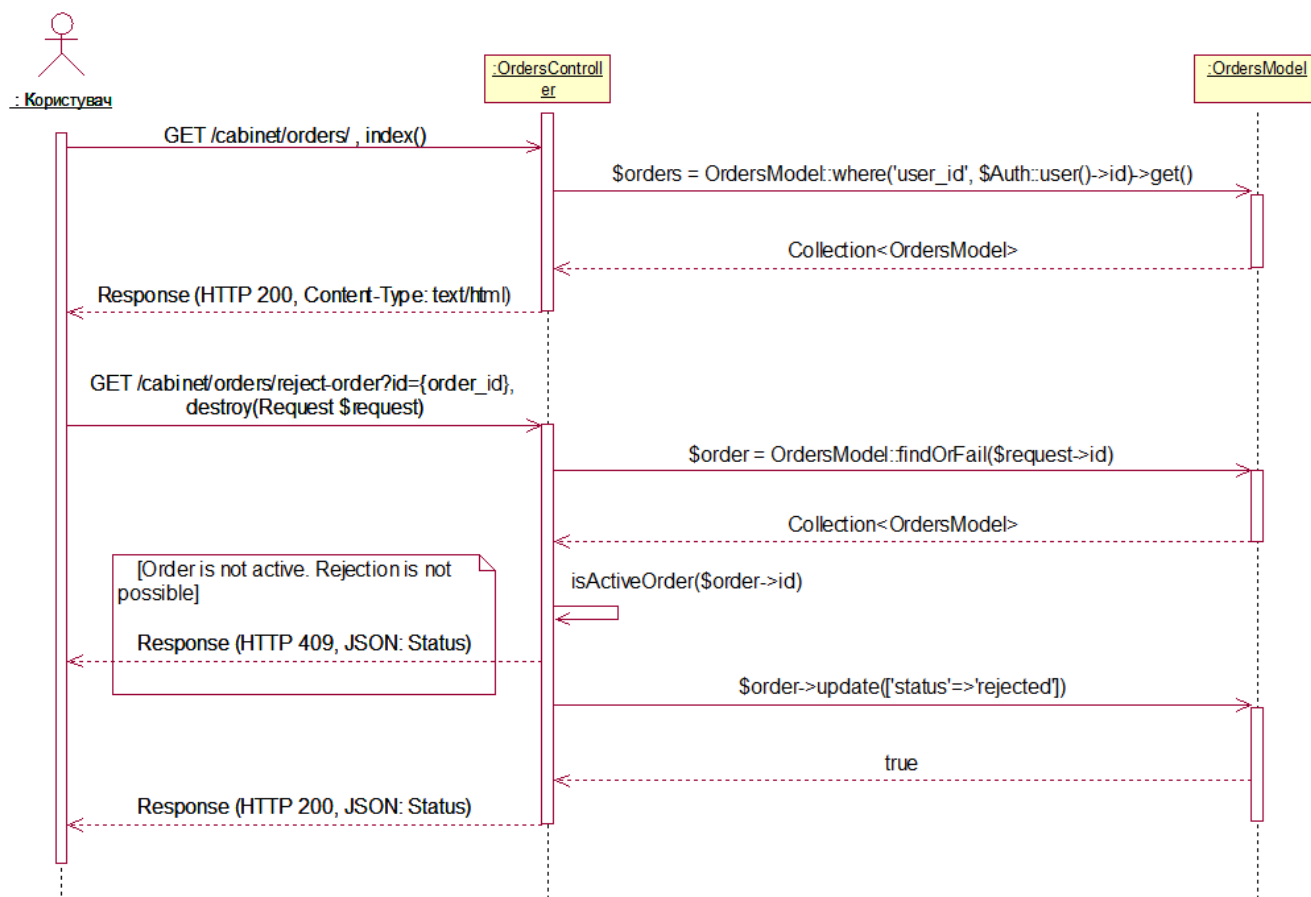


Рисунок 2.13 – Діаграма послідовності для прецеденту «Видалення замовлення»

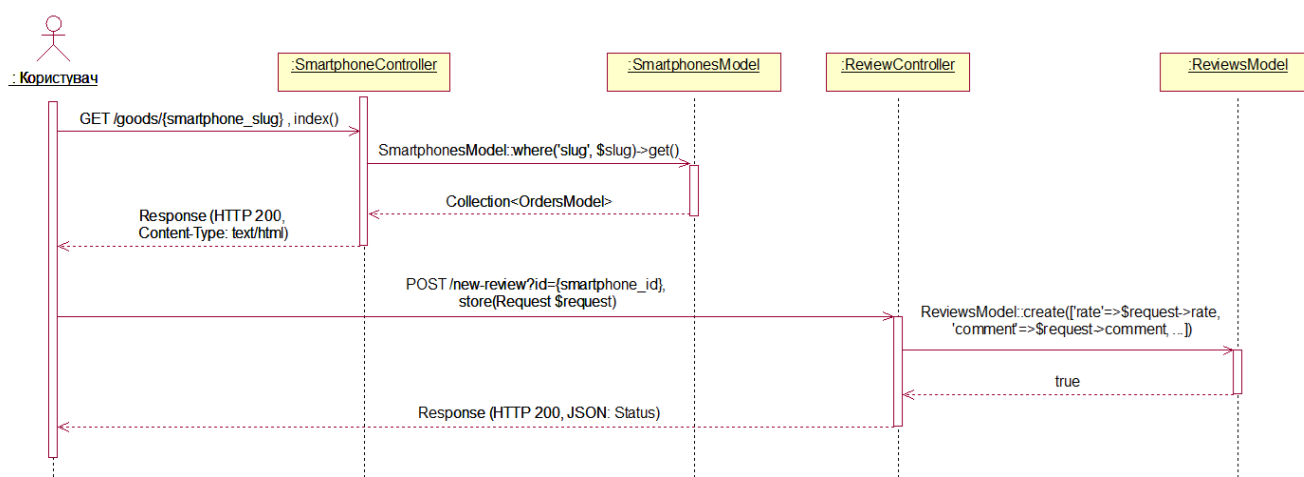


Рисунок 2.14 – Діаграма послідовності для прецеденту «Розміщення відгуків»

Розробка логічної моделі даних вебзастосунку для інтернет-магазину смартфонів «Smartafon».

Розробка логічної моделі – це побудова структури бази даних, що відображає логічні зв'язки між елементами даних, незалежно від конкретної фізичної реалізації. Логічна модель даних деталізує концептуальну модель для кращого розуміння структури та взаємозв'язків у системі.

Для побудови логічної моделі даних вебзастосунку для інтернет-магазину смартфонів «Smartafon» була обрана реляційна модель даних. Логічна модель даних була розроблена за допомогою програмного забезпечення Erwin та представлена на рисунку 2.15.

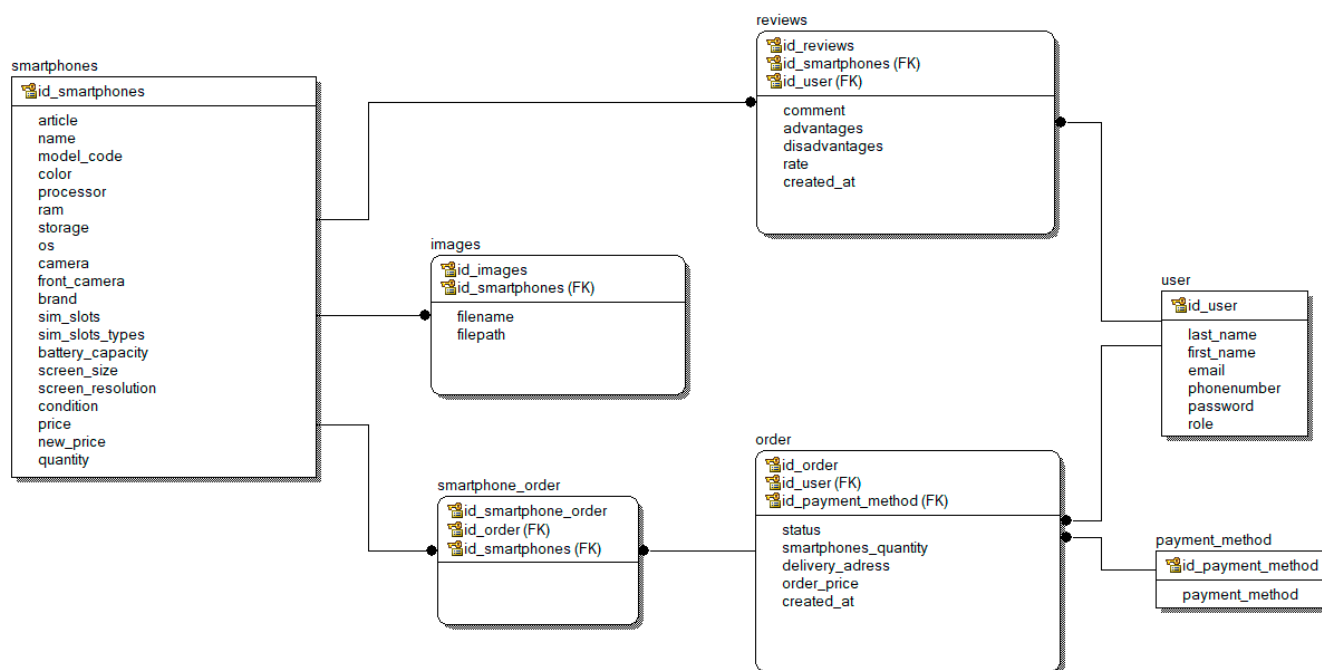


Рисунок 2.15 – Логічна модель даних вебзастосунку для інтернет-магазину смартфонів «Smartafon»

У розробленій логічній моделі даних представлено основні сутності, їхні атрибути та типи зв'язків між ними, що дозволяє краще зрозуміти структуру даних і логіку взаємозв'язків.

2.3.3 Робочий проєкт вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Обґрунтування вибору засобів програмування вебзастосунку для інтернет-магазину смартфонів «Smartafon»

В якості засобів розробки вебзастосунку для інтернет-магазину смартфонів «Smartafon» були використані наступні:

- VS Code: Для написання коду вебзастосунку використовувався універсальний та безкоштовний редактор коду Visual Studio Code від Microsoft. Цей редактор розповсюджується безкоштовно і доступний для таких платформ як Windows, Linux і OS X;
- PHP: Для написання коду для серверної частини вебзастосунку було обрано скриптову мову програмування PHP, яка, в свою чергу, є однією з найпоширеніших мов, що використовуються у сфері веброзробок. Вона виконується на сервері та генерує HTML-код, який відправляється користувачеві. PHP також підтримує роботу з різними базами даних, такими як MySQL, PostgreSQL та SQLite;
- JavaScript: Високорівнева мова програмування, що використовується для створення інтерактивних вебсторінок і вебдодатків. Вона дозволяє динамічно взаємодіяти з HTML і CSS, що дозволяє змінювати вміст сторінки без перезавантаження. Зазвичай ця мова виконується на стороні клієнта (тобто у браузері), але також може працювати на сервері за допомогою платформ, таких як Node.js. Завдяки постійному розвитку стандарту ECMAScript, JavaScript залишається однією з найпопулярніших мов програмування у світі, яка використовується не тільки у веброзробці, але й у мобільних, серверних та навіть десктопних застосунках;
- CSS: Це мова стилів, яка використовується для оформлення вебсторінок. Вона визначає вигляд HTML-елементів, включаючи кольори, шрифти, розміри, відступи, позиціонування та анімації. CSS працює за принципом

каскадності, що означає, що стилі можуть успадковуватися від батьківських елементів і перекриватися залежно від специфічності правил, що дозволяє гнучко керувати зовнішнім виглядом сайту;

- **Laravel:** Це популярний PHP-фреймворк, створений для легкої та ефективної розробки вебдодатків. Він використовує архітектурний шаблон MVC (Model-View-Controller), що забезпечує чітке розділення логіки програми, представлення та управління даними та містить вбудовані інструменти для роботи з маршрутизацією, базами даних, аутентифікацією, чергами завдань і кешуванням. Підтримує велику кількість пакетів та інструментів для роботи з frontend фреймворками;

- **SCSS:** Це розширення мови CSS, яке додає можливість використання змінних, вкладеності, функцій та інших зручних інструментів для спрощення та оптимізації стилізації вебсторінок. SCSS є частиною препроцесора SASS і має синтаксис, схожий на звичайний CSS, але з додатковими можливостями, що дозволяє писати більш структурований і зручний код, який легше підтримувати, ніж звичайний CSS;

- **Jquery:** Це популярна бібліотека JavaScript, що була створена для спрощення роботи з HTML-документами, маніпуляції DOM, обробки подій, анімацій та взаємодії з сервером за допомогою AJAX. Дозволяє писати менш громіздкий та більш кросбраузерний код, що дозволяє підтримувати сумісність з різними версіями браузерів;

- **phpMyAdmin:** Це вебзастосунок, що написаний на PHP і створений для зручного керування базами даних MySQL і MariaDB через інтуїтивно зрозумілий вебінтерфейс. Він дозволяє легко виконувати операції з базами даних без необхідності використання командного рядка.

Реалізація основних класів вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Вебзастосунок написаний мовою PHP з використанням серверного

фреймворку Laravel та системою керування базами даних MySQL. Код для декількох основних класів представлено нижче. Повний код вебзастосунку для інтернет-магазину смартфонів «Smartafon» представлено у додатку Д.

Admin/SmartphoneController.php

```
<?php

namespace App\Http\Controllers\AdminControllers;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Auth;
use App\Models\SmartphoneModel;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Gate;
use App\Services\SmartphoneImageService;

class SmartphoneController extends Controller {

    public function index() {
        Gate::denies('access-admin') && abort(403);
        $smartphones = SmartphoneModel::all();

        return view('admin.smartphonesTable', compact('smartphones'));
    }

    public function createPage() {
        Gate::denies('access-admin') && abort(403);

        return view('admin.smartphonesFormPage');
    }

    public function store(Request $request, SmartphoneImageService $imageService) {
        Gate::denies('access-admin') && abort(403);

        $smartphone = SmartphoneModel::create([
            'article' => $request->article,
            'name' => $request->name,
            'slug' => Str::slug($request->brand) . '-' . Str::slug($request->model_code) .
            '-' . Str::slug($request->article),
            'model_code' => $request->model_code,
            'color' => $request->color,
            'processor' => $request->processor,
            'ram' => $request->ram,
            'storage' => $request->storage,
            'os' => $request->os,
            'camera' => $request->camera,
            'front_camera' => $request->front_camera,
            'brand' => $request->brand,
            'sim_slots' => $request->sim_slots,
            'sim_slots_types' => $request->sim_slots_types,
            'battery_capacity' => $request->battery_capacity,
            'screen_size' => $request->screen_size,
            'screen_resolution' => $request->screen_resolution,
            'condition' => $request->condition,
```

```

        'price' => $request->price,
        'quantity' => $request->quantity,
    ]]);

    if ($request->hasFile('images')) {
        $imageService->addImages($request->file('images'), $smartphone->id);
    }

    return response()->json(['success' => true]);
}

public function editPage(Request $request) {
    Gate::denies('access-admin') && abort(403);
    $smartphone = SmartphoneModel::findOrFail($request->id);
    return view('admin.smartphonesFormPage', compact('smartphone'));
}

public function update(Request $request, SmartphoneImageService $imageService) {
    Gate::denies('access-admin') && abort(403);
    $smartphone = SmartphoneModel::findOrFail($request->id);

    $smartphone->update([
        'article' => $request->article,
        'name' => $request->name,
        'model_code' => $request->model_code,
        'color' => $request->color,
        'processor' => $request->processor,
        'ram' => $request->ram,
        'storage' => $request->storage,
        'os' => $request->os,
        'camera' => $request->camera,
        'front_camera' => $request->front_camera,
        'brand' => $request->brand,
        'sim_slots' => $request->sim_slots,
        'sim_slots_types' => $request->sim_slots_types,
        'battery_capacity' => $request->battery_capacity,
        'screen_size' => $request->screen_size,
        'screen_resolution' => $request->screen_resolution,
        'condition' => $request->condition,
        'price' => $request->price,
        'new_price' => $request->new_price,
        'quantity' => $request->quantity,
    ]);

    if ($request->hasFile('images')) {
        $imageService->updateImages($request->file('images'), $request->id);
    }

    return response()->json(['success' => true]);
}

public function destroy(Request $request, SmartphoneImageService $imageService) {
    Gate::denies('access-admin') && abort(403);
    $smartphone = SmartphoneModel::findOrFail($request->id);

    $imageService->deleteImages($smartphone->id);
    $smartphone->delete();

    return redirect()->route('admin.smartphones');
}

```

```

    }
}

```

SmartphoneImageService.php

```

<?php
namespace App\Services;
use App\Models\Image;
use Illuminate\Http\UploadedFile;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Str;

class SmartphoneImageService
{
    public function addImages(array $images, int $smartphoneId): void
    {
        foreach ($images as $image) {
            if (!$image->isValid()) continue;

            $extension = $image->getClientOriginalExtension();
            $filename = Str::random(40) . '.' . $extension;

            Storage::disk('public')->putFileAs('images', $image, $filename);

            Image::create([
                'filename' => $filename,
                'filepath' => '/storage/images/' . $filename,
                'smartphone_id' => $smartphoneId,
            ]);
        }
    }

    public function updateImages(array $images, int $smartphoneId): void
    {
        $oldImages = Image::where('smartphone_id', $smartphoneId)->get();

        foreach ($oldImages as $oldImage) {
            $imagePath = str_replace('/storage/', '', $oldImage->filepath);
            Storage::disk('public')->delete($imagePath);
            $oldImage->delete();
        }

        foreach ($images as $image) {
            $extension = $image->getClientOriginalExtension();
            $filename = Str::random(40) . '.' . $extension;
            Storage::disk('public')->putFileAs('images', $image, $filename);

            Image::create([
                'filename' => $filename,
                'filepath' => '/storage/images/' . $filename,
                'smartphone_id' => $smartphoneId,
            ]);
        }
    }

    public function deleteImages(int $smartphoneId): void
    {
        $images = Image::where('smartphone_id', $smartphoneId)->get();

        foreach ($images as $image) {

```

```

        $imagePath = str_replace('/storage/', '', $image->filepath ?? $image->url);
        Storage::disk('public')->delete($imagePath);
        $image->delete();
    }
}
}

```

Тестування та випробування вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Тестування вебзастосунку необхідне для перевірки його працездатності, стабільності та відповідності заданим вимогам, що описані в технічному завданні. Воно дозволяє виявити помилки, недоліки та потенційні вразливості ще до впровадження програмного продукту. Тестування також дає змогу переконатися, що вебзастосунок коректно взаємодіє з іншими компонентами застосунку та правильно реагує на різні сценарії використання.

Тестування прецеденту U2 «Реєстрація», що виконується методом «чорної скриньки», представлено нижче.

Тестування прецеденту U2 «Реєстрація»:

Перевірити, що перевіряється наявність введених обов'язкових даних в формі реєстрації.

Класи еквівалентності:

а) відсутність вхідних даних

Тестовий випадок для класу еквівалентності а:

Ідентифікатор: TEST-REG-EMPTY-001

Предмети тестування: Форма реєстрації

Специфікація вхідних даних:

- ПІБ: (пусто);
- Електронна пошта: (пусто);
- Пароль користувача: (пусто);
- Номер телефону користувача: (пусто).

Специфікація результатів: Відображається повідомлення про необхідність заповнити ці поля.

Тестове оточення: Веббраузер

Спеціальні процедурні вимоги: Відсутні

Залежність з іншими тестами: Відсутнє

Дії користувача:

- Відкрити сторінку реєстрації;
- Не заповнюючи обов'язкові поля, натиснути кнопку «Зареєструватись»
- Перевірити, чи з'являється повідомлення про необхідність заповнити

ці поля.

Перевірити, що правильно обробляються неправильні дані (пошта, пароль чи номер телефону).

Класи еквівалентності:

- a) Пошта не має правильної структури
- b) Пошта містить пробіли
- c) Пошта містить заборонені символи
- d) Пароль містить пробіли
- e) Пароль надто короткий (Граничні значення: 8)
- f) Номер телефону надто короткий (Граничні значення: 10)
- g) Номер телефону містить не лише цифри
- h) Номер телефону занадто довгий (Граничні значення: 10)

Тестовий випадок для класу еквівалентності a:

Ідентифікатор: TEST-REG-EMAIL-001

Предмети тестування: Форма реєстрації

Специфікація вхідних даних:

- ПІБ: Іваненко Іван Іванович;
- Електронна пошта: ivanenko@ (відсутній домен);
- Пароль користувача: 8A@_f2&7*0*Gd;

- Номер телефону користувача: 0955572555

Специфікація результатів: Відображаються повідомлення про неправильний формат пошти.

Тестове оточення: Веббраузер

Спеціальні процедурні вимоги: Відсутні

Залежність з іншими тестами: Відсутнє

Дії користувача:

- Відкрити сторінку реєстрації;
- Заповнити поля: ПІБ: Іваненко Іван Іванович; Електронна пошта: ivanenko@; Пароль користувача: 8A@_f2&7*0*Gd; Номер телефону користувача: 0955572555;
- Натиснути кнопку «Зареєструватись»
- Перевірити, чи з'являється повідомлення про помилку «Неправильний формат пошти».

Тестовий випадок для класу еквівалентності b:

Ідентифікатор: TEST-REG-EMAIL-002

Предмети тестування: Форма реєстрації

Специфікація вхідних даних:

- ПІБ: Іваненко Іван Іванович;
- Електронна пошта: ivan enko@gmail.com (містить пробіл);
- Пароль користувача: 8A@_f2&7*0*Gd;
- Номер телефону користувача: +380955572555

Специфікація результатів: Відображаються повідомлення про помилку «Пошта не повинна містити пробіли».

Тестове оточення: Веббраузер

Спеціальні процедурні вимоги: Відсутні

Залежність з іншими тестами: Відсутнє

Дії користувача:

- Відкрити сторінку реєстрації;

- Заповнити поля: ПІБ: Іваненко Іван Іванович; Електронна пошта: ivanenko@gmail.com; Пароль користувача: 8A@_f2&7*0*Gd; Номер телефону користувача: 0955572555;
- Натиснути кнопку «Зареєструватись»
- Перевірити, чи з'являється повідомлення про помилку «Пошта не повинна містити пробіли».

Тестовий випадок для класу еквівалентності с:

Ідентифікатор: TEST-REG-EMAIL-003

Предмети тестування: Форма реєстрації

Специфікація вхідних даних:

- ПІБ: Іваненко Іван Іванович;
- Електронна пошта: i#van&enko@gmail.com (заборонені символи);
- Пароль користувача: 8A@_f2&7*0*Gd;
- Номер телефону користувача: 0955572555

Специфікація результатів: Відображаються повідомлення про помилку «Пошта не повинна містити заборонені символи».

Тестове оточення: Веббраузер

Спеціальні процедурні вимоги: Відсутні

Залежність з іншими тестами: Відсутнє

Дії користувача:

- Відкрити сторінку реєстрації;
- Заповнити поля: ПІБ: Іваненко Іван Іванович; Електронна пошта: i#van&enko@gmail.com; Пароль користувача: 8A@_f2&7*0*Gd; Номер телефону користувача: 0955572555;
- Натиснути кнопку «Зареєструватись»
- Перевірити, чи з'являється повідомлення про помилку «Пошта не повинна містити заборонені символи».

Тестовий випадок для класу еквівалентності d:

Ідентифікатор: TEST-REG-PASSWORD-001

Предмети тестування: Форма реєстрації

Специфікація вхідних даних:

- ПІБ: Іваненко Іван Іванович;
- Електронна пошта: ivanenko@gmail.com;
- Пароль користувача: 1234 4567 (Містить пробіли);
- Номер телефону користувача: 0955572555

Специфікація результатів: Відображаються повідомлення про помилку «Пароль не повинен містити пробіли».

Тестове оточення: Веббраузер

Спеціальні процедурні вимоги: Відсутні

Залежність з іншими тестами: Відсутнє

Дії користувача:

- Відкрити сторінку реєстрації;
- Заповнити поля: ПІБ: Іваненко Іван Іванович; Електронна пошта: ivanenko@gmail.com; Пароль користувача: 1234 4567; Номер телефону користувача: 0955572555;
- Натиснути кнопку «Зареєструватись»
- Перевірити, чи з'являється повідомлення про помилку «Пароль не повинен містити пробіли».

Тестовий випадок для класу еквівалентності e:

Ідентифікатор: TEST-REG-PASSWORD-002

Предмети тестування: Форма реєстрації

Специфікація вхідних даних:

- ПІБ: Іваненко Іван Іванович;
- Електронна пошта: ivanenko@gmail.com;
- Пароль користувача: 1234567 (Надто короткий);
- Номер телефону користувача: 0955572555

Специфікація результатів: Відображаються повідомлення про помилку «Довжина паролю повинна бути не менше 8 символів».

Тестове оточення: Веббраузер

Спеціальні процедурні вимоги: Відсутні

Залежність з іншими тестами: Відсутнє

Дії користувача:

- Відкрити сторінку реєстрації;
- Заповнити поля: ПІБ: Іваненко Іван Іванович; Електронна пошта: ivanenko@gmail.com; Пароль користувача: 1234567; Номер телефону користувача: 0955572555;
- Натиснути кнопку «Зареєструватись»
- Перевірити, чи з'являється повідомлення про помилку «Довжина паролю повинна бути не менше 8 символів».

Тестовий випадок для класу еквівалентності f:

Ідентифікатор: TEST-REG-PHONENUM-001

Предмети тестування: Форма реєстрації

Специфікація вхідних даних:

- ПІБ: Іваненко Іван Іванович;
- Електронна пошта: ivanenko@gmail.com;
- Пароль користувача: 8A@_f2&7*0*Gd;
- Номер телефону користувача: 095557255 (Номер надто короткий. Містить менше ніж 10 цифр.)

Специфікація результатів: Відображаються повідомлення про помилку «Номер телефону занадто короткий».

Тестове оточення: Веббраузер

Спеціальні процедурні вимоги: Відсутні

Залежність з іншими тестами: Відсутнє

Дії користувача:

- Відкрити сторінку реєстрації;

- Заповнити поля: ПІБ: Іваненко Іван Іванович; Електронна пошта: ivanenko@gmail.com; Пароль користувача: 8A@_f2&7*0*Gd; Номер телефону користувача: 095557255;
- Натиснути кнопку «Зареєструватись»
- Перевірити, чи з'являється повідомлення про помилку «Номер телефону занадто короткий».

Тестовий випадок для класу еквівалентності g:

Ідентифікатор: TEST-REG-PHONENUM-002

Предмети тестування: Форма реєстрації

Специфікація вхідних даних:

- ПІБ: Іваненко Іван Іванович;
- Електронна пошта: ivanenko@gmail.com;
- Пароль користувача: 8A@_f2&7*0*Gd;
- Номер телефону користувача: (095)5570555 (Номер містить не тільки цифри)

Специфікація результатів: Відображаються повідомлення про помилку «Номер телефону повинен містити лише цифри».

Тестове оточення: Веббраузер

Спеціальні процедурні вимоги: Відсутні

Залежність з іншими тестами: Відсутнє

Дії користувача:

- Відкрити сторінку реєстрації;
- Заповнити поля: ПІБ: Іваненко Іван Іванович; Електронна пошта: ivanenko@gmail.com; Пароль користувача: 8A@_f2&7*0*Gd; Номер телефону користувача: (095)5570555;
- Натиснути кнопку «Зареєструватись»
- Перевірити, чи з'являється повідомлення про помилку «Номер телефону повинен містити лише цифри»

Тестовий випадок для класу еквівалентності h:

Ідентифікатор: TEST-REG-PHONENUM-003

Предмети тестування: Форма реєстрації

Специфікація вхідних даних:

- ПІБ: Іваненко Іван Іванович;
- Електронна пошта: ivanenko@gmail.com;
- Пароль користувача: 8A@_f2&7*0*Gd;
- Номер телефону користувача: 09555725553 (Номер занадто довгий.

Містить більше ніж 10 цифр.)

Специфікація результатів: Відображаються повідомлення про помилку «Номер телефону занадто довгий».

Тестове оточення: Веббраузер

Спеціальні процедурні вимоги: Відсутні

Залежність з іншими тестами: Відсутнє

Дії користувача:

- Відкрити сторінку реєстрації;
- Заповнити поля: ПІБ: Іваненко Іван Іванович; Електронна пошта: ivanenko@gmail.com; Пароль користувача: 8A@_f2&7*0*Gd; Номер телефону користувача: 09555725553;
- Натиснути кнопку «Зареєструватись»
- Перевірити, чи з'являється повідомлення про помилку «Номер телефону занадто довгий»

Перевірити, що перевіряється, що обліковий запис з таким номером телефоном, або поштою не існує і не створюється другий обліковий запис з цими даними.

Класи еквівалентності:

- a) Вже зареєстрована пошта
- b) Вже зареєстрований номер телефону

Тестовий випадок для класу еквівалентності а:

Ідентифікатор: TEST-REG-EMAIL-004

Предмети тестування: Форма реєстрації

Специфікація вхідних даних:

- ПІБ: Іваненко Іван Іванович;
- Електронна пошта: user@gmail.com (Вже зареєстрована пошта)
- Пароль користувача: 8A@_f2&7*0*Gd;
- Номер телефону користувача: 0955572555.

Специфікація результатів: Відображаються повідомлення про помилку «Обліковий запис з такою поштою вже зареєстровано».

Тестове оточення: Веббраузер

Спеціальні процедурні вимоги: В системі має бути створений обліковий запис з цією електронною поштою.

Залежність з іншими тестами: Відсутня

Дії користувача:

- Відкрити сторінку реєстрації;
- Заповнити поля: ПІБ: Іваненко Іван Іванович; Електронна пошта: user@gmail.com; Пароль користувача: 8A@_f2&7*0*Gd; Номер телефону користувача: 0955572555;
- Натиснути кнопку «Зареєструватись»
- Перевірити, чи з'являється повідомлення про помилку «Обліковий запис з такою поштою вже зареєстровано».

Тестовий випадок для класу еквівалентності b:

Ідентифікатор: TEST-REG-PASSWORD-003

Предмети тестування: Форма реєстрації

Специфікація вхідних даних:

- ПІБ: Іваненко Іван Іванович;
- Електронна пошта: ivanenko@gmail.com
- Пароль користувача: 8A@_f2&7*0*Gd;

– Номер телефону користувача: 0754443360 (Вже зареєстрований номер телефону).

Специфікація результатів: Відображаються повідомлення про помилку «Обліковий запис з таким номером телефону вже зареєстровано».

Тестове оточення: Веббраузер

Спеціальні процедурні вимоги: В системі має бути створений обліковий запис з цим номером телефону.

Залежність з іншими тестами: Відсутня

Дії користувача:

- Відкрити сторінку реєстрації;
- Заповнити поля: ПІБ: Іваненко Іван Іванович; Електронна пошта: ivanenko@gmail.com; Пароль користувача: 8A@_f2&7*0*Gd; Номер телефону користувача: 0754443360;
- Натиснути кнопку «Зареєструватись»
- Перевірити, чи з'являється повідомлення про помилку «Обліковий запис з таким номером телефону вже зареєстровано».

Перевірити, що при правильному введенні даних, виконується реєстрація.

Класи еквівалентності:

- а) Всі дані валідні

Тестовий випадок для класу еквівалентності а:

Ідентифікатор: TEST-REG-VALID-001

Предмети тестування: Форма реєстрації

Специфікація вхідних даних:

- ПІБ: Іваненко Іван Іванович;
- Електронна пошта: ivanenko.ii@gmail.com (валідна електронна пошта);
- Пароль користувача: 8A@_f2&7*0*Gd (валідний пароль, містить більше 7 символів);

– Номер телефону користувача: 0754243942 (валідний номер телефону, містить 10 цифр).

Специфікація результатів: Користувач успішно реєструється у системі.

Тестове оточення: Веббраузер

Спеціальні процедурні вимоги: Відсутні

Залежність з іншими тестами: Відсутні

Дії користувача:

- Відкрити сторінку реєстрації;
- Заповнити поля: ПІБ: Іваненко Іван Іванович; Електронна пошта: ivanenko.ii@gmail.com; Пароль користувача: 8A@_f2&7*0*Gd; Номер телефону користувача: 0754243942;
- Натиснути кнопку «Зареєструватись»
- Перевірити, чи з'являється підтвердження про успішну реєстрацію нового облікового запису.

Випробування вебзастосунку для інтернет-магазину смартфонів «Smartafon»

Випробування вебзастосунку для інтернет-магазину смартфонів «Smartafon» виконується згідно з програми та методики випробувань, що наведені у додатку В.

Під час випробувань вебзастосунку для інтернет-магазину смартфонів «Smartafon» було перевірено авторизацію з використанням коректних даних, додавання нового смартфона, редагування інформації вже існуючого смартфона, видалення відгуків адміністратором.

1) Випробування авторизації у вебзастосунку для інтернет-магазину смартфонів «Smartafon» з використанням коректних даних

Після відкриття вебзастосунку у веббраузері, натиснемо на сіру кнопку «Увійти», що знаходиться у правій частині шапки вебзастосунку. Після переходу до сторінки авторизації, заповнимо поля коректною інформацією та натиснемо на кнопку «Увійти». У результаті авторизації нас перенаправило на головну сторінку вебзастосунку, а кнопка «Увійти» замінилась на кнопку «Профіль», що свідчить

про те, що випробування пройшло успішно. Результат випробування представлено на рисунку 2.16.

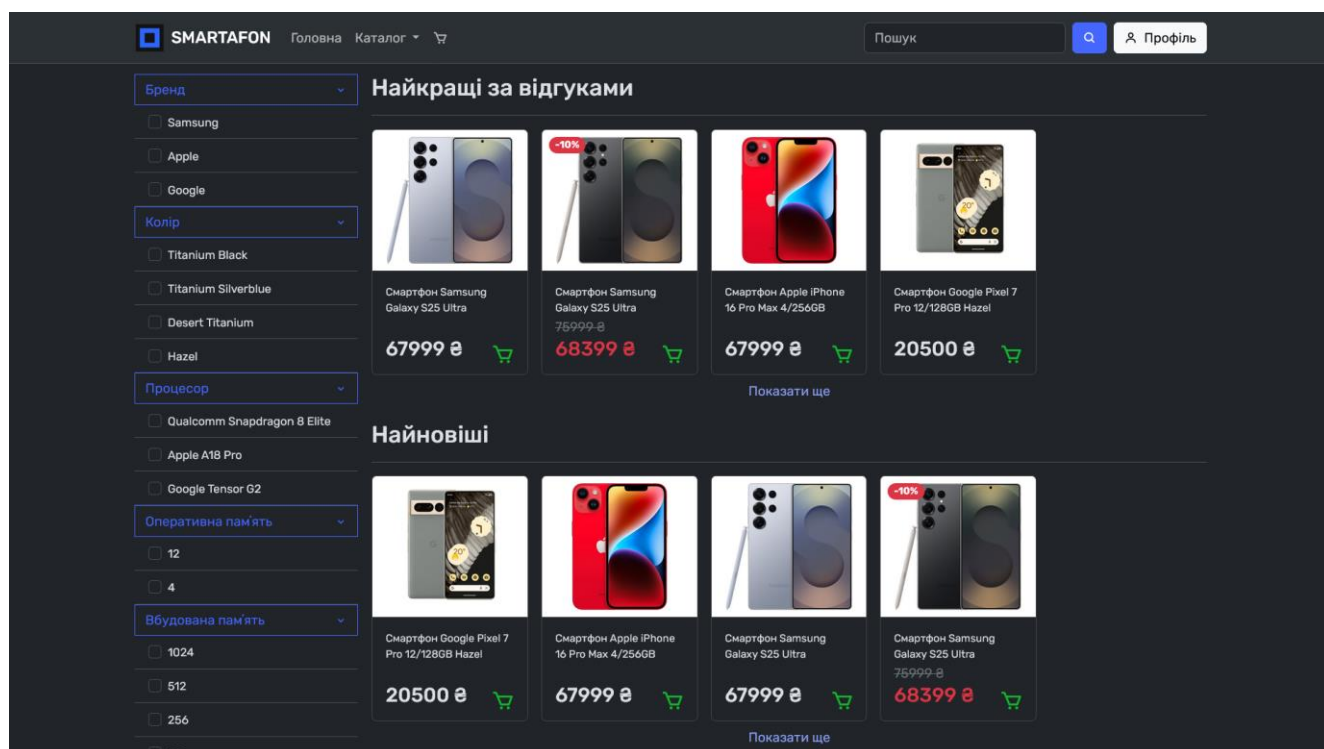


Рисунок 2.16 – Результат випробування авторизації у вебзастосунку для інтернет-магазину смартфонів «Smartafon» з використанням коректних даних

2) Випробування додавання нового смартфона

Після відкриття вебзастосунку у веббраузері та авторизації у обліковий запис адміністратора, натиснемо на кнопку «Адмін», що знаходиться у правій частині шапки вебзастосунку. Після переходу до сторінки адміністрування, відкриємо меню «Функції для адміністратора» та оберемо пункт «Додати новий смартфон», щоб перейти на сторінку з формою для додавання нового смартфона. Заповнимо всі поля форми даними та натиснемо «Далі», після чого оберемо зображення для нового смартфона та натиснемо «Додати новий смартфон». У результаті додавання нового смартфона, нас перенаправило на адміністративну сторінку з списком усіх смартфонів у вебзастосунку, який містить інформацію з зображенням смартфона, який ми тільки що додавали. Також було відображено повідомлення про успішне додавання нового смартфона. Це свідчить про те, що випробування пройшло

успішно. Результат випробування представлено на рисунку 2.17.

#	Артикул	Фото	Назва	Ціна	Акційна ціна	Код моделі	Колір	Процесор	ОЗУ	Пам'ять	ОС	Камера	Фронтальна камера	Бренд
1	473275129		Samsung Galaxy S25 Ultra	75999	68399	SM-S938BZKHEUC	Titanium Black	Qualcomm Snapdragon 8 Elite	12 ГБ	1024 ГБ	Android 16	200 Мп + 50 Мп + 50 Мп + 10 Мп	12 Мп	Samsung
2	473275199		Samsung Galaxy S25 Ultra	67999		SM-S938BZBGEUC	Titanium Silverblue	Qualcomm Snapdragon 8 Elite	12 ГБ	512 ГБ	Android 16	200 Мп + 50 Мп + 50 Мп + 10 Мп	12 Мп	Samsung
3	448428623		Apple iPhone 16 Pro Max	67999		MYWX3SX/A	Desert Titanium	Apple A18 Pro	4 ГБ	256 ГБ	iOS	48 Мп + 48 Мп + 12 Мп	12 Мп	Apple
4	485955124		Google Pixel 7 Pro	20500		0810029937443	Hazel	Google Tensor G2	12 ГБ	128 ГБ	Android 14	50 Мп + 48 Мп + 12 Мп	10.8	Google
10	87750274		Samsung Galaxy A70	9999		SM-A705FZKUSEK	Білий	Qualcomm Snapdragon 675	6 ГБ	128 ГБ	Android 9	32 Мп + 5 Мп + 8 Мп	32 Мп	Samsung

Новий смартфон було успішно додано! ✕

Рисунок 2.17 – Результат випробування додавання нового смартфона

3) Випробування редагування інформації вже існуючого смартфона

Після відкриття вебзастосунку у веббраузері та авторизації у обліковий запис адміністратора, натиснемо на кнопку «Адмін», що знаходиться у правій частині шапки вебзастосунку. Після переходу до сторінки адміністрування, відкриємо меню «Функції для адміністратора» та оберемо пункт «Список смартфонів», щоб перейти на сторінку з списком всіх смартфонів, що доступні у вебзастосунку. Прокрутимو таблицю вправо до самого кінця та натиснемо кнопку «Редагувати», що знаходиться в тому ж самому рядку, що й інша інформація про смартфон, який ми хочемо відредагувати. Після відкриття сторінки з формою відредагуємо потрібні нам поля форми та натиснемо «Далі» і «Зберегти зміни». У результаті редагування інформації про смартфон, нас перенаправило на адміністративну сторінку з списком усіх смартфонів у вебзастосунку, який містить оновлену інформацію про смартфон, який ми тільки що редагували та отримали повідомлення про успішне редагування інформації вже існуючого смартфона. Це

свідчить про те, що випробування пройшло успішно. Результат випробування представлено на рисунку 2.18.

#	Артикуль	Фото	Назва	Ціна	Акційна ціна	Код моделі	Колір	Процесор	ОЗУ	Пам'ять	ОС	Камера	Фронтальна камера	Бренд
1	473275129		Samsung Galaxy S25 Ultra	75999	68399	SM-S938BZKNEUC	Titanium Black	Qualcomm Snapdragon 8 Elite	12 ГБ	1024 ГБ	Android 16	200 Mn + 50 Mn + 50 Mn + 10 Mn	12 Mn	Samsung
2	473275199		Samsung Galaxy S25 Ultra	67999		SM-S938BZBGEUC	Titanium Silverblue	Qualcomm Snapdragon 8 Elite	12 ГБ	512 ГБ	Android 16	200 Mn + 50 Mn + 50 Mn + 10 Mn	12 Mn	Samsung
3	448428623		Apple iPhone 16 Pro Max	67999		MYWX3SX/A	Desert Titanium	Apple A18 Pro	4 ГБ	256 ГБ	iOS	48 Mn + 48 Mn + 12 Mn	12 Mn	Apple
4	485955124		Google Pixel 7 Pro	20500		0810029937443	Hazel	Google Tensor G2	12 ГБ	128 ГБ	Android 14	50 Mn + 48 Mn + 12 Mn	10.8	Google
10	87750274		Samsung Galaxy A70	9999		SM-A705FZKUSEK	Білий	Qualcomm Snapdragon 675	6 ГБ	128 ГБ	Android 9	32 Mn + 5 Mn + 8 Mn	32 Mn	Samsung

Рисунок 2.18 – Результат редагування інформації вже існуючого смартфона

4) Випробування видалення відгуків адміністратором

Після відкриття вебзастосунку у веббраузері та авторизації у обліковий запис адміністратора, натиснемо на кнопку «Адмін», що знаходиться у правій частині шапки вебзастосунку. Після переходу до сторінки адміністрування, відкриємо меню «Функції для адміністратора» та оберемо пункт «Список відгуків», щоб перейти на сторінку з списком всіх відгуків, що були написані користувачами. Натиснемо кнопку «Видалити», що знаходиться в тому ж самому рядку, що й інша інформація про відгук, який ми хочемо видалити. Натиснемо на кнопку «Видалити відгук» у модальному вікні з повідомленням про підтвердження видалення. У результаті видалення відгуку, він зник з списку, а ми отримали повідомлення про успішне видалення відгуку користувача. Це свідчить про те, що випробування пройшло успішно. Результат випробування представлено на рисунку 2.19.

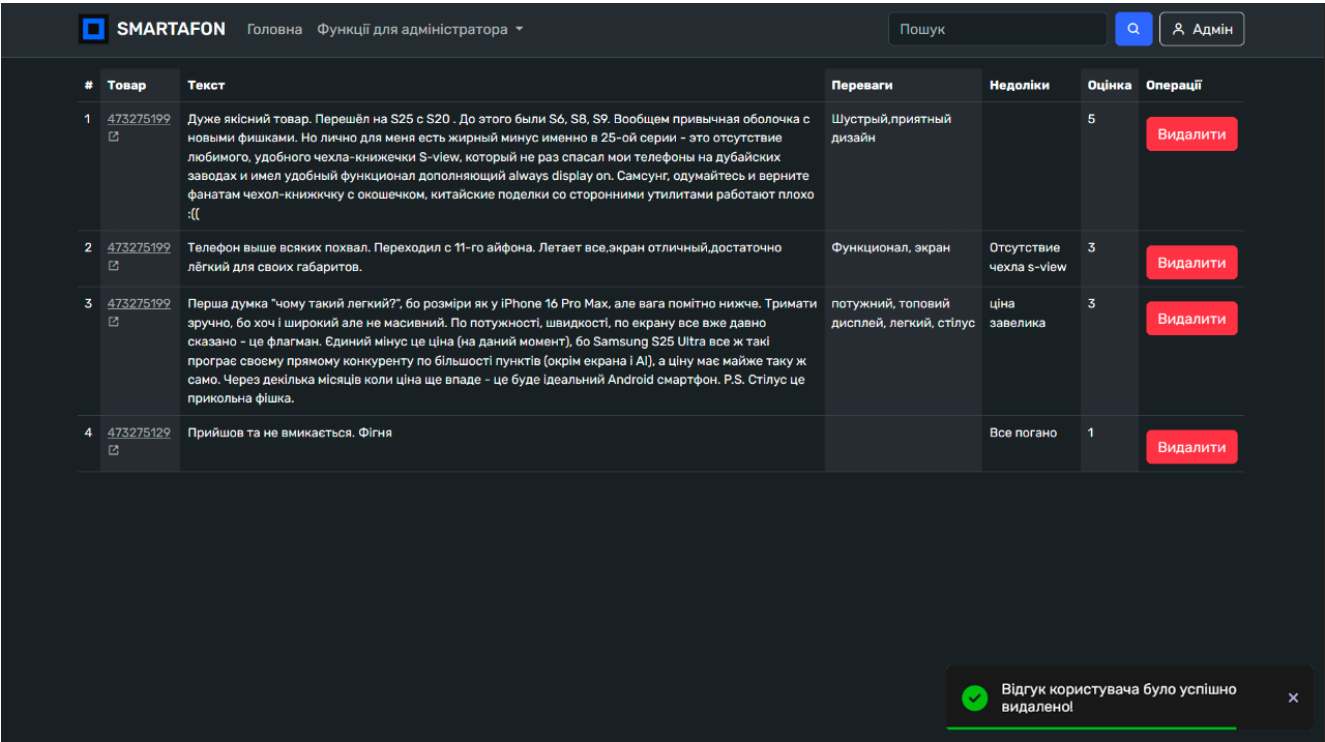


Рисунок 2.19 – Результат редагування інформації вже існуючого смартфона

Проведене тестування та випробування вебзастосунку для інтернет-магазину смартфонів «Smartafon» дало змогу перевірити функціональність, стабільність роботи, відповідність технічним вимогам, а також усунути можливі помилки.

3 РЕЗУЛЬТАТИ РОЗРОБКИ ВЕБЗАСТОСУНКУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ СМАРТФОНІВ «SMARTAFON»

В результаті виконання кваліфікаційної роботи було розроблено вебзастосунок для інтернет-магазину смартфонів «Smartafon». Було виконано аналіз практичної задачі, яку потрібно було виконати, пошук та аналіз існуючих аналогів, а також обґрунтування доцільності створення власного вебзастосунку.

У результаті аналізу предметної галузі сформульовано постановку задачі для розробки вебзастосунку, визначено основні функції вебзастосунку, користувачів, проведено аналіз вимог, обрано архітектуру та технології розробки, що найкраще відповідають вимогам вебзастосунка.

На етапі ескізного проєктування було розроблено діаграми діяльності для варіантів використання, концептуальну модель та макети інтерфейсу користувача.

В межах технічного проєктування було розроблено статичну модель, що представлена у вигляді діаграми класів, динамічну модель, що представлена у вигляді діаграм послідовності та логічну модель даних вебзастосунку.

На етапі робочого проєктування було визначено, обґрунтовано та затверджено засоби розробки вебзастосунку, розроблено та представлено програмний код вебзастосунку, а також проведено тестування та випробування вебзастосунку згідно з програми та методики випробувань, що наведені у додатку В.

Останнім етапом стала розробка інструкції користувача з візуальним представленням функціональних можливостей вебзастосунку.

Технічне завдання наведено у додатку А, опис вебзастосунку у додатку Б, програму та методику випробувань у додатку В, інструкцію користувача у додатку Г, а текст програми у додатку Д.

Загальна кількість написаного коду становить 4864 рядки, а розмір проєкту 819 КБ (без урахування встановлених залежностей). Повний розмір директорії проєкту (з встановленими залежностями) становить 144 МБ.

4 ОХОРОНА ПРАЦІ

Вступ

Охорона праці – це комплекс засобів, що направлені на збереження здоров'я і працездатності людини в процесі праці.

Порушення законодавства та інших нормативно-правових актів з охорони праці, перешкоджання діяльності посадових осіб органів державного нагляду за охороною праці, а також представників профспілок та їх об'єднань тягне за собою дисциплінарну, адміністративну, матеріальну або кримінальну відповідальність відповідно до закону (стаття 44 Закону «Про охорону праці»).

4.1 Основні законодавчі і нормативні документи, які регламентують охорону праці в Україні

Охорона праці представляє собою систему правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів, що спрямовані на збереження здоров'я та працездатності людини в процесі трудової діяльності.

Відповідно до статті 44 Закону «Про охорону праці», за порушення законодавчих та нормативно-правових актів, що регулюють охорону праці, а також за створення перешкод у діяльності органів державного нагляду та представників профспілок, винні особи можуть бути притягнені до дисциплінарної, адміністративної, матеріальної або кримінальної відповідальності відповідно до закону.

Базовим документом законодавства України у сфері охорони праці є Конституція України, яка закріплює право кожного громадянина на безпечні та нешкідливі умови праці. Окрім того, діє комплекс нормативно-правових актів, що спрямовані на забезпечення реалізації цього конституційного права. Основні

законодавчі та нормативні акти з охорони праці включають:

- Закон України «Про охорону праці» (від 14.10.1992 № 2694-XII) [5] – основний закон, який визначає загальні принципи охорони праці, права і обов'язки сторін трудового договору, основи державного нагляду та контролю;
- Кодекс законів про працю України (від 10.12.1971 № 322-VIII) [6] – містить положення, що стосуються трудових відносин і охорони праці в контексті трудового законодавства;
- Закон України «Про основи соціальної захищеності інвалідів в Україні» (від 21.03.1991 № 875-XII) [7] – визначає вимоги до створення безбар'єрного середовища для осіб з обмеженими можливостями;
- Закон України «Про загальнообов'язкове державне соціальне страхування» (від 1999 №46-47) [8] – визначає правила, права та обов'язки для забезпечення соціального захисту працівників.

4.2 Структура управління питаннями охорони праці на підприємстві

Структура управління питаннями охорони праці на підприємстві визначає, як організовано управління і контроль за дотриманням норм і вимог охорони праці. Вона включає в себе кілька ключових елементів, які забезпечують ефективне управління безпекою на робочих місцях. Основними компонентами структури управління охороною праці на підприємстві є керівництво підприємства, комісія з охорони праці, спеціалісти з охорони праці та працівники підприємства.

4.3 Забезпечення техніки безпеки та охорона праці для конкретних посад або при виконанні конкретних завдань. Розробка заходів по зменшенню дії шкідливих факторів для конкретних посад або при виконанні конкретних завдань

Забезпечення техніки безпеки та охорони праці є однією з основних складових частин управління виробничими процесами на підприємствах будь-якої галузі. Підвищення рівня безпеки на виробництві, а також мінімізація ризиків,

пов'язаних із виконанням конкретних завдань і функцій, є ключовими завданнями для підтримки здоров'я працівників та забезпечення ефективності виробничої діяльності.

Для кожної конкретної посади або завдання необхідно провести ідентифікацію потенційних небезпек, які можуть виникнути в процесі роботи. Це є першим етапом у процесі забезпечення техніки безпеки та охорони праці. Ідентифікація небезпек включає виявлення фізичних, хімічних, біологічних, ергономічних та психосоціальних факторів, які можуть загрожувати здоров'ю та безпеці працівників. Наприклад, до фізичних небезпек можна віднести болі в шії та спині через неправильну організацію робочого місця, або ожиріння, через малорухливість на робочому місці. Пожежа є хімічною небезпекою, яка може виникнути через несправність електрообладнання, коротке замикання, перегрів пристроїв, неправильне зберігання легкозаймистих матеріалів або нехтування правилами пожежної безпеки. Психосоціальними небезпеками є стрес на роботі, або конфлікти в колективі, або надмірний контроль з боку керівництва. Ці небезпеки можуть негативно позначитися на продуктивності, здоров'ї та загальному емоційному стані працівників. Тривалий вплив психосоціальних ризиків може спричинити розвиток хронічного стресу, депресії, тривожних розладів та навіть фізичних захворювань.

Оцінка ризиків є наступним етапом після ідентифікації небезпек і включає визначення ймовірності і серйозності наслідків для здоров'я та безпеки працівників. Оцінка ризиків допомагає встановити пріоритет для ризиків за їхньою важливістю і розробити відповідні заходи для їх усунення або зменшення.

Одним з ключових аспектів забезпечення безпеки є розробка чітких інструкцій та процедур для конкретних посад або завдань. Інструкції повинні містити інформацію про безпечні методи роботи, використання обладнання та дії у разі виникнення аварійних ситуацій. Ці інструкції повинні бути на місцях праці в декількох копіях, а також представлені працівникам до ознайомлення. Також повинні бути створені організаційні заходи, які включають в себе навчання, контроль за дотриманням норм безпеки та впровадження системи моніторингу безпеки.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було вирішено практичну задачу інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розроблено вебзастосунок для інтернет-магазину смартфонів «Smartafon».

Для досягнення мети було вирішено такі завдання:

- проведено аналіз практичної задачі інженерії програмного забезпечення;
- проведено аналіз існуючих аналогів та обґрунтовано необхідність розробки власного вебзастосунку;
- розроблено постановку задачі та технічне завдання;
- виконано аналіз вимог;
- розроблено архітектуру вебзастосунку;
- розроблено вебзастосунок для інтернет-магазину смартфонів «Smartafon»;
- проведено тестування та розроблено документацію до розробленого вебзастосунку;
- розроблено розділ охорони праці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Світ смартфонів. URL: <https://world-smartphones.com.ua> (дата звернення 25.03.2025).
2. TOUCH. URL: <https://touch.com.ua> (дата звернення 25.03.2025).
3. Smartfon. URL: <https://smartfon.com.ua/> (дата звернення 25.03.2025).
4. Діаграма послідовності (Sequence Diagrams). URL: <https://www.maxzosim.com/sequence-diagrams/> (дата звернення 03.05.2025).
5. Про охорону праці. Закон України від 14.10.1992 № 2694-XII. Дата оновлення 25.04.2024. URL: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення 12.03.2025).
6. Про працю України. Кодекс законів від 10.12.1971 № 322-VIII. Дата оновлення 20.06.2024. URL: <https://zakon.rada.gov.ua/laws/show/322-08> (дата звернення 12.03.2025).
7. Про основи соціальної захищеності інвалідів в Україні. Закон України від 21.03.1991 № 875-XII. Дата оновлення 09.11.2023. URL: <https://zakon.rada.gov.ua/laws/show/875-12> (дата звернення 13.03.2025).
8. Про загальнообов'язкове державне соціальне страхування. Закон України від 1999 №46-47. Дата оновлення 25.04.2024. URL: <https://zakon.rada.gov.ua/laws/show/1105-14> (дата звернення 13.03.2025).

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

Вступ

Повна назва розроблюваного проєкту: «Вебзастосунок для інтернет-магазину смартфонів «Smartafon»».

Галузь застосування: торгівля смартфонами, аналітика та звітність.

1 Підстави для розробки

Підставою для розробки вебзастосунку є завдання на кваліфікаційну роботу на здобуття ступеня вищої освіти «бакалавр».

2 Призначення розробки

2.1 Функціональне призначення вебзастосунку

Функціональне призначення вебзастосунку полягає в наданні можливості переглядати широкий вибір смартфонів різних брендів і моделей з детальними описами, характеристиками і фотографіями, що дозволяє користувачам легко знайти бажаний смартфон. Крім того, вебзастосунок надаватиме можливість використовувати фільтри і сортування для швидкого пошуку за різними параметрами, переглядати огляди та рецензії, оформлювати замовлення, а також створювати особистий кабінет для зручності користувачів.

2.2 Експлуатаційне призначення вебзастосунку

Експлуатаційним призначенням вебзастосунку для інтернет-магазину смартфонів є полегшення пошуку смартфонів, надання зручного вибору та спрощення процесу придбання смартфонів. Також, експлуатаційним призначенням вебзастосунку є полегшення управління смартфонами та операціями магазину для адміністраторів.

3 Вимоги до програмного забезпечення

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

Функції вебзастосунку для користувача:

- реєстрація (Вхідні дані: Прізвище та ім'я користувача; електронна пошта користувача; пароль користувача; номер телефону користувача. Результат: створення облікового запису користувача, який дозволяє доступ до персоналізованих функцій вебзастосунку та відображення повідомлення про створення облікового запису);
- авторизація (Вхідні дані: адреса електронної пошти користувача та пароль користувача; або тільки номер телефону користувача. Результат: доступ до особистого кабінету користувача з можливістю перегляду історії замовлень, редагування профілю тощо; відображення повідомлення про вхід в обліковий запис);
- відображення списку смартфонів (Вхідні дані: натиснення кнопки «Каталог». Результат: відображення списку смартфонів з неповною інформацією про смартфон, а саме назва смартфона, зображення смартфона, опис смартфона, ціна смартфона, оцінка смартфона, що дозволяє користувачеві ознайомитися з асортиментом);
- відображення інформації про смартфон (Вхідні дані: ID смартфона. Результат: відображення всієї інформації про смартфон, що дозволяє користувачеві ознайомитися з смартфоном);
- додавання відгуку (Вхідні дані: смартфон; текст відгуку; текст розділу «Переваги»; текст розділу «Недоліки»; рейтинг; дата та час створення відгуку; ID користувача. Результат: додавання відгуку до смартфона);
- відображення списку відгуків (Вхідні дані: ID смартфона. Результат: відображення списку відгуків, написаних іншими користувачами (ID відгуку; смартфон; текст розділу «Переваги»; текст розділу «Недоліки»; текст відгуку; користувач, який написав; дата написання));

- видалення відгуку (Вхідні дані: натиснення кнопки «Видалити»; ID відгуку. Результат: видалення відгуку до смартфона);
- створення замовлення смартфона (Вхідні дані: смартфон; кількість смартфонів; адреса доставки; вартість замовлення; дата та час створення замовлення; спосіб оплати; клієнт. Результат: формування замовлення з підтвердженням та деталями, яке обробляється програмним забезпеченням);
- відображення списку замовлень (Вхідні дані: натиснення кнопки «Мої замовлення». Результат: відображення списку замовлень з такими даними: ID замовлення; назва смартфона; кількість смартфонів; дата та час створення; спосіб доставки; адреса доставки; одержувач замовлення; вартість замовлення; спосіб оплати);
- скасування замовлень (Вхідні дані: ID замовлення. Результат: скасування замовлення, створеного користувачем).

Функції вебзастосунку для адміністратора:

- авторизація (Вхідні дані: адреса електронної пошти адміністратора; пароль адміністратора. Результат: доступ до адмін-панелі для управління вебзастосунком; відображення повідомлення про вхід в обліковий запис);
- додавання, редагування смартфона (Вхідні дані: натиснення посилання «Додати смартфон»/«Редагувати смартфон»; ID смартфона; артикул; код моделі; назва; колір; процесор; кількість оперативної пам'яті; кількість основної пам'яті; операційна система; камера; фронтальна камера; виробник; роздільна якість екрана; діагональ екрана; ємність батареї; ціна; акційна ціна; кількість слотів для SIM-карт; стан; зображення, тип слотів для SIM-карт. Результат: додавання або редагування даних смартфона ; відображення повідомлення про статус виконання дії);
- видалення смартфона (Вхідні дані: ID смартфона. Результат: видалення даних смартфона. Відображення повідомлення про статус виконання дії);
- видалення відгуку (Вхідні дані: натиснення кнопки «Видалити відгук»; ID відгуку. Результат: видалення відгуку для забезпечення якості та безпечності);

- запит на звіт продажів (Вхідні дані: натиснення кнопки «Звіт продажів за тиждень/місяць/квартал». Результат: звіт продажів (загальна кількість замовлень; кількість виконаних замовлень; кількість скасованих замовлень; виручка; дата початку звіту; дата кінця звіту));
- відображення списку замовлень (Вхідні дані: натиснення посилання «Замовлення». Результат: відображення списку замовлень з такими даними: ID замовлення; назва смартфонів; кількість смартфонів; дата та час створення; спосіб доставки; адреса доставки; одержувач замовлення; вартість замовлення; спосіб оплати);
- відображення списку відгуків (Вхідні дані: натиснення посилання «Відгуки»; ID смартфона. Результат: відображення списку відгуків написаних іншими користувачами (ID відгуку; смартфон; оцінка; текст розділу «Переваги»; текст розділу «Недоліки»; користувач, який написав; дата написання));
- відображення списку смартфонів (Вхідні дані: натиснення посилання «Смартфон». Результат: відображення списку смартфонів з інформацією про смартфон, а саме назва смартфона, зображення смартфона, характеристики смартфона, ціна смартфона, оцінка смартфона).

3.1.2 Вимоги до організації вхідних і вихідних даних

Вхідні дані – це інформація, що вводиться вручну (залежно від типу поля введення). Дані можуть бути: числовими (наприклад, ціни), текстовими (наприклад, назви моделей), або датами. Також можуть бути вибічковими списками (наприклад, список брендів смартфонів, тип операційної системи, тощо), які полегшують процес введення.

Вихідною інформацією є відображення результатів операцій у формі або повідомлення, а також сформовані звіти. Вихідними даними для звітів буде файл Excel-таблиць, в якому буде міститись вся потрібна інформація.

3.2 Вимоги до надійності

Вебзастосунок інтернет-магазину «Smartafon» буде доступним 24/7. Клієнти

та адміністратори матимуть змогу відкривати вебзастосунок в будь-який момент. Адміністраторам потрібна можливість працювати з смартфонами у будь-який момент часу для підтримки правильного функціоналу сайту, актуалізації інформації на сайті, тощо.

Система буде автоматично, кожні 24 годин робити копію файлів вебзастосунків та копію бази даних, щоб у випадку виникнення помилки, адміністратор міг відновити систему.

Також, для уникнення помилок, система буде перевіряти дані на відповідність формату та/або обмеженням при їх додаванні, або редагуванні.

3.3 Вимоги до складу й параметрів технічних засобів

Система повинна задовольняти вказаним мінімальним вимогам на сервері, а саме:

- процесор: 4 ядра 1.8 ГГц;
- оперативна пам'ять: 8 ГБ;
- пам'ять: 100 ГБ вільного місця на диску.

Користувачський комп'ютер повинен задовольняти таким мінімальним вимогам:

- процесор: 2 ядра 1.6 ГГц;
- оперативна пам'ять: 2 ГБ;
- доступ до інтернету.

3.4 Вимоги до інформаційної й програмної сумісності

Мінімальні вимоги інформаційної й програмної сумісності сервера:

- ОС: Windows 10 або Linux Ubuntu 20.10;
- PHP 8.2;
- phpMyAdmin 5.2;
- Система керування базами даних MySQL 8.0.

Мінімальні вимоги інформаційної й програмної сумісності комп'ютера

користувача:

- ОС: Windows 7 або Linux Ubuntu 10;
- Браузер: Google Chrome 79, Mozilla Firefox 71, Microsoft Edge 79, Opera 65 або Safari 13.

4 Вимоги до програмної документації

У програмну документацію повинні входити:

- технічне завдання;
- керівництво користувача;
- програма та методика випробувань;
- опис програмного забезпечення;
- код програми.

5 Техніко-економічні показники

Техніко-економічні показники не розраховувались.

6 Стадії та етапи розробки

Стадії та етапи розробки ПЗ представлені у таблиці А.1.

Таблиця А.1 – Стадії та етапи розробки ПЗ

№	Стадії та етапи	Термін виконання
1	Аналіз предметної області	12.02.2025
2	Аналіз вимог до ПЗ	20.02.2025
3	Розробка архітектури ПЗ	10.03.2025
4	Розробка проєкту ПЗ	15.04.2025
5	Реалізація та тестування ПЗ	24.05.2025

Представлені стадії та етапи розробки ПЗ дозволяють відстежити послідовність дій та забезпечити контроль за дотриманням встановлених термінів.

7 Порядок контролю й приймання

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

Приймання проводиться відповідно до програми і методики випробувань і документується за допомогою протоколу проведення випробувань. У разі знаходження помилок під час приймання програмного виробу складається акт про знайдені помилки, який підписується представниками замовника і розробника і затверджується керівниками організації замовника та організації розробника. Розробник повинен протягом не більше ніж двох тижнів виправити зазначені помилки і оповістити замовника про повторне проведення перевірки.

ДОДАТОК Б – ОПИС ВЕБЗАСТОСУНКУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ СМАРТФОНІВ «SMARTAFON»

1 Загальні відомості

1.1 Позначення і найменування програми

Найменування: Вебзастосунок для інтернет-магазину смартфонів «Smartafon».

Позначення: Вебзастосунок.

1.2 Програмне забезпечення, необхідне для функціонування програми

Сервер має відповідати таким мінімальним вимогам: операційна система не нижче Windows 10 або Linux Ubuntu 20.10; встановлене програмне забезпечення PHP версії 8.2, phpMyAdmin не нижче версії 5.2, а також MySQL версії 8.0.

Комп'ютер користувача повинен мати операційну систему Windows 7, Linux Ubuntu 10, MacOS X 10.7 чи новіші, а також один із підтримуваних браузерів: Google Chrome 79, Mozilla Firefox 71, Microsoft Edge 79, Opera 65, Safari 13, або новіші версії цих браузерів.

1.3 Мови програмування

Серверну частину вебзастосунку інтернет-магазину смартфонів «Smartafon» було реалізовано з використанням мови програмування PHP та фреймворку Laravel. Для зберігання даних було використано базу даних MySQL.

Клієнтську частину вебзастосунку створено за допомогою бібліотеки jQuery та фреймворку Bootstrap для забезпечення інтерактивності та адаптивного інтерфейсу.

2 Функціональне призначення

Функціональне призначення вебзастосунку полягає в наданні можливості переглядати широкий вибір смартфонів різних брендів і моделей з детальними описами, характеристиками і фотографіями, що дозволяє користувачам легко знайти бажаний смартфон. Крім того, вебзастосунок надаватиме можливість використовувати фільтри і сортування для швидкого пошуку за різними параметрами, переглядати огляди та рецензії, оформлювати замовлення, а також створювати особистий кабінет для зручності користувачів.

Вебзастосунок інтернет-магазину смартфонів «Smartafon» має лише одне функціональне обмеження – потреба в підключенні до мережі Інтернет.

3 Опис логічної структури

3.1 Алгоритми вебзастосунку

Вебзастосунок побудований за клієнт-серверною архітектурою, де клієнтською частиною виступає веббраузер користувача, а серверною – один або кілька комп'ютерів, на яких розгорнуто сам вебзастосунок, а також додаткові сервіси, необхідні для його роботи, зокрема система управління базами даних MySQL.

Послідовність взаємодії між клієнтом і сервером відбувається наступним чином:

- 1) Користувач, взаємодіючи із вебзастосунком (клієнтською частиною), ініціює надсилання запитів до сервера;
- 2) Браузер надсилає запит до сервера;
- 3) Сервер приймає запит і передає його серверному застосунку;
- 4) Серверна частина обробляє запит, викликає відповідний контролер та за потреби звертається до бази даних;
- 5) Сервер формує відповідь та надсилає її назад клієнту;
- 6) Браузер отримує відповідь і відображає її користувачу.

3.2 Методи, що використовуються

Вебзастосунок інтернет-магазину смартфонів «Smartafon» було розроблено використовуючи мову програмування PHP, яка повністю підтримує об'єктно-орієнтоване програмування. Для розробки було використано фреймворк Laravel, що базується на ООП-підходах і реалізує архітектурний шаблон MVC (Model View Controller), через що було забезпечено чітке розділення обов'язків між логікою обробки даних, інтерфейсом користувача та бізнес-логікою.

3.3 Структура програми

Архітектурною структурою вебзастосунку інтернет-магазину смартфонів «Smartafon» є «Клієнт-Сервер» та архітектурний шаблон проектування MVC (Model View Controller), де застосунок ділить обов'язки між трьома компонентами:

- Model (Модель): Відповідає за бізнес-логіку та управління даними, отримує, зберігає, обробляє дані, а також взаємодіє з базами даних;
- View (Подання/Вид/Представлення): Відповідає за відображення інформації користувачеві. Отримує дані від моделі і відображає їх у зручному вигляді;
- Controller (Контролер): Відповідає за отримання вводу від користувача (наприклад, через, форми чи команди). Обробляє отриманні дані, передає їх до моделі і визначає, яке подання/представлення має бути оновлене.

4 Необхідні технічні засоби

Система повинна задовольняти вказаним мінімальним вимогам на сервері, а саме:

- процесор: 4 ядра 1.8 ГГц;
- оперативна пам'ять: 8 ГБ;
- пам'ять: 100 ГБ вільного місця на диску.

Користувацький комп'ютер повинен задовольняти таким мінімальним вимогам:

- процесор: 2 ядра 1.6 ГГц;

- оперативна пам'ять: 2 ГБ;
- доступ до інтернету.

5 Виклик і завантаження

Вебзастосунок інтернет-магазину «Smartafon» розгортається на хостинговому сервері, що підтримує мову PHP та систему керування базами даних MySQL. У результаті правильного розгортання вебзастосунок стає доступним через веббраузер за допомогою доменного імені, пов'язаного з хостингом.

6 Вхідні дані

Вхідні дані – це інформація, що вводиться вручну (залежно від типу поля введення). Дані можуть бути: числовими (наприклад, ціни), текстовими (наприклад, назви моделей), або датами. Також можуть бути вибілковими списками (наприклад, список брендів смартфонів, тип операційної системи, тощо), які полегшують процес введення.

7 Вихідні дані

Вихідною інформацією є відображення результатів операцій у формі або повідомлення, а також сформовані звіти. Звіти формуються у вигляді файлів Excel.

ДОДАТОК В – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ВЕБЗАСТОСУНКУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ СМАРТФОНІВ «SMARTAFON»

1 Об'єкт випробувань

Об'єктом випробувань є вебзастосунок для інтернет-магазину смартфонів «Smartafon», що розроблено в рамках завдання кваліфікаційної роботи.

Областю застосування вебзастосунку є продаж смартфонів через інтернет, забезпечуючи користувачам зручний пошук смартфонів, перегляд смартфонів та оформлення замовлень. Також забезпечує керування смартфонами, замовленнями та звітність для адміністратора.

2 Мета випробувань

Метою випробувань вебзастосунку для інтернет-магазину смартфонів є перевірка відповідності реалізованих програмних рішень функціональним і нефункціональним вимогам, встановленим у технічному завданні, пошук допущених помилок, виявлення потенційних вразливостей, а також підтвердження стабільної роботи в умовах, що максимально наближені до реальних.

3 Вимоги до програми

Під час випробування вебзастосунків повинен відповідати вимогам, що були встановлені в розділі «Вимоги до функціональних характеристик» технічного завдання.

4 Вимоги до програмної документації

Програмна документація вебзастосунку для інтернет-магазину смартфонів «Smartafon» повинна складатись з:

- Технічного завдання;
- Опису вебзастосунку;

- Програми та методики випробувань;
- Інструкції користувача;
- Тексту вебзастосунку.

5 Засоби та порядок випробувань

Засоби випробувань представлені в таблиці В.1.

Таблиця В.1 – Засоби випробувань

Апаратні засоби випробувань для серверу	
Процесор	4 ядра 1.8 ГГц
Оперативна пам'ять	8 ГБ
Пам'ять	100 ГБ вільного місця на диску
Апаратні засоби випробувань для клієнта	
Процесор	2 ядра 1.6 ГГц
Оперативна пам'ять	2 ГБ
Доступ до інтернету	Доступ до інтернету
Програмні засоби випробувань для серверу	
ОС	Windows 10 або Linux Ubuntu 20.10
PHP	Версія 8.2
phpMyAdmin	Версія 5.2
Система керування базами даних	MySQL 8.0
Програмні засоби випробувань для клієнта	
ОС	Windows 7 або Linux Ubuntu 10
Браузер	Google Chrome 79 / Mozilla Firefox 71 / Microsoft Edge 79 / Opera 65 / Safari 13

Порядок проведення випробувань:

- 1) Випробування всіх функцій вебзастосунку;
- 2) Фіксація результатів випробувань та подальший їх аналіз;
- 3) Пошук проблем, помилок, зауважень;

При знаходженні проблем, або помилок

- 4) Виділення часу для виправлення помилок та саме виправлення;

5) Повторне випробування.

6 Методика випробувань

Для оцінювання та випробувань застосовуються методи «чорної скриньки». Для функціонального тестування використовуються принципи еквівалентного розбиття, граничних значень та тестування станів.

В таблиці В.2 представлено очікувані результати випробувань для деяких функцій.

Таблиця В.2 – Очікувані результати випробувань для деяких функцій

Дія	Очікуваний результат
Реєстрація з правильно введеними даними.	Успішна реєстрація нового облікового запису.
Авторизація з правильно введеними даними.	Успішна авторизація в обліковий запис. Перенаправлення користувача на головну сторінку.
Реєстрація облікового запис на номер телефону, або пошту, що вже зареєстровані.	Відображення помилки з повідомленням про те, що такий обліковий запис вже існує.
Авторизація до не існуючого облікового запису.	Відображення помилки з повідомленням про те, що такого облікового запису не існує
Прокручування сторінки з списком смартфонів, кількість яких більше ніж 12.	Динамічне завантаження нових смартфонів.
Відкриття сторінки смартфона, який не містить зображень.	Відображення зображення «заглушки» для цього смартфона.
Відкриття форми написання відгуку, без авторизації в обліковий запис.	Відображення повідомлення про необхідність авторизації, для написання відгуку.

Наведені випробування вебзастосунку для інтернет-магазину смартфонів «Smartafon» нададуть можливість перевірити функціональність, стабільність роботи, відповідність технічним вимогам, а також виявити та усунути можливі помилки в роботі.

ДОДАТОК Г – ІНСТРУКЦІЯ КОРИСТУВАЧА

1 Призначення програми

1.1 Функціональне призначення

Функціональне призначення вебзастосунку полягає в наданні можливості переглядати широкий вибір смартфонів різних брендів і моделей з детальними описами, характеристиками і фотографіями, що дозволяє користувачам легко знайти бажаний смартфон. Крім того, вебзастосунок надаватиме можливість використовувати фільтри і сортування для швидкого пошуку за різними параметрами, переглядати огляди та рецензії, оформлювати замовлення, а також створювати особистий кабінет для зручності користувачів

1.2 Експлуатаційне призначення

Експлуатаційним призначенням вебзастосунку для інтернет-магазину смартфонів є полегшення пошуку смартфонів, надання зручного вибору та спрощення процесу придбання смартфонів. Також, експлуатаційним призначенням вебзастосунку є полегшення управління смартфонами та операціями магазину для адміністраторів

1.3 Склад функцій

Нижче наведено склад функцій, які повинен надавати вебзастосунок для інтернет-магазину смартфонів «Smartafon».

Функції вебзастосунку для користувача:

- реєстрація (Вхідні дані: Прізвище та ім'я користувача; електронна пошта користувача; пароль користувача; номер телефону користувача. Результат: створення облікового запису користувача, який дозволяє доступ до персоналізованих функцій вебзастосунку та відображення повідомлення про створення облікового запису);

- авторизація (Вхідні дані: адреса електронної пошти користувача та пароль користувача; або тільки номер телефону користувача. Результат: доступ до особистого кабінету користувача з можливістю перегляду історії замовлень, редагування профілю тощо; відображення повідомлення про вхід в обліковий запис);
- відображення списку смартфонів (Вхідні дані: натиснення кнопки «Каталог». Результат: відображення списку смартфонів з неповною інформацією про смартфон, а саме назва смартфона, зображення смартфона, опис смартфона, ціна смартфона, оцінка смартфона, що дозволяє користувачеві ознайомитися з асортиментом);
- відображення інформації про смартфон (Вхідні дані: ID смартфона. Результат: відображення всієї інформації про смартфон, що дозволяє користувачеві ознайомитися з смартфоном);
- додавання відгуку (Вхідні дані: смартфон; текст відгуку; текст розділу «Переваги»; текст розділу «Недоліки»; рейтинг; дата та час створення відгуку; ID користувача. Результат: додавання відгуку до смартфона);
- відображення списку відгуків (Вхідні дані: ID смартфона. Результат: відображення списку відгуків, написаних іншими користувачами (ID відгуку; смартфон; текст розділу «Переваги»; текст розділу «Недоліки»; текст відгуку; користувач, який написав; дата написання));
- видалення відгуку (Вхідні дані: натиснення кнопки «Видалити»; ID відгуку. Результат: видалення відгуку до смартфона);
- створення замовлення смартфона (Вхідні дані: смартфон; кількість смартфонів; адреса доставки; вартість замовлення; дата та час створення замовлення; спосіб оплати; клієнт. Результат: формування замовлення з підтвердженням та деталями, яке обробляється програмним забезпеченням);
- відображення списку замовлень (Вхідні дані: натиснення кнопки «Мої замовлення». Результат: відображення списку замовлень з такими даними: ID замовлення; назва смартфона; кількість смартфонів; дата та час створення; спосіб

доставки; адреса доставки; одержувач замовлення; вартість замовлення; спосіб оплати);

- скасування замовлень (Вхідні дані: ID замовлення. Результат: скасування замовлення, створеного користувачем).

Функції вебзастосунку для адміністратора:

- авторизація (Вхідні дані: адреса електронної пошти адміністратора; пароль адміністратора. Результат: доступ до адмін-панелі для управління вебзастосунком; відображення повідомлення про вхід в обліковий запис);

- додавання, редагування смартфона (Вхідні дані: натиснення посилання «Додати смартфон»/«Редагувати смартфон»; ID смартфона; артикул; код моделі; назва; колір; процесор; кількість оперативної пам'яті; кількість основної пам'яті; операційна система; камера; фронтальна камера; виробник; роздільна якість екрана; діагональ екрана; ємність батареї; ціна; акційна ціна; кількість слотів для SIM-карт; стан; зображення, тип слотів для SIM-карт. Результат: додавання або редагування даних смартфона ; відображення повідомлення про статус виконання дії);

- видалення смартфона (Вхідні дані: ID смартфона. Результат: видалення даних смартфона. Відображення повідомлення про статус виконання дії);

- видалення відгуку (Вхідні дані: натиснення кнопки «Видалити відгук»; ID відгуку. Результат: видалення відгуку для забезпечення якості та безпечності);

- запит на звіт продажів (Вхідні дані: натиснення кнопки «Звіт продажів за тиждень/місяць/квартал». Результат: звіт продажів (загальна кількість замовлень; кількість виконаних замовлень; кількість скасованих замовлень; виручка; дата початку звіту; дата кінця звіту));

- відображення списку замовлень (Вхідні дані: натиснення посилання «Замовлення». Результат: відображення списку замовлень з такими даними: ID замовлення; назва смартфонів; кількість смартфонів; дата та час створення; спосіб

доставки; адреса доставки; одержувач замовлення; вартість замовлення; спосіб оплати);

- відображення списку відгуків (Вхідні дані: натиснення посилання «Відгуки»; ID смартфона. Результат: відображення списку відгуків написаних іншими користувачами (ID відгуку; смартфон; оцінка; текст розділу «Переваги»; текст розділу «Недоліки»; користувач, який написав; дата написання));

- відображення списку смартфонів (Вхідні дані: натиснення посилання «Смартфон». Результат: відображення списку смартфонів з інформацією про смартфон, а саме назва смартфона, зображення смартфона, характеристики смартфона, ціна смартфона, оцінка смартфона).

2 Умови виконання програми

2.1 Мінімальний склад апаратних засобів

Система повинна задовольняти вказаним мінімальним вимогам на сервері, а саме:

- процесор: 4 ядра 1.8 ГГц;
- оперативна пам'ять: 8 ГБ;
- пам'ять: 100 ГБ вільного місця на диску.

Користувацький комп'ютер повинен задовольняти таким мінімальним вимогам:

- процесор: 2 ядра 1.6 ГГц;
- оперативна пам'ять: 2 ГБ;
- доступ до інтернету.

2.2 Мінімальний склад програмних засобів

Мінімальні вимоги інформаційної й програмної сумісності сервера:

- ОС: Windows 10 або Linux Ubuntu 20.10;
- PHP 8.2;
- phpMyAdmin 5.2;

- Система керування базами даних MySQL 8.0.

Мінімальні вимоги інформаційної й програмної сумісності комп'ютера користувача:

- ОС: Windows 7 або Linux Ubuntu 10;
- Браузер: Google Chrome 79, Mozilla Firefox 71, Microsoft Edge 79, Opera 65 або Safari 13.

2.3 Вимоги до персоналу (користувачів)

Користувачі вебзастосунку для інтернет-магазину смартфонів «Smartafon» повинні мати базові навички користування комп'ютером та веббраузером. Також Користувачі вебзастосунку повинні інтуїтивно розуміти інтерфейси схожих вебзастосунків для більшого розуміння та ефектної роботи з інтерфейсом вебзастосунку для інтернет-магазину смартфонів «Smartafon».

3 Виконання програми

3.1 Завантаження та запуск програми

Для відкриття вебзастосунку для інтернет-магазину смартфонів «Smartafon» користувач повинен відкрити веббраузер та перейти за доменним ім'ям, що прив'язано до вебсерверу, на якому розгорнуто вебзастосунок. Після відкриття вебзастосунку, користувач опиняється на головній сторінці. Головну сторінку вебзастосунку та сторінку з детальним описом смартфона представлено на рисунках Г.1 та Г.2.

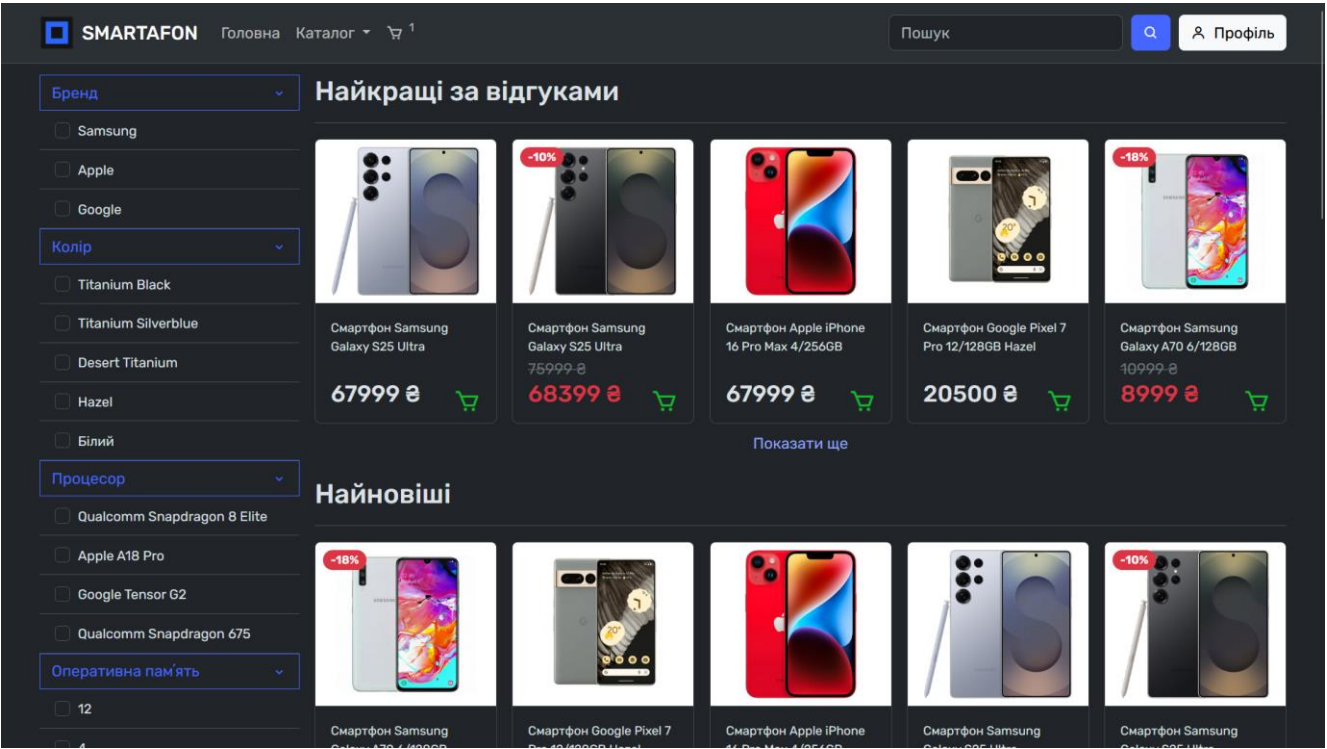


Рисунок Г.1 – Головна сторінка вебзастосунку для інтернет-магазину «Smartafon»

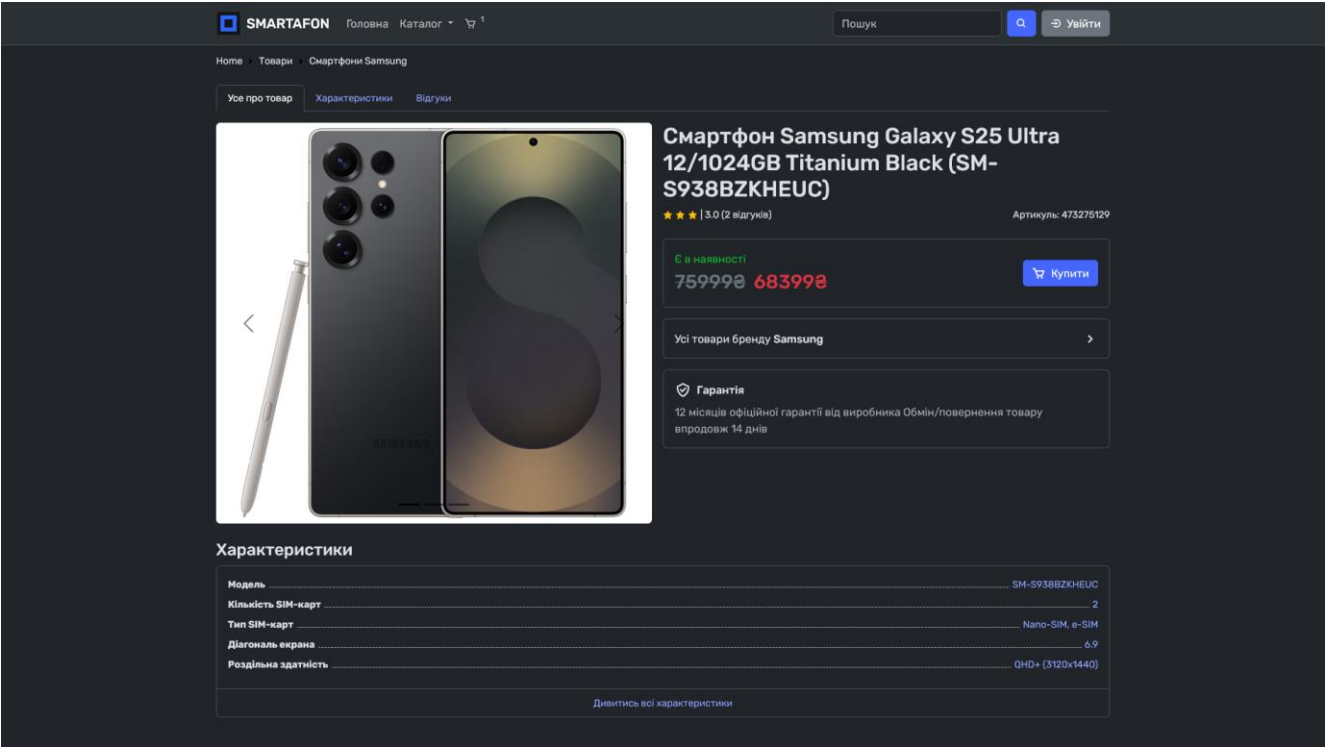


Рисунок Г.2 – Сторінка вебзастосунку для інтернет-магазину «Smartafon» з детальним описом смартфона

Описаних дій достатньо для того, щоб успішно розпочати роботу із вебзастосунком для інтернет-магазину смартфонів «Smartafon».

3.2 Виконання програми

На рисунку Г.3 виділена основна частина елементів вебзастосунку з номерами для подальшого пояснення їх функціоналу.

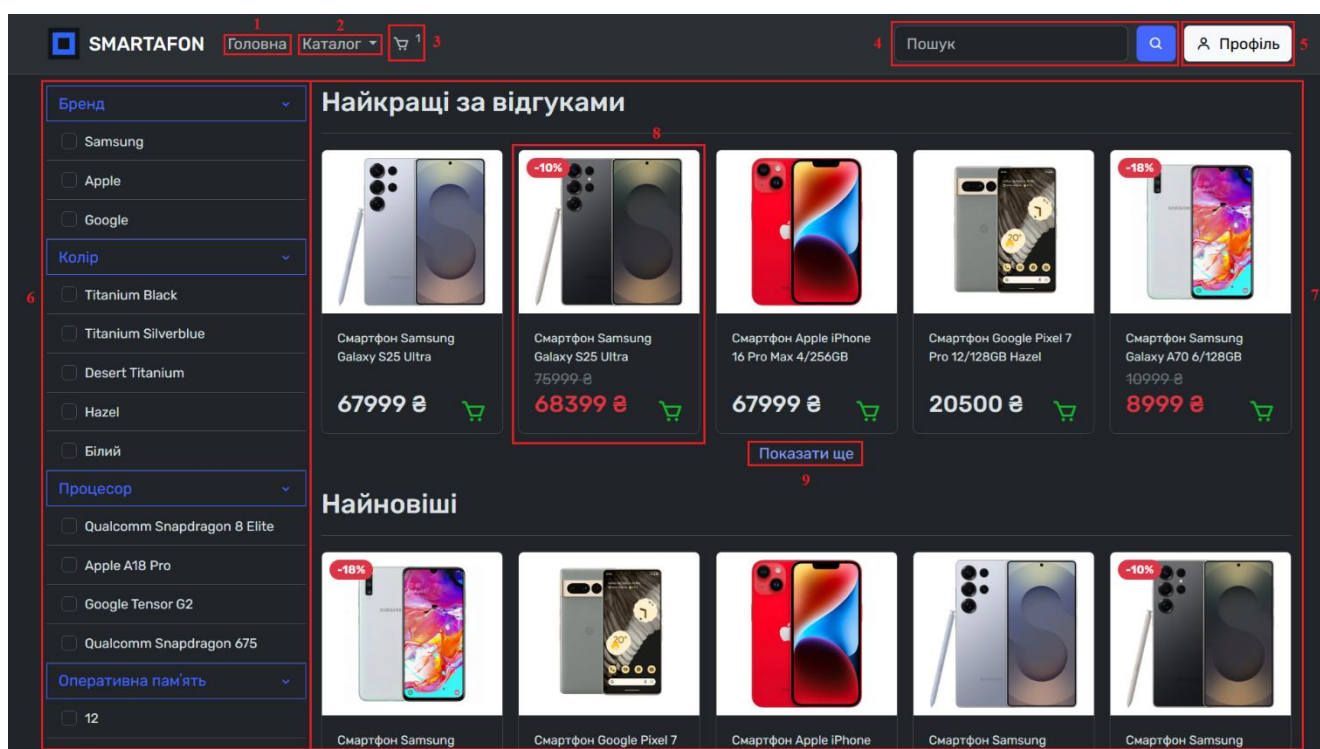


Рисунок Г.3 – Головна сторінка вебзастосунку для інтернет-магазину «Smartafon» з виділеною частиною основних елементів

Пояснення до елементів вебзастосунку:

- 1) Пункт навігаційного меню для переходу на головну сторінку вебзастосунка;
- 2) Випадаючий список каталогу, який містить посилання на певні категорії смартфонів, певні фільтри чи сегмент смартфонів;
- 3) Пункт навігаційного меню для переходу до кошику з смартфонами, що були додані користувачем;
- 4) Поле для пошуку смартфонів;

- 5) Кнопка для переходу до особистого кабінету, де можна знайти та відредагувати інформацію про користувача;
- 6) Фільтр для фільтрації смартфонів;
- 7) Не повні списки смартфонів, що розташовані за певним критерієм сортування, який відображений у заголовку секції над списком;
- 8) Карточка смартфона, що містить зображення, назву, колір смартфона, кількість пам'яті, ціну та кнопку для додавання до кошику;
- 9) Кнопка, що дозволяє перейти до сторінки вебзастосунка з повним списком смартфонів, що було відсортовано за певним критерієм сортування.

3.3 Завершення роботи програми

Для закриття вебзастосунку, користувачу потрібно лише закрити вкладку в веббраузері, або повністю закрити сам веббраузер.

ДОДАТОК Д – ТЕКСТ ВЕБЗАСТОСУНКУ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ СМАРТФОНІВ «SMARTAFON»

Нижче представлено текст вебзастосунок для інтернет-магазину смартфонів «Smartafon».

Admin/SmartphoneController.php

```
<?php

namespace App\Http\Controllers\AdminControllers;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Auth;
use App\Models\SmartphoneModel;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Gate;
use App\Services\SmartphoneImageService;

class SmartphoneController extends Controller {

    public function index() {
        Gate::denies('access-admin') && abort(403);
        $smartphones = SmartphoneModel::all();

        return view('admin.smartphonesTable', compact('smartphones'));
    }

    public function createPage() {
        Gate::denies('access-admin') && abort(403);

        return view('admin.smartphonesFormPage');
    }

    public function store(Request $request, SmartphoneImageService $imageService) {
        Gate::denies('access-admin') && abort(403);

        $smartphone = SmartphoneModel::create([
            'article' => $request->article,
            'name' => $request->name,
            'slug' => Str::slug($request->brand) . '-' . Str::slug($request->model_code) .
            '-' . Str::slug($request->article),
            'model_code' => $request->model_code,
            'color' => $request->color,
            'processor' => $request->processor,
            'ram' => $request->ram,
            'storage' => $request->storage,
            'os' => $request->os,
            'camera' => $request->camera,
            'front_camera' => $request->front_camera,
            'brand' => $request->brand,
            'sim_slots' => $request->sim_slots,
```

```

        'sim_slots_types' => $request->sim_slots_types,
        'battery_capacity' => $request->battery_capacity,
        'screen_size' => $request->screen_size,
        'screen_resolution' => $request->screen_resolution,
        'condition' => $request->condition,
        'price' => $request->price,
        'quantity' => $request->quantity,
    ]));

    if ($request->hasFile('images')) {
        $imageService->addImages($request->file('images'), $smartphone->id);
    }

    return response()->json(['success' => true]);
}

public function editPage(Request $request) {
    Gate::denies('access-admin') && abort(403);
    $smartphone = SmartphoneModel::findOrFail($request->id);
    return view('admin.smartphonesFormPage', compact('smartphone'));
}

public function update(Request $request, SmartphoneImageService $imageService) {
    Gate::denies('access-admin') && abort(403);
    $smartphone = SmartphoneModel::findOrFail($request->id);

    $smartphone->update([
        'article' => $request->article,
        'name' => $request->name,
        'model_code' => $request->model_code,
        'color' => $request->color,
        'processor' => $request->processor,
        'ram' => $request->ram,
        'storage' => $request->storage,
        'os' => $request->os,
        'camera' => $request->camera,
        'front_camera' => $request->front_camera,
        'brand' => $request->brand,
        'sim_slots' => $request->sim_slots,
        'sim_slots_types' => $request->sim_slots_types,
        'battery_capacity' => $request->battery_capacity,
        'screen_size' => $request->screen_size,
        'screen_resolution' => $request->screen_resolution,
        'condition' => $request->condition,
        'price' => $request->price,
        'new_price' => $request->new_price,
        'quantity' => $request->quantity,
    ]));

    if ($request->hasFile('images')) {
        $imageService->updateImages($request->file('images'), $request->id);
    }

    return response()->json(['success' => true]);
}

public function destroy(Request $request, SmartphoneImageService $imageService) {
    Gate::denies('access-admin') && abort(403);
    $smartphone = SmartphoneModel::findOrFail($request->id);

```

```

        $imageService->deleteImages($smartphone->id);
        $smartphone->delete();

        return redirect()->route('admin.smartphones');
    }
}

```

SmartphoneImageService.php

```

<?php
namespace App\Services;
use App\Models\ImageModel;
use Illuminate\Http\UploadedFile;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Str;

class SmartphoneImageService
{
    public function addImages(array $images, int $smartphoneId): void
    {
        foreach ($images as $image) {
            if (!$image->isValid()) continue;

            $extension = $image->getClientOriginalExtension();
            $filename = Str::random(40) . '.' . $extension;

            Storage::disk('public')->putFileAs('images', $image, $filename);

            ImageModel::create([
                'filename' => $filename,
                'filepath' => '/storage/images/' . $filename,
                'smartphone_id' => $smartphoneId,
            ]);
        }
    }

    public function updateImages(array $images, int $smartphoneId): void
    {
        $oldImages = ImageModel::where('smartphone_id', $smartphoneId)->get();

        foreach ($oldImages as $oldImage) {
            $imagePath = str_replace('/storage/', '', $oldImage->filepath);
            Storage::disk('public')->delete($imagePath);
            $oldImage->delete();
        }

        foreach ($images as $image) {
            $extension = $image->getClientOriginalExtension();
            $filename = Str::random(40) . '.' . $extension;
            Storage::disk('public')->putFileAs('images', $image, $filename);

            ImageModel::create([
                'filename' => $filename,
                'filepath' => '/storage/images/' . $filename,
                'smartphone_id' => $smartphoneId,
            ]);
        }
    }
}

```

```

public function deleteImages(int $smartphoneId): void
{
    $images = ImageModel::where('smartphone_id', $smartphoneId)->get();

    foreach ($images as $image) {
        $imagePath = str_replace('/storage/', '', $image->filepath ?? $image->url);
        Storage::disk('public')->delete($imagePath);
        $image->delete();
    }
}
}

```

ImagesModel.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\SmartphonesModel;

class ImagesModel extends Model
{
    use HasFactory;

    protected $fillable = [
        'filename',
        'filepath',
        'smartphone_id'
    ];

    public function smartphone()
    {
        return $this->belongsTo(SmartphonesModel::class);
    }
}

```

SmartphonesModel.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use App\Models\UsersModel;
use App\Models\ImagesModel;

class SmartphonesModel extends Model
{
    use HasFactory;

    protected $fillable = [
        'article',
        'model_code',
        'name',
        'slug',
    ];
}

```

```

        'model_code',
        'color',
        'processor',
        'ram',
        'storage',
        'os',
        'camera',
        'front_camera',
        'brand',
        'sim_slots',
        'sim_slots_types',
        'battery_capacity',
        'screen_size',
        'screen_resolution',
        'condition',
        'price',
        'new_price',
        'new_price',
        'quantity',
    ];

    public function reviews()
    {
        return $this->hasMany(ReviewsModel::class);
    }

    public function images()
    {
        return $this->hasMany(ImagesModel::class);
    }
}

```

ReviewsModel.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use App\Models\UsersModel;
use App\Models\SmartphonesModel;

class ReviewsModel extends Model
{
    use HasFactory;

    protected $fillable = [
        'text',
        'advantages',
        'disadvantages',
        'rate',
        'user_id',
        'smartphone_id',
    ];

    public function author()
    {
        return $this->belongsTo(UsersModel::class, 'user_id');
    }
}

```

```

    public function smartphone()
    {
        return $this->belongsTo(SmartphonesModel::class, 'smartphone_id');
    }
}

```

OrdersModel.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use App\Models\UsersModel;
use App\Models\SmartphonesModel;

class OrdersModel extends Model
{
    use HasFactory;

    protected $fillable = [
        'status',
        'smartphones_quantity',
        'delivery_address',
        'order_price',
        'user_id',
    ];

    public function user()
    {
        return $this->belongsTo(UsersModel::class, 'user_id');
    }
    public function smartphones()
    {
        return $this->belongsTo(SmartphonesModel::class, 'smartphone_id');
    }
}

```

Admin/ReviewController.php

```

<?php

namespace App\Http\Controllers\AdminControllers;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Auth;
use App\Models\ReviewsModel;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Gate;

class ReviewController extends Controller {

    public function index() {
        Gate::denies('access-admin') && abort(403);
        $reviews = ReviewsModel::all();
    }
}

```

```

        return view('admin.reviewsTable', compact('reviews'));
    }

    public function destroy(Request $request) {
        Gate::denies('access-admin') && abort(403);
        $review = ReviewsModel::findOrFail($request->id);

        $review->delete();

        return redirect()->route('admin.reviews');
    }
}

```

UsersModel.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use App\Models\ReviewsModel;
use App\Models\OrdersModel;

class UsersModel extends Authenticatable
{
    use HasFactory, Notifiable;

    protected $fillable = [
        'last_name',
        'first_name',
        'phonenummer',
        'email',
        'password',
        'role',
    ];

    protected $hidden = [
        'password',
        'remember_token',
    ];

    protected function casts(): array
    {
        return [
            'email_verified_at' => 'datetime',
            'password' => 'hashed',
        ];
    }

    public function reviews()
    {
        return $this->hasMany(ReviewsModel::class);
    }

    public function orders()
    {

```

```

        return $this->hasMany(OrdersModel::class);
    }
}

```

SmartphoneController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use App\Models\SmartphonesModel;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Gate;

```

```
class SmartphoneController extends Controller {
```

```

    public function index($slug) {
        $smartphone = SmartphonesModel::where('slug', $slug)->first();

        if (!$smartphone) {
            abort(404, 'Ця сторінка не існує, але, можливо існує в іншій реальності.');
```

```
        }

        return view('SmartphoneDetailed', compact('smartphone'));
    }

```

```

    private function getFilterItems()
    {

```

```

        $filterNames = [
            'brand' => 'Бренд',
            'color' => 'Колір',
            'processor' => 'Процесор',
            'ram' => 'Оперативна пам'ять',
            'storage' => 'Вбудована пам'ять',
            'os' => 'Операційна система',
            'camera' => 'Основна камера',
            'front_camera' => 'Фронтальна камера',
            'sim_slots' => 'Кількість SIM-карт',
            'sim_slots_types' => 'Тип SIM-карт',
            'battery_capacity' => 'Ємність акумулятора',
            'screen_size' => 'Діагональ екрана',
            'screen_resolution' => 'Роздільна здатність',
            'condition' => 'Стан',
        ];

```

```

        $filters = [
            'brand', 'color', 'processor', 'ram', 'storage', 'os',
            'camera', 'front_camera', 'sim_slots', 'sim_slots_types',
            'battery_capacity', 'screen_size', 'screen_resolution', 'condition'
        ];

```

```

        $filterItems = [];
        foreach ($filters as $filter) {
            $filterItems[] = [
                'title' => $filterNames[$filter] ?? $filter,
                'data' => SmartphonesModel::pluck($filter)->unique()->filter()-

```

```

>map(fn($val) => (string)$val->values()->toArray(),
    'queryName' => $filter
    ];
}
return $filterItems;
}

private function applyFilters(Request $request, $query)
{
    $filters = [
        'brand', 'color', 'processor', 'ram', 'storage', 'os',
        'camera', 'front_camera', 'sim_slots', 'sim_slots_types',
        'battery_capacity', 'screen_size', 'screen_resolution', 'condition'
    ];
    foreach ($filters as $filter) {
        $values = $request->query($filter);
        if (!empty($values)) {
            $query->whereIn($filter, (array)$values);
        }
    }
    return $query;
}

private function applySorting(Request $request, $query, &$page_title)
{
    $sort = $request->query('sort');
    switch ($sort) {
        case 'rank':
            $page_title = 'Найкращі за відгуками';
            $query->withAvg('reviews', 'rate')->orderByDesc('reviews_avg_rate');
            break;
        case 'novelty':
            $page_title = 'Новинки';
            $query->orderByDesc('created_at');
            break;
        case 'cheap':
            $page_title = 'Від дешевих до дорогих';
            $query->orderByRaw('coalesce(new_price, price)');
            break;
        case 'expensive':
            $page_title = 'Від дорогих до дешевих';
            $query->orderByRaw('coalesce(new_price, price) desc');
            break;
        default:
            $page_title = 'Смартфони';
    }
    return $query;
}

public function homePage() {
    $smartphones_best_by_reviews = SmartphonesModel::withAvg('reviews', 'rate')
        ->orderByDesc('reviews_avg_rate')
        ->take(10)
        ->get();
    $smartphones_latest = SmartphonesModel::latest()->take(10)->get();
    $smartphones = SmartphonesModel::all();
    $filterItems = $this->getFilterItems();
    return view('MainPage', compact('smartphones', 'smartphones_best_by_reviews',
'smartphones_latest', 'filterItems'));
}

```

```

public function goodsPage(Request $request) {
    $page_title = '';
    $query = SmartphonesModel::query();
    $query = $this->applyFilters($request, $query);
    $query = $this->applySorting($request, $query, $page_title);

    $smartphones = $query->take(20)->get();
    $filterItems = $this->getFilterItems();

    return view('SearchedPage', compact('smartphones', 'page_title', 'filterItems'));
}
}

```

ReviewController.php

<?php

```

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Auth;
use App\Models\ReviewsModel;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Gate;

class ReviewController extends Controller {

    public function store(Request $request) {
        Gate::denies('access-author') && abort(403);
        $review = ReviewsModel::create([
            'text' => $request->comment,
            'advantages' => $request->advantages,
            'disadvantages' => $request->disadvantages,
            'rate' => $request->rate,
            'user_id' => Auth::id(),
            'smartphone_id' => $request->id,
        ]);

        return response()->json(['success' => true]);
    }

    public function destroy(Request $request) {
        Gate::denies('access-author') && abort(403);
        $review = ReviewsModel::findOrFail($request->review_id);

        if ($review->user_id !== Auth::id()) {
            abort(403, 'Unauthorized action.');
        }
        $review->delete();

        return response()->json(['success' => true]);
    }
}

```

views/layouts/main.blade.php

```

<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}" data-bs-theme="dark">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@yield('title', 'Smartafon - Інтернет-магазин смартфонів')</title>
    <meta name="csrf-token" content="{{ csrf_token() }}">
    <meta property="description" content="@yield('description', 'Smartafon - Інтернет-
магазин смартфонів | Багато, гарно та дешево')"/>
    <meta property="og:url" content="{{ config('app.url') }}" />.
    <meta property="og:type" content="website"/>.
    <meta property="og:title" content="@yield('title', 'Smartafon - Інтернет-магазин
смартфонів')"/>
    <meta property="og:description" content="@yield('description', 'Smartafon - Інтернет-
магазин смартфонів | Багато, гарно та дешево')"/>
    <meta property="og:image" content="{{ asset('img/favicon.png') }}" />
    <meta name="msapplication-TileColor" content="#0075ff">

    {{-- WebsiteIcons --}}
    <link type="image/x-icon" rel="shortcut icon" href="{{ asset('img/favicon.png') }}">
    <link rel="icon" type="image/png" sizes="32x32" href="{{ asset('img/favicon.png') }}">
    <link rel="icon" type="image/png" sizes="16x16" href="{{ asset('img/favicon.png') }}">
    <meta name="msapplication-config" content="{{ asset('browserconfig.xml') }}" />
    <link rel="manifest" href="{{ asset('site.webmanifest') }}">

    <link rel="stylesheet" href="{{ asset('css/app.min.css?v1') }}">
    {{-- Boxicons --}}
    <link href="https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css" rel="stylesheet">
    {{-- JQuery --}}
    <script src="https://code.jquery.com/jquery-3.7.1.min.js" integrity="sha256-
/JqT3SQfawRcv/BIHPThkBvs00EvtFFmqPF/lyI/Cxo=" crossorigin="anonymous"></script>

    @yield('header_scripts')
</head>

<body>
    <div class="loader-wrapper">
        <div class="loader"></div>
    </div>
    @yield('header')

    <main class="main_section">
        <div class="container-fluid">
            @yield('content')
        </div>
    </main>
    @yield('footer')

    @yield('scripts')
    <script src="{{ asset('js/app.js?v1') }}" defer></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.6/dist/js/bootstrap.bundle.min.js"
integrity="sha384-j1CDi7MgGQ12Z7Qab0qlWQ/Qqz24Gc6BM0thvEMVjHnfyYGF0rmFCozFSxQBxwHKO"
crossorigin="anonymous"></script>
</body>

</html>

```

views/MainPage.blade.php

```

@extends('layouts.main')

@section('header_scripts')
@endsection

@section('header')
    @include('includes.header')
@endsection

@section('content')
    <div class="filter-block d-flex bg-body sticky-element">
        <button type="button" class="btn w-100 btn-sm btn-primary align-items-center
filter-btn filter-btn-long" data-bs-toggle="offcanvas"
        data-bs-target="#categories" aria-controls="categories" aria-label="Фільтр">
            <i class='bx bxs-filter-alt'></i>
        </button>
    </div>
    <div class="d-flex">
        <div class="d-flex">
            @include('includes.filter')
        </div>
        <div class="d-flex flex-column h-100">
            @include('includes.mainPageSection', [
                'header' => 'Найкращі за відгуками',
                'link' => route('goods', ['sort' => 'rank']),
                'smartphones' => $smartphones_best_by_reviews,
            ])
            @include('includes.mainPageSection', [
                'header' => 'Найновіші',
                'link' => route('goods', ['sort' => 'novelty']),
                'smartphones' => $smartphones_latest,
            ])
        </div>
    </div>
@endsection

@section('footer')
    @include('includes.footer')
@endsection

```

views/SmartphoneDetailed.blade.php

```

@extends('layouts.main')

@section('header')
    @include('includes.header')
@endsection

@section('content')
    <div class="page-header">
        <div class="d-flex page-header-content justify-content-between">
            <div class="header-top d-flex flex-column">
                {{-- Breadcrumbs --}}
                <nav class="mb-2" aria-label="breadcrumb">
                    <ol class="breadcrumb mb-0 fs-7">
                        <li class="breadcrumb-item">
                            <a href="{{ route('home') }}">Домашня сторінка</a>
                        </li>

```

```

        <li class="breadcrumb-item">
            <a href="{ route('goods') }">Товари</a>
        </li>
        <li class="breadcrumb-item" aria-current="page">
            Смартфони {{ $smartphone->brand }}
        </li>
    </ol>
</nav>
</div>
</div>
</div>
<ul class="nav nav-tabs my-3 fs-7 bg-body border-bottom" id="tabs-nav">
    <li class="nav-item">
        <a role="button" onclick="changeTab('info')" data-tab-link="info" class="nav-link active fs-7" aria-current="page">Усе про товар</a>
    </li>
    <li class="nav-item">
        <a role="button" onclick="changeTab('specifications')" data-tab-link="specifications" class="nav-link fs-7">Характеристики</a>
    </li>
    <li class="nav-item">
        <a role="button" onclick="changeTab('reviews')" data-tab-link="reviews" class="nav-link fs-7">Відгуки</a>
    </li>
</ul>

<div id="tabs">
    @include('includes.detailedPage.info')
    @include('includes.detailedPage.specifications')
    @include('includes.detailedPage.reviews')
</div>
@endsection

@section('footer')
    @include('includes.footer')
@endsection

@section('scripts')
    <script>
        function changeTab(tabId) {
            $('.nav-link[data-tab-link]').removeClass('active');
            $('.nav-link[data-tab-link]').removeAttr('aria-current');
            $('.nav-link[data-tab-link="' + tabId + '"]').addClass('active');
            $('.nav-link[data-tab-link="' + tabId + '"]').attr('aria-current', 'page');

            $('#tabs [data-tab]').hide();
            $('#tabs [data-tab="' + tabId + '"]').show();
            $(window).scrollTop(0);
            $('html, body').scrollTop(0);
        }
    </script>

    <script>
        $(' .star-rating:not(.readonly) label').click(function() {
            $(this).css('transform', 'scale(1.2)');
            setTimeout(() => {
                $(this).css('transform', 'scale(1)');
            }, 200);
        });
    </script>
@endsection

```

views/includes/header.blade.php

```

<nav class="navbar sticky-top navbar-expand-lg bg-body-tertiary border-bottom">
    <div class="container-fluid">
        <a class="navbar-brand d-flex align-items-center" href="{{ route('home') }}">
            SMARTAFON
            </a>

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent"
            aria-controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                <li class="nav-item">
                    <a class="nav-link" href="{{ route('home') }}">
                        Головна
                    </a>
                </li>
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-
toggle="dropdown"
                        aria-expanded="false">
                        Каталог
                    </a>
                    <ul class="dropdown-menu">
                        <li>
                            <a class="dropdown-item" href="{{ route('goods', ['brand'
=> 'Samsung']) }}">
                                Samsung
                            </a>
                        </li>
                        <li>
                            <a class="dropdown-item" href="{{ route('goods', ['brand'
=> 'Apple']) }}">
                                Apple
                            </a>
                        </li>
                        <li>
                            <a class="dropdown-item" href="{{ route('goods', ['brand'
=> 'Xiaomi']) }}">
                                Xiaomi
                            </a>
                        </li>
                        <li>
                            <hr class="dropdown-divider">
                        </li>
                        <li><a class="dropdown-item" href="{{ route('goods', ['sort'=>
'rank']) }}">Найкращі за відгуками</a></li>
                        <li><a class="dropdown-item" href="{{ route('goods', ['sort'=>
'cheap']) }}">Найдешевші</a></li>
                        <li><a class="dropdown-item" href="{{ route('goods', ['sort'=>
'expensive']) }}">Найдорожчі</a></li>
                    </ul>
                </li>
                <li class="nav-item d-flex align-item-center">
                    <a class="nav-link icon-link" href="{{ route('cart') }}"><i

```

```

class='bx bx-cart'></i><sup class="counter" id="cart-counter">{{Auth::user()->cart()->count()}}</sup></a>
        <a class="nav-link mobile" href="{{ route('cart') }}">Кошик<sup
class="counter" id="cart-counter">{{Auth::user()->cart()->count()}}</sup></a>
    </li>
</ul>

    <form class="header-right-section d-flex gap-2" role="search">
        <div class="d-flex w-100 gap-2">
            <input class="form-control" type="search" placeholder="Пошук" aria-
label="Search">
                <button class="btn btn-primary d-flex align-items-center"
type="submit"><i class='bx bx-search'></i></button>
            </div>

            @if(Auth::check())
                {{-- Кнопка особистого кабінету --}}
                <a class="btn d-flex align-items-center justify-content-sm-center btn-
light" type="button"
                    href="{{ route('profile.edit') }}" id="account-btn" aria-
label="Обліковий запис" aria-small-label="Профіль">
                    <i class='bx bx-user'></i>
                </a>
            @else
                {{-- Кнопка авторизації --}}
                <a class="btn d-flex align-items-center justify-content-sm-center btn-
secondary" type="button"
                    href="{{ route('login') }}" id="account-btn" aria-label="Увійти в
обліковий запис" aria-small-label="Увійти">
                    <i class='bx bx-log-in-circle' ></i>
                </a>
            @endif

            @if(Auth::check() && Auth::user()->role == 'admin')
                <a class="btn d-flex align-items-center justify-content-sm-center btn-
outline-light" type="button"
                    href="{{ route('admin.index') }}" id="account-btn" aria-
label="Адмін панель" aria-small-label="Адмін">
                    <i class='bx bx-user'></i>
                </a>
            @endif
        </form>
    </div>
</div>
</nav>

```