МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри_____ Є. О. Давиденко
підпис
«___» _____2022 р.

КОМПЛЕКСНА КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРІВ

Програмне забезпечення інтернет-магазину товарів для косметологів

Спеціальність «Інженерія програмного забезпечення»

Том 0

121 – ККРБ.1 – 408.21810811,408.21810802

Студент _____ В. Д. Діденко
підпис
«__» _____ 2022 р.

Студент _____В.О. Антонов
підпис
«__» _____ 2022 р.

Консультант канд. тех. наук, доцент _____ А. О. Алексєєва
підпис
«__» _____ 2022 р.

Керівник канд. тех. наук, доцент _____ Г. В. Горбань
підпис
«__» _____ 2022 р.

Миколаїв – 2022

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

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії програмного забезпечення, канд.техн.наук, доцент,

_____Є.О. Давиденко

«____»_____2022 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи бакалавра

Видано студентам групи _408_ факультету комп'ютерних наук

Антонов В'ячеслав Олександрович_____
.

(прізвище, ім'я, по батькові студента)

Діденко Владислав Денисович _____
.

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

«Програмне забезпечення інтернет-магазину товарів для косметологів

Затверджена наказом по ЧНУ від «01» _____грудня_____ 2021 р. № ____314
_

2. Строк представлення кваліфікаційної роботи «____» _____ 2022 р.

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

Вхідні дані до роботи − функціональні та нефункціональні вимоги до програмного забезпечення, відстеження помилок. Результат – функціюнуючий інтернет магазин.

4. Перелік питань, що підлягають розробці

- аналіз предметної області та існуючих аналогів;
- створення блок-схеми алгоритму роботи модулів;
- розробка вимог до системи на основі інформації про предметну область та існуючі аналоги;
- проєктування зовнішнього вигляду вебзастосунку;

5. Перелік графічних матеріалів:

Презентація

.

6. Завдання до спеціальної частини

Дослідження питань охорони праці, які безпосередньо пов'язані з діяльністю розробника програмного забезпечення .

7. Консультанти:

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

Керівник роботи канд. техн. наук, доцент Горбань Гліб Валентинович

(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Антонов В'ячеслав Олександрович

(прізвище, ім'я, по батькові студента)

(підпис)

2022р Діденко В. Д. 121 – ККРБ – 408.21810811

2022р Антонов В. О. 121 – ККРБ – 408.21810802

Кафедра інженерії програмного забезпечення

Програмне забезпечення інтернет-магазину товарів для косметологів

Діденко Владислав Денисович

(прізвище, ім'я, по батькові студента)

_____

(підпис)

Дата видачі завдання «____» _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

Тема: «Програмне забезпечення інтернет-магазину товарів для косметологів»

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	12.11.2021	13.11.2021	
2.	Огляд літератури за темою роботи	15.11.2021	17.11.2021	
3.	Порівняльний аналіз аналогів	18.11.2021	21.11.2021	
4.	Специфікація вимог до ПЗ	05.01.2022	30.05.2022	
6.	Моделювання та проєктування вебзастосунку	07.01.2022	30.05.2022	
7.	Моделювання прототипу користувацького інтерфейсус	15.02.2022	30.05.2022	
8.	Відгук керівника КРБ	08.05.2022	20.05.2022	
9.	Оформлення КРБ та презентації	21.05.2022	21.05.2022	
10.	Попередній захист	22.05.2022	22.05.22	
11.	Рецензування	14.06.2022	14.06.2022	
12.	Завершення оформлення КРБ та презентації	15.06.2022	19.06.2022	
13.	Захист кваліфікаційної роботи	28.06.2022	28.06.2022	

Розробив студент Діденко Владислав Денисович _____

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

«__» _____ 20__ р.

2022р Діденко В. Д. 121 – ККРБ – 408.21810811
2022р Антонов В. О. 121 – ККРБ – 408.21810802

Кафедра інженерії програмного забезпечення

Програмне забезпечення інтернет-магазину товарів для косметологів

Розробив студент Антонов В'ячеслав Олександрович _____

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

«__» _____ 20__ р.

Керівник роботи

Канд. техн. наук, доцент Горбань Гліб Валентинович _____

«__» _____ 20__

АНОТАЦІЯ

До комплексної кваліфікаційної роботи бакалавра

«Програмне забезпечення інтернет-магазину товарів для косметологів»

Студент 408 гр.: Діденко Владислав Денисович

Студент 408 гр.: Антонов Вячеслав Олександрович

Керівник: канд. технічних наук, доцент Горбань Г.В.

Актуальність: полягає в створенні кращих умов для просування товарів для косметологів і їх продажу використовуючи вебзастосунок.

Мета: удосконалення салону краси, а саме представлення косметологічних послуг та товарів, шляхом створення вебзастосунку для просування компанії та продажу косметологічних продуктів та послуг з метою покращення прибутку компанії.

Для досягнення мети треба вирішити наступні задачі:

- аналіз існуючих аналогів;
- моделювання вебзастосунку;
- проєктування вебзастосунку;
- розробка вебзастосунку;
- тестування системи із визначенням відповідності до поставлених задач.

Об’єкт роботи: інформаційні процеси управління та функціонування системи інтернет-магазину товарів для косметологів.

Предметом роботи: є механізми та практичні аспекти реалізації інтернет-магазину товарів для косметологів.

У вступі визначається актуальність теми, що приймається за мету та невеликий огляд поставленої задачі, предмет роботи та об’єкт роботи.

У першому розділі описується аналітична частина, тобто огляд існуючих інтернет магазинів. Визначається основна особливість обраних веб застосунків, завдяки чому було сформовано загальне розуміння предметної області.

У другому розділі описується архітектура вебзастосунку сценарії використання системи і моделювання взаємодії між об'єктами. Розробка прототипу та діаграма класів.

У висновках проводиться аналіз роботи та отриманих результатів.

Загальна частина викладена на 35 сторінок, вона містить 2 розділи, 13 ілюстрацій, 2 таблиці, 21 джерел в переліку посилань.

Ключові слова: Miro, web, JS, e-commerce, Spring, інтеренет магазин, MVC, база даних, архітектура,

ABSTRACT

of the Bachelor’s Thesis

«Online store software for cosmetologists »

Student 408 gr.: Didenko Vladislav

Student 408 gr.: Antonov Vyacheslav

Supervisor: Candidate of Technical Sciences (Ph. D.), Associate Professor Horban G.

Relevance of the qualification work is to create better conditions for the promotion of products for cosmetologists and their sale using a web application.

Purpose of the qualification work is to improve the beauty salon, namely the presentation of cosmetic services and goods, by creating a web application to promote the company and the sale of cosmetic products and services in order to improve the company's profits.

To achieve this goal you need to solve the following tasks:

– analysis of existing analogues;
– web application modeling;
– web application design;
– web application development;
– testing of the system to determine compliance with the tasks.

Object of the qualification work is information processes of management and functioning of the system of online store of goods for cosmetologists.

The subject of the qualification work are the mechanisms and practical aspects of the implementation of the online store of goods for cosmetologists.

The introduction determines the relevance of the topic to be taken as a goal and a brief overview of the task, the subject of research and the object of research.

The first section describes the analytical part, ie an overview of existing online stores. The main feature of the selected web applications is determined, thanks to which a general understanding of the subject area was formed.

The second section describes the web application architecture, system usage scenarios, and object interaction modeling. Prototype development and class diagram.

The conclusions analyze the work and the results obtained.

The general part is set out on 35 pages, it contains 2 sections, 13 illustrations, 2 tables, 21 sources in the list of references.

Keywords: Java, web, JS, e-commerce, Spring, architecture, Miro, database, MVC

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ 10

ВСТУП 11

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВА ЗАДАЧІ 13

1.1 Порівняльний аналіз аналогів 13

1.2 Опис предметної сфери 19

1.3 Специфікація вимог до ПЗ 20

Висновки до розділу 1 24

2 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ 25

2.1 Сценарії використання системи 25

2.2 Моделювання взаємодії між об’єктами 30

2.3 Діаграма класів 34

2.4 Моделювання прототипу користувацького інтерфейсу 38

Висновки до розділу 2 40

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ 41

ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ – програмне забезпечення

БД – база даних

VSC – Visual Studio Code

CSS – Cascading Style Sheets

SASS – Syntactically Awesome Stylesheets

SPA – Single page application

MPA – Multi page application

MVC – Model, View та Controller

CDN – Content Delivery Network

HTTP – HyperText Transfer Protocol

ВСТУП

Кожен володар офлайн-бізнесу хоче вийти на максимально можливий заробіток для нього. Є багато варіантів, як зробити свій дохід більшим, найпростішим і найефективнішим способом є просування свого продукту за допомогою створення вебзастосунку, за допомогою якого є можливість збільшити продаж продуктів використовуючи функціонал Інтернет-магазину.

Актуальність та науково-практичне значення.

Сучасний світ все більше поринає у світ інформаційних послуг та інформаційного контенту. Велика кількість користувачів в кожній країні використовують мережу інтернет для задоволення власних потреб та бажань. Задоволення власних потреб та бажань є надзвичайно важливим та пріоритетним для високого рівня соціальної завершеності.

Кожен працівник або підприємство у сфері послуг не може існувати без постійного потоку клієнтів. Тому, в останній час просування сфери послуг переходить у мережу інтернет. Послуги потрібно продавати через вебзастосунки. Візитки та біл-борди як альтернатива просуванню певного підприємства вже не така продуктивна як була 10 років тому.

XXI століття – століття комп’ютерних технологій та настання цифрової ери. Майже кожен має доступ до інтернету та проводить за телефоном чи іншим пристроєм майже весь свій час. Зараз продаж послуг та товарів здебільшого проводиться у вебзастосунках. Також цьому сприяє напрямок України у сферу діджиталізації – “Країна у смартфоні”. У теперішній час навіть люди похилого віку знають як замовляти товари в інтернеті.

Особливої потреби у Інтернет-магазині, компанія “PerfectLine” набула в період пандемії коронавірусу в світі, коли більшість людей або зовсім залишилась вдома або перейшла на онлайн форму життя. Приблизно 80% покупців косметики були клієнти косметологічного центру, а інші із

соціальної мережі Instagram. Так як разом із пандемією, а потім з воєнною агресією російської федерації кількість клієнтів косметологічного центру зменшилася, а працювати потрібно, то без існування інтернет магазину не можливо уявити існування центру.

Також у застосунку повинен бути інтернет магазин, щоб клієнт міг з легкістю замовити товар або послугу онлайн.

Об'єктом роботи є інформаційні процеси управління та функціонування системи інтернет-магазину товарів для косметологів.

Предметом роботи є механізми та практичні аспекти реалізації інтернет-магазину товарів для косметологів.

Метою роботи є удосконалення салону краси, а саме представлення косметологічних послуг та товарів, шляхом створення вебзастосунку для просування компанії та продажу косметологічних продуктів та послуг з метою покращення прибутку компанії.

Відповідно до мети визначено такі завдання:

- аналіз існуючих аналогів;
- моделювання вебзастосунку;
- проєктування вебзастосунку;
- розробка вебзастосунку;
- тестування системи із визначенням відповідності до поставлених задач.

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВА ЗАДАЧІ

1.1 Порівняльний аналіз аналогів

Для аналізу сучасного стану предметної області проведено дослідження вебзастосунків з функціоналом Інтернет-магазину. Для аналізу було обрано декілька популярних косметологічних компаній, таких як Alvi prague, Piel cosmetics і Zemits. Кожна з цих компаній орієнтована на український ринок, саме тому вони були обрані для дослідження, оскільки є прямими конкурентами.

Один із лідерів в якості та продажі косметологічної продукції котрий використовує вебзастосунк є Alvi prague.

Alvi prague – Європейський виробник косметологічного і медичного устаткування найвищої якості. В асортименті компанії широко представлені як апарати які включають класичні методики, так і інноваційні технології [1].

Веб-сайт Alvi prague володіє великою кількістю можливостей та функції.

Виробник – Україна.

Мова реалізації: PHP 7 + Laravel framework & eCommerce CMS OpenCart. JavaScript libraries and functions: Sentry, Vue JS, core-js, React, Bootstrap.js

Основі функції, характеристики:

- реєстрація;
- пошук товарів;
- порівняння товарів;
- зміна мови інтерфейсу;
- оплата товару;
- додавання товару до кошику.

До переваг Alvi prague можна віднести: швидкість загрузки сайту при великій кількості товарів та адаптивність. Усі сторінки вебсайту виконані в єдиному стилі

Основним недоліком вебзастосунку є занадто плоска архітектура, через яку деякі товари винесені в окремий розділ, конкретна пропозиція може бути настільки новаторською, що віднести її до однієї категорії – отже, ввести в оману цільову аудиторію. Користувачі зможуть зрозуміти пропозицію набагато краще, якщо увімкнути її в кілька проміжних категорій, щоб встановити контекст офферу [4]. Наприклад, Alvi prague показує відвідувачам довгий список із більш ніж 30 пунктів у категорії «Косметологічні апарати». Замість того, щоб бути «похованими» в кількох широких категоріях, окремі апарати відкриваються на своїй сторінці, будучи доступними безпосередньо з меню в глобальній навігації. Щоб добратися до деяких товарів потрібно зробити більше ніж три кліки.

Інтерфейс вебзастосунку можна побачити на рис. 1.1.

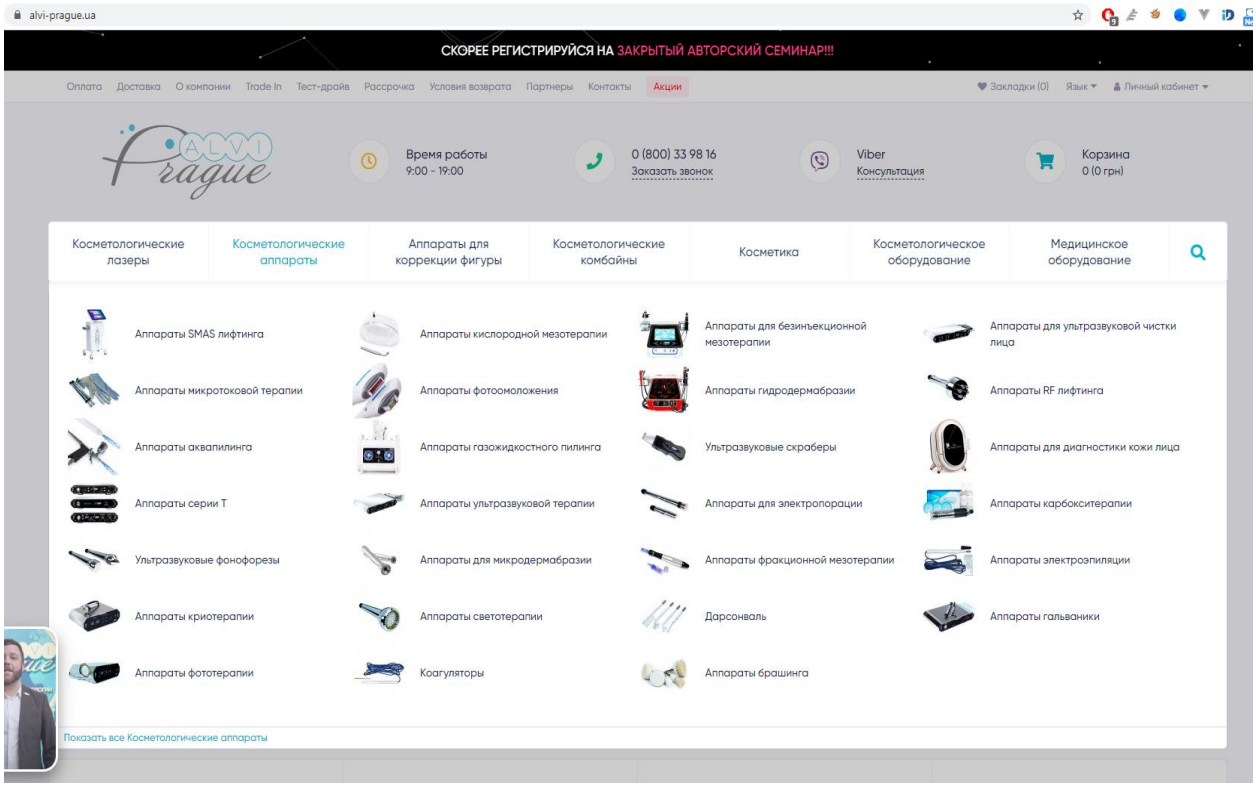

Рисунок 1.1 – Інтерфейс вебзастосунку Alvi Prague

Другим вебзастосунком, який досліджувався, є Piel cosmetics.

Piel cosmetics – це косметичний бренд на основі гіалуронової кислоти і цитрату срібла, який допомагає сучасній жінці виглядати так, ніби вона часто відвідує косметолога [2]. З функціональним веб-сайтом.

Виробник: Україна.

Мова реалізації: PHP 7 + Laravel framework & eCommerce CMS OpenCart

Основі функції, характеристики:

− реєстрація;

− онлайн-чат;

− пагінація;

− зміна мови інтерфейсу;

− оплата товару;

− додавання товару до кошику;

− усі сторінки веб-сайту виконані в єдиному стилі.

До переваг вебзастосунку можна віднести швидкість загрузки сайту та сучасні способи взаємодії, а саме зв’зок через онлайн-чат з користувачем.

Основними недоліками вебзастосунку є:

− погана структура сайту;

− застарілий дизайн;

− використовування маленьких шрифтів;

− використовування різних розмірів для медіа файлів, на одній сторінці;

У головному меню відображені однакові розділи сайту з різними назвами, це плутає відвідувача. Непослідовність навігації, користувач повинен зрозуміти меню навігації з першого погляду на веб-інтерфейс, щоб впевнено застосувати це розуміння до виконання задачі. Структура сайту є не зрозумілою.

Інтерфейс вебзастосунку Piel cosmetics на рис. 1.2.

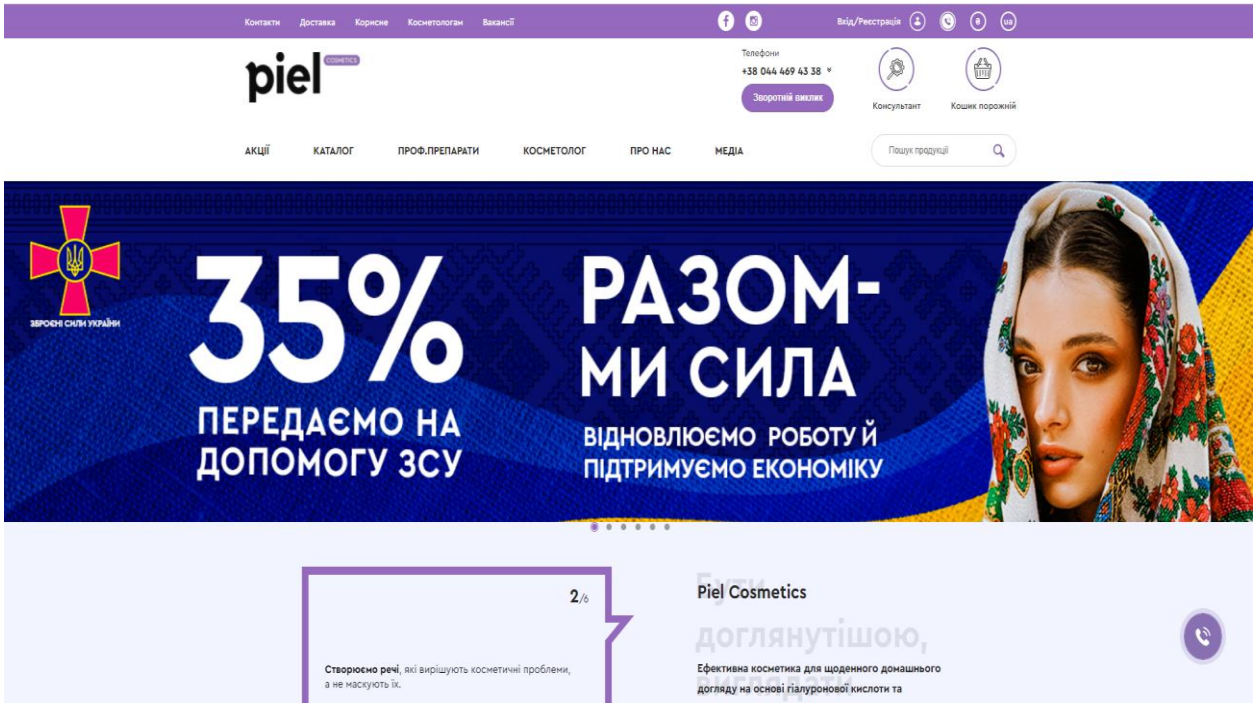

Рисунок 1.2 – Інтерфейс вебзастосунку Piel cosmetics.

Останнім об’єктом дослідженим став вебзастосунок програмним рішенням став мобільний застосунок Zemits.

Zemits – це виробник обладнання для салонів краси. Купуючи у нас продукцію для свого косметичного салону. Заснований у 2010 році [3].

Виробник: Україна.

Мова реалізації: PHP 7 + Laravel framework & CMS Tilda, js-core, React, Hammer JS.

Основі функції, характеристики:

- онлайн-чат;
- пагінація;
- зміна мови інтерфейсу;
- оплата товару;
- сортування товарів.

До переваг вебзастосунку можна віднести високі показники в пошукових системах, швидкість вебсайту. Також, архітектуру вебсайту, яка є плоскою, виконана за правилами, тобто потрібно зробити максимум 3 кліки, щоб потрапити на потрібну сторінку. Навігаційне меню має всі потрібні сторінки-якорі для навігації.

Власник вебсайту використовує лише якісну графіку для свого продукту, що дає розуміння як власник ставиться до свого бренду. Значна частина сайтів страждають від графіки низької якості.

Недоліки:

- Багато одноманітних категорій котрі можуть бути об’єднані в одну;
- Перевантажене навігаційне меню;
- Застарілий дизайн.

Інтерфейс вебзастосунку можна побачити на рис. 1.3.

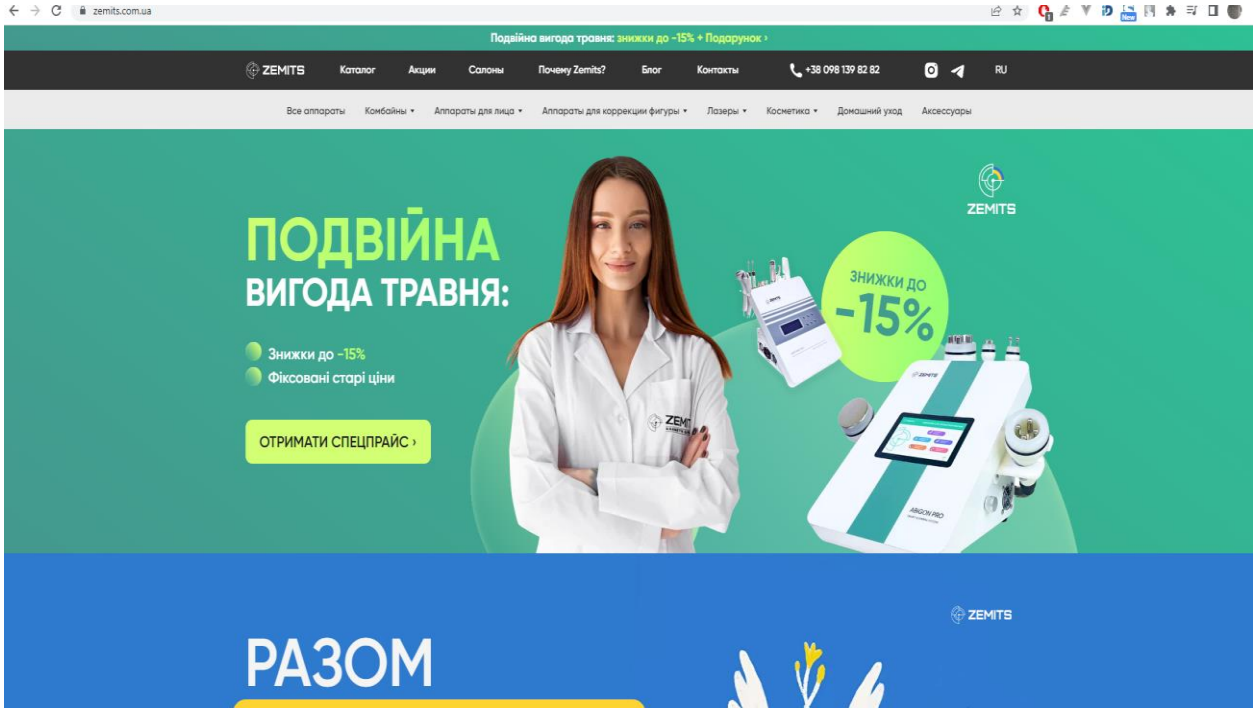

Рисунок 1.3 – Інтерфейс вебзастосунку Zemits.

В таблиці 1.1 приведено порівняння аналогічних програмних рішень з системою, яка розроблюється.

Таблиця 1.1 – Порівняння аналогічних програмних рішень

Показники	Alvi prague	Piel cosmetics	Zemits	Система КРБ
Створення власного кабінету користувача	+	+	-	+
Оплата товару за допомогою PayPal	-	+	-	+
Багатомовний веб-сайт	+	+	+	-
Багатоаспектна або фасетна навігація	-	-	-	+
Пагінація, безкінечна прокрутка	+	+	+	+
Порівняння товарів	+	-	-	-
Сумарні показники	4	4	2	4

Проаналізувавши та порівнявши переваги та недоліки вже існуючих вебзастосунків з функцією Інтернет-магазину, зроблено висновок, що існує потреба у створенні вебзастосунку, який усуне недоліки проаналізованих програмних продуктів, об’єднає їх переваги та реалізує нові для українського ринку способи оплати товарів за допомогою PayPal.

1.2 Опис предметної сфери

Предметною сферою дослідження є вебзастосунок Інтернет-магазину для косметологів.

Інтернет-магазин – це купівля та продаж товарів чи послуг через інтернет, а також передача грошей та даних для завершення продажу. Він також відомий як електронна комерція або Інтернет-комерція [5].

Сьогодні інтернет магазин використовується майже в усіх сферах бізнесу. Існує 5 популярних типи моделей електронної комерції:

− B2C – це найпопулярніша модель з існуючих. Від бізнесу до споживача, тобто продаж товарів або послуг відбувається між бізнесом і споживачем;

− B2B – від бізнесу до бізнеса. Тобто коли бізнес продає товари або послуги іншому підприємству. Така комерція зазвичай не орієнтована на споживачів. Здебільшого включає в себе такі продукти як сировина або програмне забезпечення;

− D2C – пряма електронна комерція для споживачів. Є найновішою моделлю яка постійно змінюється. D2C означає, що компанія продає напряму споживачу, не звертаючись до продавця, популярним прикладом є підписка або продажі через соціальні мережі;

− C2C – комерція відноситься до продажу товару або послуги іншому споживачеві. Продаж від споживачів до споживачів. Гарним прикладом є OLX;

− C2B – від споживачів до бізнесу, коли особа продає свої продукти або послуги компанії, здебільшого такі споживачі працюють як незалежні журналісти, автори, фотографи або програмісти.

В нашому випадку буде прийнята модель B2C, продаж косметологічних товарів та послуг. Для того, щоб побудувати гарну архітектуру сайту, потрібно знати з якою електронною комерцією потрібно працювати. Модель B2C представлено на рис. 1.5.

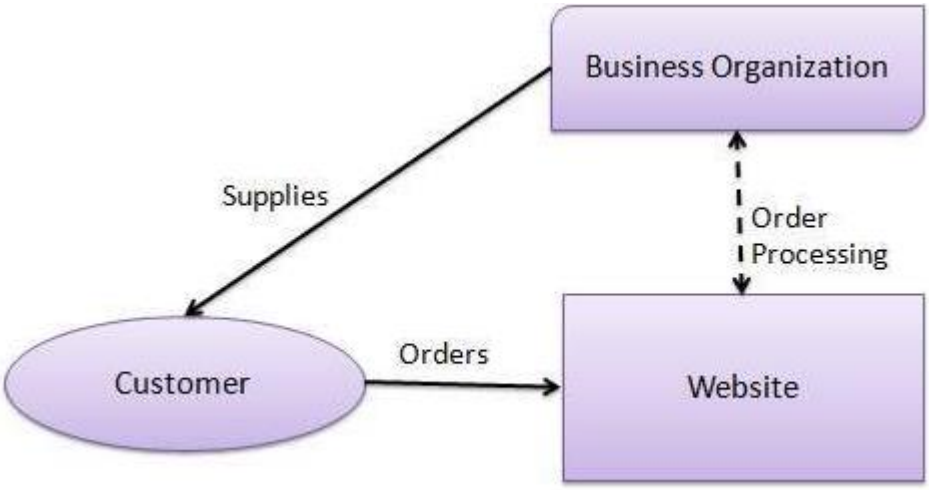

Рисунок 1.5 – Модель B2C

В цій моделі споживач переходить на веб-сайт, вибирає продукт після чого оформлює заказ. Після оплати продукт відправляється до клієнта.

Для залучення клієнтів потрібна потужна реклама. Високі інвестиції з точки зору обладнання та програмного забезпечення. Підтримка або гарне обслуговування клієнтів.

1.3 Специфікація вимог до ПЗ

На основі аналізу предметної області та дослідження існуючих програмних рішень вебзастосунків з функціоналом Інтернет-магазину сформовані вимоги до програмного забезпечення, що розроблюється, відповідно до мети. Сформовано специфікацію вимог до програмного

забезпечення, що представляє собою опис поведінки програмного забезпечення, що розроблюється.

ПРИЗНАЧЕННЯ ПРОЄКТУ

Вебзастосунок створюється для підвищення продаж та появи нових клієнтів зі всієї України.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Продаж косметологічних послуг та товарів за допомогою вебсайту.

Характеристики користувачів

У рамках системи вебзастосунку можна визначити декілька класів користувачів:

− Зареєстрований користувач: користувач, який має доступ до дозволеного функціоналу вебзастосунка, який розроблюється в межах КРБ;

− Відвідувач: користувач, який не має власного аккаунту, має доступ лише для перегляду вебсторінки;

− Адміністратор: привілейований користувач, який має доступ управління до категорій брендів, продуктів, клієнтів, замовлень, звітів про продажі.

ФУНКЦІЇ СИСТЕМИ

Функція системи 1:

Застосунок для покупок.

Вхідна і вихідна інформація

Вхідна інформація:

− користувач реєструється;

− користувач переходить до сторінки з товаром;

− обирає необхідні параметри товару;

− система зберігає ці налаштування;

− клієнт оформлює замовлення.

Вихідна інформація:

- зберігається інформація про купівлю в особистому кабінеті;
- інформація про купівлю надходить до Google Chart;
- клієнту доставляється товар.

Функціональні вимоги

Вебзастосунок повинен надсилати електронні листи для реєстрації клієнта та підтвердження замовлення.

Функція системи 2:

Оплата товару за допомогою PayPal.

Вхідна і вихідна інформація

Вхідна інформація:

- клієнт обирає товар;
- вводить свою PayPal-інформацію;
- переходить на сторінку чека.

Вихідна інформація:

- зберігається інформація про купівлю в особистому кабінеті;
- інформація про купівлю надходить до Google Chart;
- гроші нараховуються на PayPal- аккаунт;
- клієнту доставляється товар.

Функціональні вимоги

Вебзастосунок повинен використовувати PayPal API для оплати.

Функція системи 3:

Створення облікового запису

Вхідна і вихідна інформація

Вхідна інформація:

- покупець переходить до форми реєстрації та заповнює контактні дані;
- система здійснює валідацію даних;

- система зберігає обліковий запис;
- покупець отримує повідомлення про успішну реєстрацію.

Вихідна інформація:

- збережено новий обліковий запис.

Функціональні вимоги

Вебзастосунок повинен мати лише одного клієнта на одну почту.

Функція системи 4

Переклад розпізнаного тексту

Вхідна і вихідна інформація

Вхідна інформація:

- покупець:
  - заповнює контактну інформацію;
  - пише текст відгука;
  - ставить оцінку;
  - обирає збереження відгуку;
- система записує відгук до БД.

Вихідна інформація:

- Відгук збережено.

Функціональні вимоги

Застосунок повинен перекладати розпізнаний текст

ВИМОГО ДО ПРОГРАМНОГО ЗЕБЕЗПЕЧЕННЯ

Мова і технологія розробки

Для розробки вебзастосунку використано середовище розробки Eclipse та мову програмування Java, Spring Framework, Javascript.

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

Програмне забезпечення повинно бути доступним для користувача завжди, і мати можливість працювати безперервно.

Практичність:

- вебзастосунок має бути простим і зручним у використанні;
- дизайн вебзастосунку має бути адаптивним та зрозумілим для використання;
- вебзастосунок має бути розроблено за допомогою плоскої архітектури.

Надійність:

- зашифрована інформація про користувача в базі даних;
- програмна обробка можливих помилок;
- стабільність роботи при нестандартних умовах.

Переносимість

Вебзастосунок повинен працювати на декстоп та мобільних пристроях в будь-якому браузері.

Висновки до розділу 1

У результаті проведеного аналізу визначено, що створення системи вебзастосунку Інтернет-магазину товарів для косметологів є актуальним, тому що на даному етапі немає українських сайтів, котрі використовують PayPal, оскільки доступ для України було недавно надано.

Проведено дослідження аналогічних мобільних застосунків та визначено необхідність створення даного програмного застосунку. На основі проведеного дослідження предметної області сформовано вимоги до програмного забезпечення, що розроблюється.

2 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ

2.1 Сценарії використання системи

В даному розділі буде проведено ознайомлення зі сценаріями використання системи. Для цього буде розроблено діаграму варіантів використання та буде описано варіанти використання системи за допомогою сценарної техніки опису взаємодії Use Case. Необхідність розробки Use Case діаграми, або як її ще називають діаграмою прецендентів описана у книзі Алістера Коберна [7]. Діаграма прецедентів фіксує угоду між учасниками системи щодо її поведінки. Також вона описує поведінку системи при її відповідях на запит одного з учасників, що називається основною дійовою особою, у різних умовах.

Для створення діаграми обрано інструмент для розробки UML-діаграм MiroUml від розробника Miro [6].

Система містить три дійові особи – клієнта, менеджера та адміністратора. Кожен має свою роль та функції на сайті. Діаграму варіантів використання можна побачити на рис. 2.1. На ній зображено дійові особи та доступні їм функції вебзастосунку. Клієнт має три головні поведінки. Він може передивлятися каталог, управляти акаунтом та робити замовлення. Ці поведінки розширюють або включають у себе інші поведінки. Менеджер може управляти заказом, каталогом, консультувати клієнта. Адміністратор може додавати новини та управляти акаунтами.

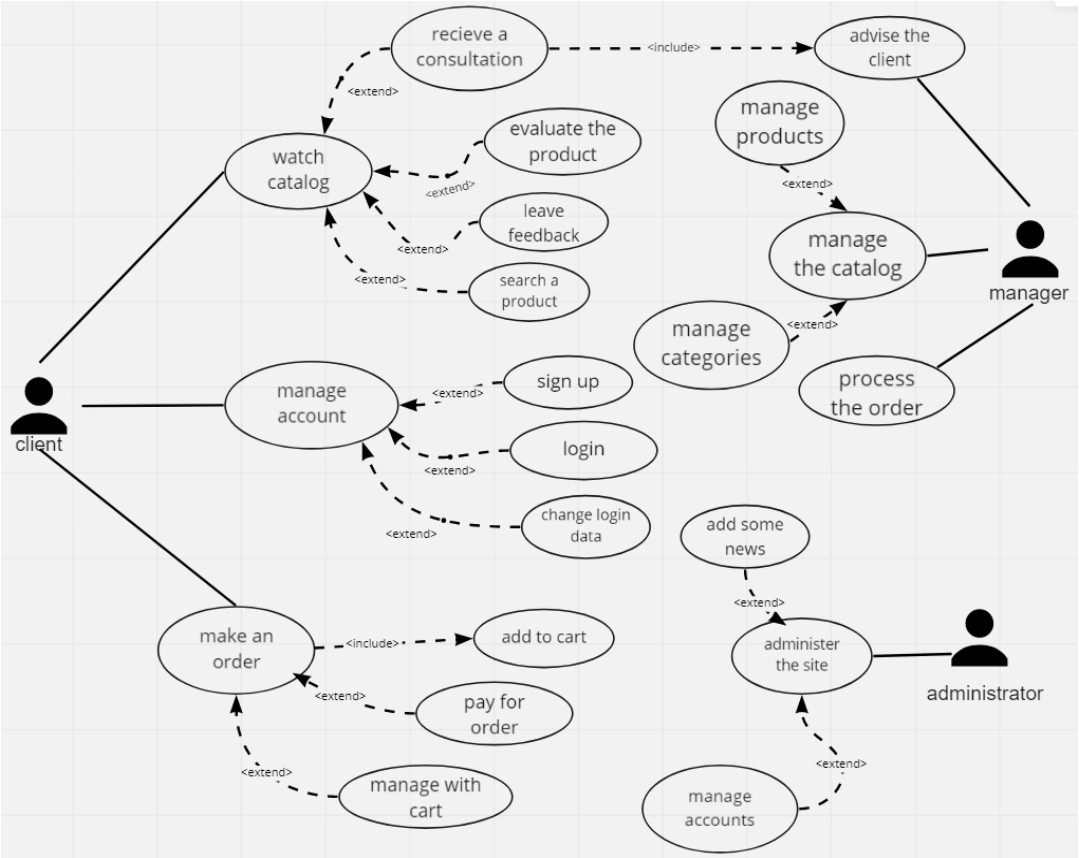

Рисунок 2.1 – Діаграма варіантів використання

Задля повного розуміння деяких функцій вебзастосунку необхідно розробити варіанти використання цих функцій. Найважливіші варіанти використання будуть описані в повній формі за допомогою таблиці. Менш значні варіанти використання будуть описані в поверхневій формі.

У таблиці 2.1. можна побачити повну форму варіанту використання покупки товару у вебзастосунку.

Таблиця 2.1 – Варіант використання покупки товару у вебзастосунку

Use Case Name	покупка товару
Scope	вебзастосунок продажу товарів
Level	User-goal
Primary Actor	Користувач, Менеджер

Кінець таблиці 2.1

Stakeholders and interests	1. Клієнт: зацікавлений в швидкому придбанні товарів (отриманні послуг) та зручному відображенні розрахунків суми покупки. 2. Компанія (продавець): зацікавлена в точній обробці транзакцій при покупці товарів та задоволенні інтересів користувача. 3. Державна податкова адміністрація: зацікавлена в отриманні податків від кожного товару. 4. Менеджер: зацікавлений у швидкому розв’язанні проблем під час повернення товарів та легкій перевірці створених заказів.
Precondition	Клієнт перейшов на сайт застосунку Менеджер пройшов авторизацію у вебзастосунку.
Success guarantee	Клієнт обирає товар, сплачує його та отримує.
Main Success Scenario	1. Клієнт обирає товар, який він хоче придбати. 2. Клієнт додає цей товар у кошик 3. Клієнт переходить до кошика та обирає функцію оформити товар. 4. Клієнт вводить вірно усі контактні дані та дані по оплаті товару. 5. Система створює замовлення.

	6. Менеджер дзвонить клієнту для уточнення замовлення. 7. Клієнт підтверджує замовлення. 8. Менеджер відправляє замовлення поштою. 9. Клієнт отримує замовлення.

Кінець таблиці 2.1

Extensions	1. Клієнт оформлює замовлення невірно. 2. Клієнт невірно заповнює контактні дані. 3. Замовлення не створюється. 4. Клієнт хоче оплатити карткою онлайн. 5. В клієнта не вистачає грошей на картці. 6. Заказ не створюється. 7. В клієнта стоїть невірний інтернет-ліміт. 8. Заказ не створюється. 9. В клієнта закінчився срок дії картки 10. Заказ не створюється. 11. Клієнт обрав пункт не дзвонити для уточнення заказу. 12. Менеджер оформлює заказ без

	дзвінка клієнту
Special Requirements	1. Пристрій із можливістю доступу до мережі інтернет. 2. Мережа інтернет.
Frequency of Occurrence	Система може працювати безперервно.

Для функції логіну до існуючого акаунту використана поверхева форма варіантів використання, і вона наведена нижче.

Головний сценарій (успішний):

Користувач переходить на сайт. Переходить у розділ входу на акаунт. Обирає тип входу. Вводить правильний логін, пароль від його акаунту та Re-captcha. Виконується успішний вхід на акаунт користувача.

Альтернативні сценарії:

1) Користувач не оплатив провайдеру за мережу інтернет та в нього помилка “немає з’єднання”.

2) Користувач вводить невірні дані входу в акаунт.

3) Технічний збій роботи системи. Видається повідомлення „немає зв’язку з сервером баз даних”.

4) Користувач не проходе Re-captcha.

Сценарій: Додавання у корзину

Користувач переходить на сайт. Обирає товар який хоче придбати. Переходить на сторінку цього товару. Користувач додає в корзину цей товар. Користувач покидає сторінку товару та находить ще один товар у цьому розділі. Додає щє один товар до корзини. Користувач перевіряє корзину, на наявність доданих товарів.

Альтернативні сценарії:

1) Користувач додав товар не у кошик, а у мої закладки.
2) Користувач додав не тої товар.

2.2 Моделювання взаємодії між об’єктами

Взаємодію між об’єктами можно показати зявдяки діаграмі послідовності. Вона відображає взаємодії об’єктів, впорядкованих за часом. У проєктуванні вебзастосунку діаграми послідовностей використовуються для уточнення діаграм прецедентів, які були наведені у попередньому підрозділі, та більш детального опису логіки сценаріїв використання. Діаграми послідовностей зазвичай містять об'єкти, які взаємодіють у рамках сценарію, повідомлення, якими вони обмінюються, і результати, пов’язані з повідомленнями, що повертаються.

На рис. 2.2 представлена діаграма послідовності реєстрації. Об’єктами є клієнтська частина, клієнт-серверна частина та база даних. Об'єкти позначаються овалами із іменами (щоб відрізнити їх від класів). Повідомлення (виклики методів) – лініями зі стрілками. Результати, що повертаються – пунктирними лініями зі стрілками. Прямокутники на вертикальних лініях під кожним об’єктом показують “час життя” (фокус) об'єктів [8]. Користувач переходить за посиланням на сторінку реєстрації, після чого браузер відправляє запит на сервер, а сервер повертає сторінку. Користувач заповнює поля, які перенесуть інформацію з них до бази даних після реєстрації, та додає фото, після цього браузер передає дані запитом на сервер. Далі є декілька можливостей перебігу подій: якщо дані вірні, сервер звертається до бази даних та записує дані в неї, якщо дані не вірні, сторінка перезавантажується і виводить помилку про некоректність введення даних або про поля які не були заповнені.

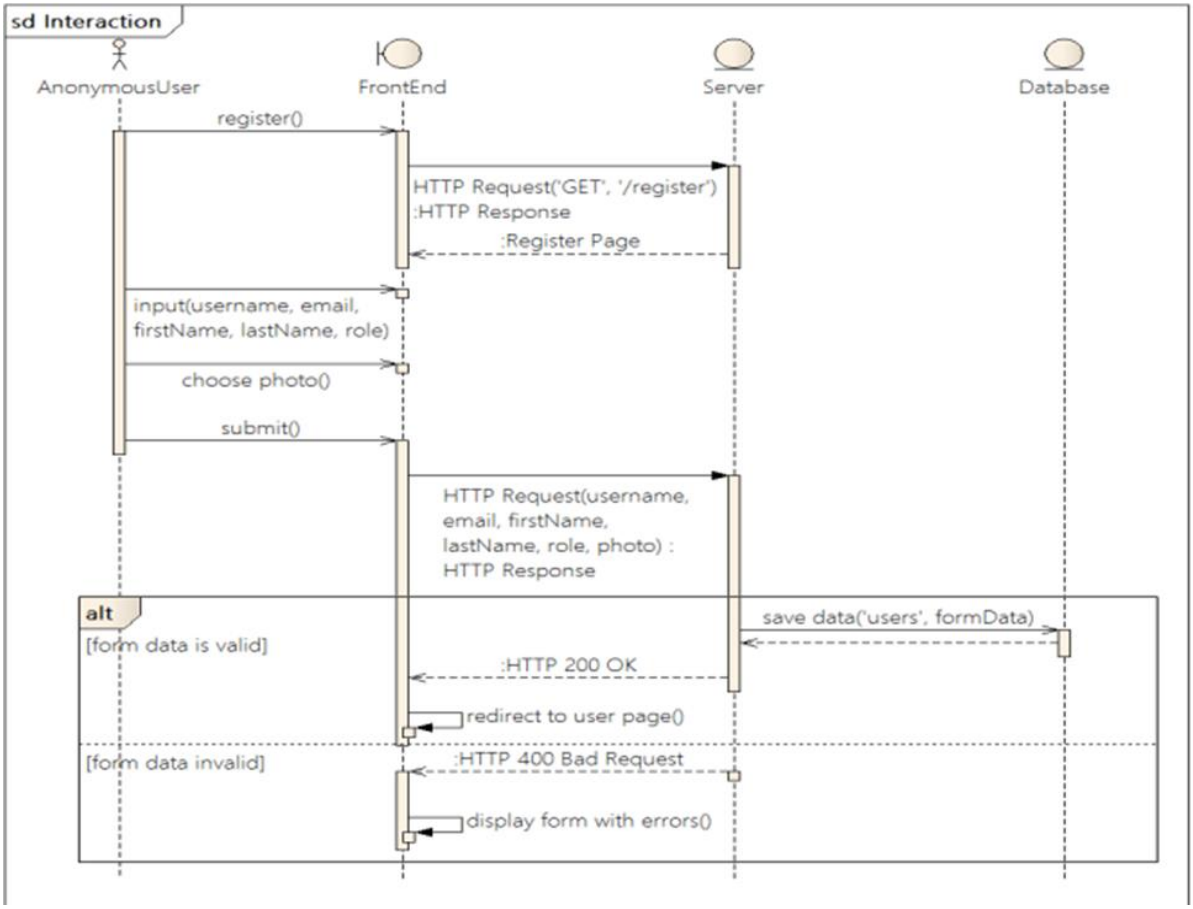

Рисунок 2.2 – Діаграма варіантів послідовності реєєстрації

Пояснення до діаграми послідовності для варіанту використання реєстрації нового користувача:

1. register(): Користувач переходить на сторінку реєєстрації;
2. HTTP Request(“GET”, “/register)”: Користувач посилає запит GET на сервер для того, щоб перевірити чи не зареєстрований користувач;
3. HTTP Response Register Page: сервер повертає сторінку із реєстрацією;
4. input (email, username, firstName, lastName, role): користувач вводить свої дані;
5. choose photo(): користувач має можливість вибрати фото профілю;
6. submit(): користувач посилає на клієнтській системі дані;

7. HTTP Request(email, username, firstName, lastName,role): з клієнтської частини посилається на серверну через запит POST;

8. save data(“users”, fromData): Сервер відправляє дані у базу даних;

9. [form data is valid]: користувач відправив коректні дані;

10. HTTP 200 ok: запит пройшов успішно. Сервер відповів без помилок;

11. [form data is invalid]: користувач відправив некоректні дані;

12. HTTP 400 Bad Request: запит не пройшов успішно. Сервер відповів помилкою.

На рис. 2.3 представлена діаграма послідовності авторизації. Користувач заповнює дані, попередньо зайшовши на сторінку вебзастосунку, після цього браузер відправляє дані на сервер, який перевіряє їх з базою даних. Якщо дані вірні сервер передає сторінку користувача, інакше сторінку з помилкою.

Пояснення до діаграми послідовності для варіанту використання реєстрації нового користувача:

1. enter site(): користувач переходить на сторінку вебзастосунку;

2. HTTP Request(“GET”, “/)”: відправляється GET-запит на сервер;

3. HTTP Response: сервер повертає сторінку із авторизацією;

4. input(email or username, password): користувач вводить свої дані;

5. submit login(): користувач відправляє дані;

6. HTTP Request(“POST”, “/login), (username, password)”: клієнтська частина відправляє дані на сервер;

7. user = get data (username, password): сервер порівнює дані, які ввів користувач із даними із бази даних;

8. HTTP 200 ok: запит пройшов успішно. Сервер відповів без помилок;

9. [user.hash = hash(password)]: дані, введені користувачем, співпали із даними, які зберігаються у базі даних;

10. [else]: інші випадки;

11. HTTP 401 Unathorized: запрос не пройшов успішно. Сервер відповів помилкою.

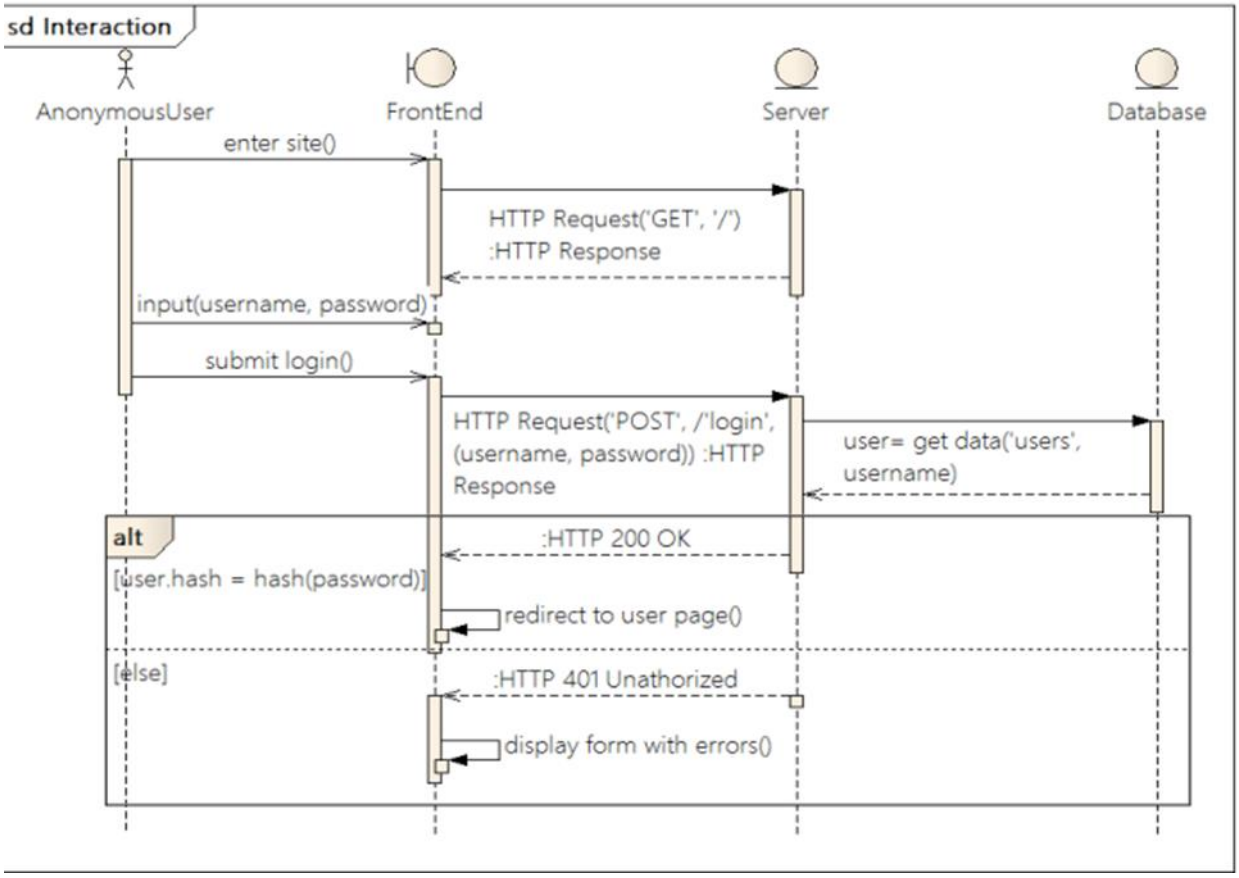

Рисунок 2.3 – Діаграма варіантів послідовності авторизації

По структурі Інтернет-магазин має 2 частини: клієнтська та адміністративна. Кожна із цих частин має свої функції. На рис. 2.4 представлена структурна схема магазину, на якій показані адміністративні та клієнтські частини вебзастоснку, а саме функції які можуть виповнювати ці частини.

Рисунок 2.4 – Структурна схема сайту

До адміністративної частини має доступ тільки адміністратор та менеджер. Вхід до адміністративної частини здійснюється завдяки переходу за спеціальною адресою та з IP-адресою, яку має та знає відповідно, тільки менеджер та адміністратор До клієнтської частини мають доступ усі користувачі, як зареєстровані так і анонімні.

2.3 Діаграма класів

Діаграма класів має провідну роль у побудові архітектури проекту. Вона визначає типи класів системи та різного роду зв'язки між ними. Також зображуються атрибути класів, операції класів та обмеження, що накладаються на зв'язки між класами. Вигляд та інтерпретація діаграми класів істотно залежить від точки зору (рівня абстракції): класи можуть представляти сутності предметної галузі (у процесі аналізу) або елементи програмної системи (у процесах проектування та реалізації)[9].

Вебзастосунок, що створюється, складається із 7 класів. Діаграму класів інтернет магазину товарів для косметологів можна побачити на рис. 2.5.

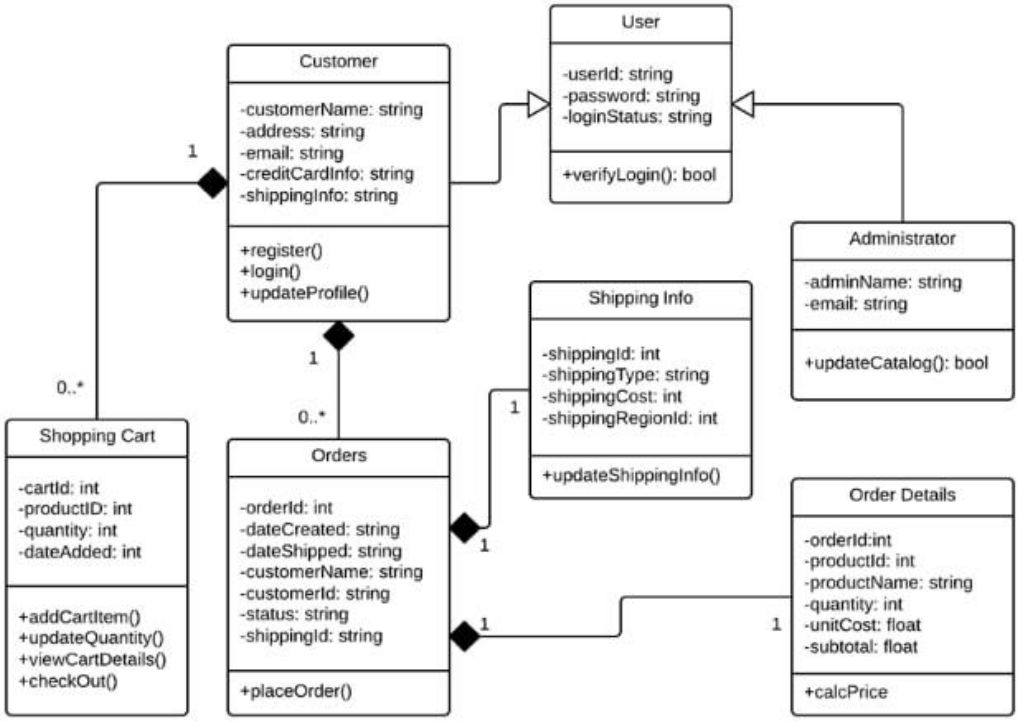

Рисунок 2.5 – Діаграму класів інтернет магазину

Пояснення до діаграми класів клієнтської частини застосунку. Так як клієнтська частина застосунка буде побудована на фреймворку ReactJs, який в останніх оновленнях не використовує класові компоненти, то класи будуть умовні. Замість класів фреймворк використовує функціїні компоненти, які по суті не мають великих відмінностей для діаграми. Тому суть діаграми класів це не змінює.

Опис компонента Shoping Cart:

Компонент Shoping Cart відповідає за товари у кошику.

Атрибути:

− CartId – ідентифікаційний номер для корзини;

− ProductId – ідентифікаційний номер для продукту;

- Quantity – кількість продукту;
- dateAdded – дата коли було додано товар у кошик;
- addCartItem() – метод додавання товару у кошик;
- updateQuanity() – метод зміни кількості товару;
- viewCartDetails() – метод перегляду кошика;
- checkOut() - метод для переходу до оплати товару.

Опис компонента Customer:

Компонент Customer відповідає за клієнта.

Атрибути:

- customerName – ім’я клієнта;
- adress – адреса клієнта;
- email – електрона адреса клієнта;
- creditCardInfo – інформація про кредитну картку;
- shipingInfo – інформація про доставку;
- register() – метод реєстрації;
- login() – метод авторизації;
- updateProfile() – метод оновлення профілю.

Опис компонента Orders:

Компонент Orders відповідає за замовлення.

Атрибути:

- OrderId – ідентифікаційний номер замовлення;
- dateCreated – дата створення заказу;
- dateShiped – дата доставки заказу;
- customerName – ім’я замовника;
- customerId – ідентифікаційний номер користувача;
- status – статус замовлення;
- shipingId – ідентифікаційний номер замовлення;
- placeOrder() – метод створення заказу.

Опис компонента User:

Компонент User відповідає користувача.

Атрибути:

- UsertId – ідентифікаційний номер для користувача;
- Password – пароль користувача;
- loginStatus – статус входу;
- verifyLogin() – перевірка на реєстрацію.

Опис компонента Shiping Info:

Компонент Shiping Info відповідає за доставку товару.

Атрибути:

- shipingId– ідентифікаційний номер заказу;
- shipingCost – ціна товару;
- shipingType – тип доставки;
- shipingRegionID – регіон доставки;
- updateShipingInfo() – оновити інформацію про доставку.

Опис компонента Administrator:

Компонент Administrator відповідає за панель адміністратора.

Атрибути:

- adminName – ім’я адміністратора;
- email – електрона пошта адміністратора;
- updateCatalog – оновлення каталогу.

Опис компонента Order Details:

Компонент Order Details відповідає за деталі замовлення.

Атрибути:

- orderId – ідентифікаційний номер замовлення;
- productID – ідентифікаційний номер товару;
- productName – назва товару;
- Quantity – кількість товару;

− unitCost – ціна товару;

− subtotal – вартість товару;

− calcPrice() – метод по разрахунку вартості то.

2.4 Моделювання прототипу користувацького інтерфейсу

Моделювання прототипу користувацького інтерфейсу є важливим єтапом проектування вебзастосунку. Саме через це між замовниками та виконавцями буває більш всього розбіжностей. Інтерфейс повинен бути зручним та функціональним. Також він повинен відповідати усім сучасним вимогам. Користувач не буде затримуватись довго на сайті на якому він нічого не розуміє. Для кращої розробки користувацького інтерфейсу в 1 розділі було розроблено порівняльний аналіз аналогів, де було підкреслено позитивні та негативні сторони інтерфейсу інтернет-магазину товарів для косметологів. Прототип головної сторінки та за сумісністю каталогу товарів можна побачити на рис. 2.6.

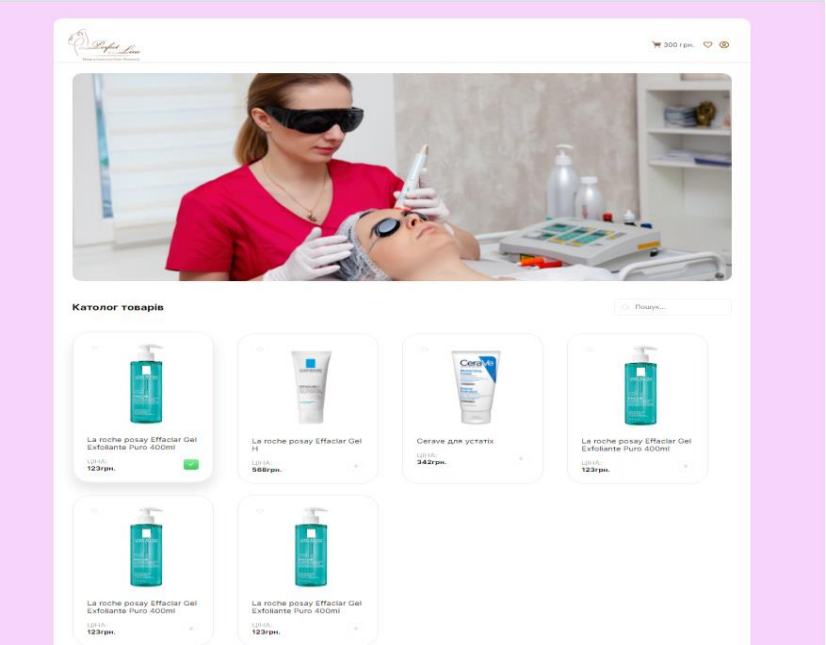

Рисунок 2.6 – Макет каталогу товарів

В каталозі немає нічого зайвого. В ньому є шапка, слайдер з акційними пропозиціями, та сам каталог товарів. При наведені на товар з’являється

ефект наведення та змінюється курсор. Можно перейти на сторінку товару або додати його. Цей макет дуже легко адаптується. Після того, як товар додали, він потрпляє у кошик. Макет кошику можно побачити на рис. 2.7.

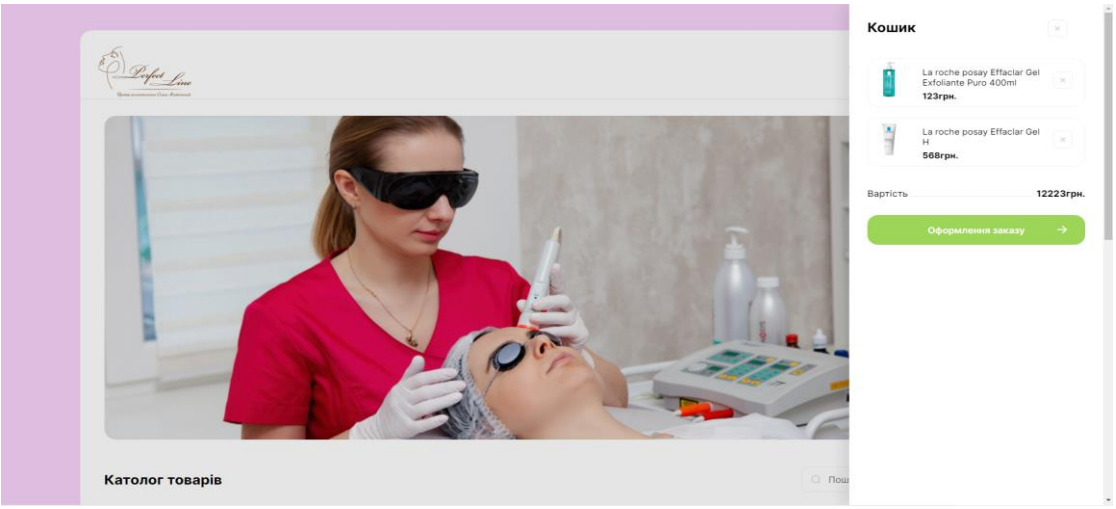

Рисунок 2.7 – Макет кошику

Для адміністрування сайту було розроблено окремий прототип, так як функціонал адміністративної частини відрізняється. Макет логіну в адміністративну панель зображено на рис. 2.8.

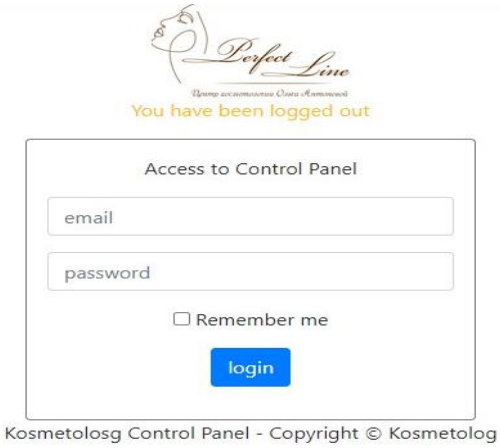

Рисунок 2.8 – Макет логіну в адміністративну панель

Так як адміністративна частина потребує набагато більшого функціоналу, тому вона більш наповнена. Також саму адміністративну панель можно побачити на рис. 2.9.

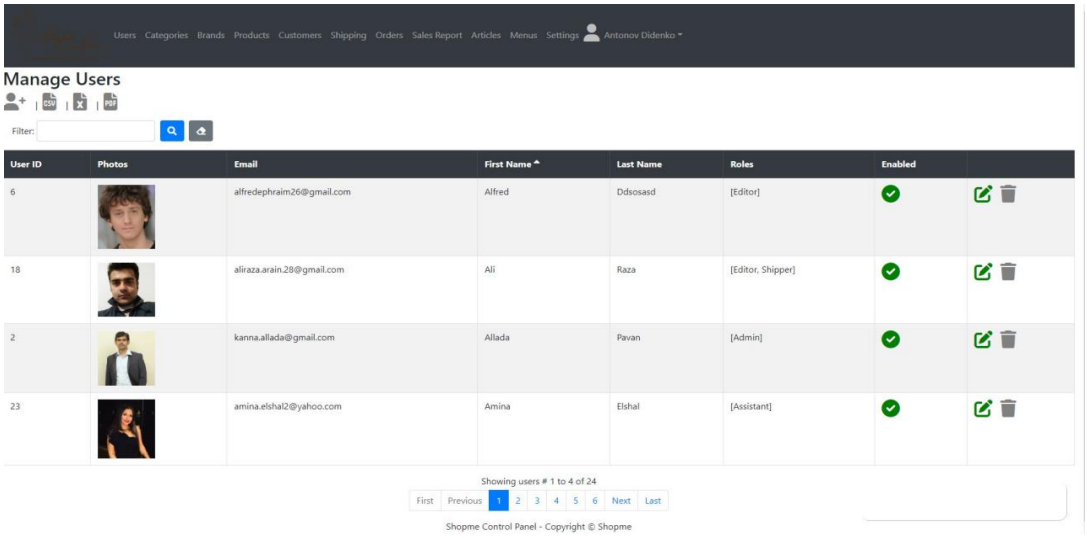

Рисунок 2.9 – Макет адміністративної панелі

На макеті адміністративної панелі зображено одну із багатьох сторінок клієнтів інтернет магазину.

Висновки до розділу 2

У другому розділі комплексної кваліфікаційної роботи бакалавра розглянуто такі етапи розробки вебзастосунку як архітектура, моделювання та проєктування. В результаті моделювання вебзастосунку спочатку розроблено діаграму варіантів використання із створенням повних та поверхневих форм UseCase.

Під час виконання роботи змодельовано взаємодії між об’єктами вебзастосунку. Було побудовано діаграми послідовності реєстрації та авторизації, в яких об’єктами була серверна, клієнт-серверна частина та база даних. Також було розроблено та описано діаграму класів інтернет-магазину товарів для косметологів.

В кінці розділу було розглянуто моделювання прототипів користувацького інтерфейсу. В ньому було побудовано макети головної сторінки, кошику та адміністративної панелі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Інформація про сайт Alvi Prague. URL: https://alvi-prague.ua/ (дата звернення: 11.05.2022).

2. Інформація про сайт Piel cosmetics. URL: https://pielcosmetics.com/ (дата звернення: 11.05.2022).

3. Інформація про сайт Zemits: Shallow Waters. URL: https://zemits.com/ (дата звернення: 11.05.2022).

4. Інформація про архітектуру вебзастосунка. URL: https://hostiq.ua/blog/15-site-architecture-lifehacks/ (дата звернення: 11.05.2022) .

5. Інформація про електрону комерцію. URL: https://www.the-future-of-commerce.com (дата звернення: 11.05.2022).

6. Вебзастосунок для створення UML діаграм: https://miro.com/ (дата звернення: 01.06.2022) .

7. Алистер Коберн.: Сучасні методи опису функціональних вимог до систем. 2012. 264 с. URL: https://www.livelib.ru/book/1000823328-sovremennye-metody-opisaniya-funktsionalnyh-trebovanij-k-sistemam-alister-kobern

8. Інформація про діаграму варіантів послідовності: https://flexberry.github.io/ru/fd_sequence-diagram.html (дата звернення: 07.06.2022).

9. Інформація про діаграму класів: https://flexberry.github.io/ru/gpg_class-diagram.html (дата звернення: 07.06.2022).

10. Інформація про SASS: https://tokar.ua/read/6672 (дата звернення: 07.06.2022).

11. Інформація про Thymeleaf: https://www.thymeleaf.org/ (дата звернення: 07.06.2022).

12. Інформація про JavaScript: https://developer.mozilla.org/ru/docs/Web/JavaScript/About_JavaScript (дата звернення: 07.06.2022).

13. Інформація про Bootstrap: https://getbootstrap.com/ (дата звернення: 07.06.2022).

14. Інформація про MySQL Workbench. URL: https://www.mysql.com/products/workbench/ (дата звернення: 02.06.2022).

15. Інформація про Spring Security. URL: https://spring.io/projects/spring-securit (дата звернення: 03.06.2022).

16. Інформація про Spring Data JPA. URL: https://uk.helpr.me/8227-what-is-repositoryrestresource (дата звернення: 03.06.2022).

17. M. Macero García. Learn Microservices with Spring Boot. Apress, Berkeley, CA, 2020. P. 25-74. URL: https://doi.org/10.1007/978-1-4842-6131-6_3 (дата звернення 20.05.2022).

18. F. Gutierrez. Pro Spring Boot 2. Apress, Berkeley, CA, 2019. P. 87-125. URL: https://doi.org/10.1007/978-1-4842-3676-5_4 (дата звернення 20.05.2022).

19. P. Reddy, K. Siva, Beginning Spring Boot 2. Apress, Berkeley, CA, 2017. P. 289-300. URL: https://doi.org/10.1007/978-1-4842-2931-6_19 (дата звернення 20.05.2022).

20. P. Reddy, K. Siva, Beginning Spring Boot 2. Apress, Berkeley, CA, 2017. P. 107-132. URL: https://doi.org/10.1007/978-1-4842-2931-6_10 (дата звернення 20.05.2022).

21. F. Gutierrez. Pro Spring Boot 2. Apress, Berkeley, CA, 2019. P. 289-300. URL: https://doi.org/10.1007/978-1-4842-3676-5_10 (дата звернення 20.05.2022).