

Державний торговельно-економічний університет  
Кафедра інженерії програмного забезпечення та кібербезпеки

**ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ**  
на тему:

**«Програмний додаток торговельного підприємства з реалізації  
спортивного харчування»**

Студента 4 курсу, б групи,  
спеціальності 121 «Інженерія  
програмного забезпечення»  
освітньої програми  
«Інженерія програмного  
забезпечення»

Долі Олексія  
Сергійовича

\_\_\_\_\_

підпис студента

Науковий керівник  
Кандидат технічних наук,  
доцент кафедри інженерії  
програмного забезпечення  
та кібербезпеки

Власенко Лідія  
Олександрівна

\_\_\_\_\_

підпис керівника

Гарант освітньої програми  
кандидат технічних наук,  
доцент кафедри інженерії  
програмного забезпечення  
та кібербезпеки

Рзасва Світлана  
Леонідівна

\_\_\_\_\_

підпис гаранта

# Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

## Затверджую

Зав. кафедри інженерії програмного  
забезпечення та кібербезпеки

Криворучко О. В.

«14» листопада 2022 р.

## Завдання

**на випускний кваліфікаційний проєкт студентіві**

Долі Олексія Сергійовича

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Програмний додаток  
торговельного підприємства з реалізації спортивного харчування»

Затверджена наказом ректора від «6» грудня 2022 р. № 3288

2. Строк здачі студентом закінченого проєкту 5 червня 2023

3. Цільова установка та вихідні дані до проєкту

Мета проєкту – дослідити тему про розробку додатка для компанії яка  
займається продажем спортивного харчування

Об'єкт дослідження – торговельне підприємство з реалізації спортивного  
харчування

Предмет дослідження – розробка програмного додатка для функціонування  
торговельного підприємства з реалізації спортивного харчування

4. Консультанти проєкту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проєкту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз ринку спортивного харчування

1.2. Огляд програмного додатка торговельного підприємства

1.3. Висновки до розділу 1

РОЗДІЛ 2. ПРОЄКТУВАННЯ ПРОГРАМНОГО ДОДАТКА ТОРГОВЕЛЬНОЇ ДІЯЛЬНОСТІ

2.1. Вимоги до програмного додатка

2.2. Архітектура програмного додатка

2.3. Розробка бази даних

2.4. Розробка інтерфейсу користувача

2.5. Висновки до розділу 2

РОЗДІЛ 3. ТЕСТУВАННЯ ПРОГРАМНОГО ДОДАТКА

3.1. Опис методів тестування програмного додатка

3.2. Тестування розробленого додатка

3.3. Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2022	21.09.2022
2.	<i>Розробка та затвердження завдання на проєкт</i>	14.11.2022	14.11.2022
3.	<i>Вступ та перелік літературних джерел</i>	23.12.2022	23.12.2022
4.	<i>Розділ 1. Аналіз предметної області та постановка технічного завдання</i>	27.01.2023	27.01.2023
5.	<i>Розділ 2. Проектування програмного додатка</i>	03.03.2023	03.03.2023
6.	<i>Розділ 3. Процес розробки програмного додатка</i>	14.04.2023	14.04.2023
7.	<i>Висновки</i>	28.04.2023	28.04.2023
8.	<i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	17.05.2023	17.05.2023
9.	<i>Підготовка автореферату та презентації доповіді</i>	26.05.2023	26.05.2023
10.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	29.05.2023 – 02.06.2023	31.05.2023
11.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	05.06.2023	05.06.2023
12.	<i>Здача прошого випускного кваліфікаційного проєкту на кафедрі</i>	05.06.2023	05.06.2023
13.	<i>Публічний захист випускного кваліфікаційного проєкту</i>	19.06.2023	19.06.2023

7. Дата видачі завдання «14» листопада 2022 р.

8. Науковий керівник випускного кваліфікаційного проєкту Власенко Л.О.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Рзаєва С.Л.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Доля О.С.

(прізвище, ініціали, підпис)

**11. Відгук керівника випускного кваліфікаційного проєкту**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Науковий керівник випускного кваліфікаційного проєкту

\_\_\_\_\_ (підпис, дата)

Відмітка про попередній захист \_\_\_\_\_ (ПІБ, підпис, дата)

**12. Висновок про випускний кваліфікаційний проєкт**

Випускний кваліфікаційний проєкт студента \_\_\_\_\_ Доля О.С.  
(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми \_\_\_\_\_ Рзаєва С.Л.  
(прізвище, ініціали, підпис)

Завідувач кафедри \_\_\_\_\_ Криворучко О. В.  
(підпис, прізвище, ініціали)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ р.

## АНОТАЦІЯ

Даний дипломний проект присвячений розробці програмного додатка для торгової компанії, що спеціалізується на продажі спортивного харчування. Метою проекту є створення зручного та ефективного інструменту для автоматизації процесів управління продажами та складського обліку.

Основні функції програмного додатка включають в себе: ведення інформації про клієнтів, товари та замовлення, генерацію звітів про продажі, стан складу, а також інші функції, необхідні для ефективного управління бізнесом.

**Ключові слова:** програмний додаток, торгова компанія, спортивне харчування, управління продажами, складський облік, звіти, клієнти, замовлення.

## ABSTRACT

This diploma project is dedicated to the development of software for a trading company specializing in the sale of sports nutrition. The aim of the project is to create a convenient and effective tool for automating sales management and warehouse accounting processes.

The main functions of the software include: maintaining information about customers, products and orders, generating reports on sales, warehouse status, as well as other functions necessary for effective business management.

**Keywords:** software, trading company, sports nutrition, sales management, warehouse accounting, reports, customers, orders.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

XML (англ. Extensible Markup Language) – розширювана мова розмітки



					<i>ДТЕУ 121 06-10.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Програмний додаток підприємства з реалізації спортивного харчування	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.	Криворучко О.В.			14.04.23		ПС	2	50
Керівник	Власенко Л.О.			14.04.23		Факультет інформаційних технологій 4 курс, 6 група		
Гарант	Рзаєва С.Л.			14.04.23				
Розробив	Доля О.С.			14.04.23				
					<i>Перелік умовних скорочень</i>			

## ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1. Аналіз ринку спортивного харчування.....	6
1.2. Огляд програмного додатка торговельного підприємства.....	12
1.3. Висновки до розділу 1.....	17
РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ДОДАТКА ТОРГОВЕЛЬНОЇ ДІЯЛЬНОСТІ.....	18
2.1. Вимоги до програмного додатка.....	18
2.2. Архітектура програмного додатка.....	19
2.3. Розробка бази даних.....	21
2.4. Розробка інтерфейсу користувача.....	23
2.5. Висновки до розділу 2.....	31
РОЗДІЛ 3. ТЕСТУВАННЯ ПРОГРАМНОГО ДОДАТКА.....	33
3.1. Опис методів тестування програмного додатка.....	33
3.2. Тестування розробленого додатка.....	35
3.3. Висновки до розділу 3.....	36
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	40
ДОДАТКИ.....	42

					<i>ДТЕУ 121 06-10.БР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	Програмний додаток підприємства з реалізації спортивного харчування	Стадія	Аркуш	Аркушів
Зав. каф.		Криворучко О.В.		23.12.22		3	3	50
Керівник		Власенко Л.О.		23.12.22		Факультет інформаційних технологій 4 курс, 6 група		
Гарант		Рзаєва С.Л.		23.12.22				
Розробив		Доля О.С.		23.12.22				
					Зміст			



## ВСТУП

Харчова промисловість завжди була життєво важливим сектором економіки будь-якої країни. Вона відіграє вирішальну роль у забезпеченні населення продуктами харчування та створює можливості для працевлаштування багатьох людей. Зі зростанням популярності фітнесу та спорту значно збільшився попит на продукти спортивного харчування. Ринок продуктів спортивного харчування неухильно зростає протягом багатьох років, і очікується, що в найближчі роки він зростатиме ще більше.

Ринок спортивного харчування є прибутковим сектором для підприємців, які хочуть розпочати свій бізнес. Однак ведення успішного бізнесу зі спортивного харчування може бути складним завданням, особливо якщо підприємцю бракує необхідних ресурсів для ефективного управління бізнесом. В останні роки програмне забезпечення відіграє життєво важливу роль в управлінні бізнесом у різних галузях. Тому використання програмного забезпечення в управлінні бізнесом спортивного харчування може забезпечити конкурентну перевагу бізнесу.

Метою цього проекту є розробка програмного додатка, яке допоможе підприємцям ефективно управляти своїм бізнесом спортивного харчування. Програмний додаток буде розроблено з урахуванням специфічних потреб бізнесу спортивного харчування, таких як управління запасами, управління продажами та управління взаємовідносинами з клієнтами.

Цей проект розпочнеться з аналізу ринку спортивного харчування, включаючи поточний стан ринку, попит на продукти спортивного харчування

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 06-10.БР			
Зав. каф.	Криворучко О.В.			23.12.22	Програмний додаток підприємства з реалізації спортивного харчування	Стадія	Аркуш	Аркушів
Керівник	Власенко Л.О.			23.12.22		В	4	50
Гарант	Рзасва С.Л.			23.12.22		Факультет інформаційних технологій		
Розробив	Доля О.С.			23.12.22		4 курс, 6 група		
					Вступ			

та аналіз конкурентів на ринку. Наступним розділом буде огляд програмного додатка, що використовується в управлінні бізнесом, включаючи класифікацію та опис сучасного програмного забезпечення, що використовується в різних галузях.

Основна увага в цьому проекті буде зосереджена на розробці програмного додатка для підприємств, що займаються виробництвом спортивного харчування. У розділі будуть описані вимоги до програмного додатка, архітектура програмного додатка, дизайн бази даних та дизайн користувацького інтерфейсу. Буде детально описана функціональність програмного додатка, включаючи управління запасами, управління продажами та управління взаємовідносинами з клієнтами.

Програмний додаток буде ретельно протестований для забезпечення його функціональності та надійності. Фаза тестування включатиме різні методи, такі як тестування "чорної скриньки" та тестування на прийнятність для користувача. Результати етапу тестування будуть представлені в цьому проекті.

Нарешті, в цьому проекті будуть представлені висновки, засновані на результатах аналізу та розробки програмного додатка. Проект надасть рекомендації щодо використання програмного додатка для управління бізнесом спортивного харчування, а також потенційні переваги використання програмного додатка в управлінні бізнесом в цілому

Таким чином, цей проект спрямований на розробку програмного додатка, яке допоможе підприємцям ефективно управляти своїм бізнесом у сфері спортивного харчування. Програмний додаток буде розроблене з урахуванням специфічних потреб бізнесу спортивного харчування і дозволить підприємцям приймати обґрунтовані рішення на основі точних даних.

						ДТЕУ 121 06-10.БР	Аркуш
							5
Зм.	Аркуш	№ докум	Підпис	Дата			

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

#### 1.1. Аналіз ринку спортивного харчування

Ринок спортивного харчування зазнав значного зростання в останні роки завдяки зростанню обізнаності про переваги здорового способу життя та регулярної фізичної активності. Індустрія спортивного харчування в першу чергу зосереджена на продуктах, які забезпечують поживними речовинами спортсменів, бодібілдерів та любителів фітнесу для покращення їхніх спортивних результатів, росту м'язів та загального стану здоров'я.

##### 1.1.1. Огляд сучасного стану ринку спортивного харчування

Світовий ринок спортивного харчування у 2020 році оцінювався у 17,8 млрд доларів США і, як очікується, досягне 26,7 млрд доларів США до 2027 року, а його середньорічний темп зростання становитиме 5,8% протягом прогнозованого періоду. Зростання ринку в першу чергу зумовлене збільшенням попиту на продукти спортивного харчування та зростаючою тенденцією до здоров'я та фітнесу серед населення.

Північна Америка домінує на ринку спортивного харчування, на її частку припадає понад 35% світового ринку. Сполучені Штати є найбільшим ринком спортивного харчування в Північній Америці з часткою ринку понад 70%. Європа є другим за величиною ринком спортивного харчування з часткою ринку понад 30%. В Азіатсько-Тихоокеанському регіоні також очікується значне зростання ринку спортивного харчування завдяки зростаючій популярності здорового способу життя та збільшенню наявного доходу.

Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		27.01.23	Програмний додаток підприємства з реалізації спортивного харчування	Стадія	Аркуш	Аркушів
Керівник		Власенко Л.О.		27.01.23		РІ	6	50
Гарант		Рзаєва С.Л.		27.01.23		Факультет інформаційних технологій		
Розробив		Доля О.С.		27.01.23		4 курс, 6 група		
					Аналіз предметної області			

ДТЕУ 121 06-10.БР

Ринок спортивного харчування сегментований на різні типи продуктів, включаючи протеїновий порошок, протеїнові батончики, спортивні напої, спортивні гелі та інші добавки. Протеїновий порошок є найбільшим сегментом ринку спортивного харчування з часткою ринку понад 50%. Зростаюча популярність білкових добавок на рослинній основі та тенденція веганства сприяють зростанню сегменту протеїнового порошку.

Ринок спортивного харчування також сегментований на основі каналу дистрибуції, включаючи онлайн та офлайн продажі. Онлайн-продажі зростають швидшими темпами, ніж офлайн-продажі, завдяки зростаючій популярності електронної комерції та зручності онлайн-покупок. Пандемія COVID-19 ще більше прискорила зростання онлайн-продажів, оскільки все більше людей вважають за краще робити покупки онлайн, щоб уникнути фізичного контакту.

Ринок спортивного харчування є висококонкурентним, на ньому присутні кілька ключових гравців, серед яких Glanbia plc, Abbott Laboratories, GNC Holdings Inc, The Coca-Cola Company, PepsiCo Inc. та інші. Ці компанії інвестують у дослідження та розробки для впровадження інноваційних продуктів, щоб задовольнити мінливі споживчі вподобання. Вони також зосереджуються на злиттях і поглинаннях, щоб розширити свою присутність на ринку і збільшити частку ринку.

Таким чином, ринок спортивного харчування демонструє значне зростання завдяки зростаючому попиту на продукти спортивного харчування та зростаючій тенденції до здоров'я та фітнесу серед населення. Ринок є висококонкурентним, з присутністю кількох ключових гравців, які інвестують в дослідження і розробки з метою впровадження інноваційних продуктів для задоволення мінливих споживчих уподобань. Очікується, що ринок продовжить зростати в найближчі роки завдяки все більшому поширенню здорового способу життя та зростанню наявного доходу населення.

						Аркуш
						7
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-10.БР	

### 1.1.2 Аналіз попиту на спортивне харчування

Зростаюча популярність здоров'я та фітнесу серед населення призвела до сплеску попиту на продукти спортивного харчування. Спортсмени, бодібілдери та ентузіасти фітнесу все частіше шукають продукти, які допоможуть їм досягти своїх фітнес-цілей та покращити спортивні результати. Попит на продукти спортивного харчування також обумовлений зростаючим усвідомленням важливості здорового харчування і ролі харчування в досягненні фітнес-цілей.

Попит на продукти спортивного харчування не обмежується професійними спортсменами та бодібілдерами. Населення в цілому також все частіше вживає спортивне харчування для підтримки здорового способу життя та покращення загального самопочуття. Зростаюча тенденція веганства та рослинної дієти також призвела до збільшення попиту на білкові добавки на рослинній основі.

Пандемія COVID-19 також сприяла зростанню попиту на продукти спортивного харчування. Пандемія призвела до зміни споживчої поведінки, коли все більше людей зосереджуються на своєму здоров'ї та гарному самопочутті. Закриття спортзалів та фітнес-центрів під час пандемії також призвело до збільшення попиту на домашні тренажери та спортивне харчування.

Ринок спортивного харчування рухається не лише завдяки попиту на продукти спортивного харчування, але й завдяки зміні споживчих уподобань. Споживачі все частіше шукають продукти, які є натуральними, органічними та не містять штучних інгредієнтів. Вони також шукають продукти, які зручні, прості у використанні та можуть вписатися в їхній насичений спосіб життя.

На попит на спортивне харчування також впливають маркетингові стратегії, які застосовують компанії. Компанії інвестують значні кошти в

						ДТЕУ 121 06-10.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			8

маркетинг і рекламу, щоб створити впізнаваність бренду і залучити клієнтів. Використання соціальних мереж та маркетингу впливових осіб також стало популярним у галузі спортивного харчування.

Очікується, що попит на продукти спортивного харчування продовжить зростати в найближчі роки. Очікується, що зростаюче прийняття здорового способу життя та зростаюча тенденція до фітнесу та велнесу будуть стимулювати попит на продукти спортивного харчування. Очікується, що наявність широкого асортименту продуктів спортивного харчування та зростаюча увага до інновацій і розробок продуктів сприятимуть подальшому зростанню ринку.

Попит на продукти спортивного харчування зумовлений зростаючою популярністю здоров'я та фітнесу серед населення, а також підвищенням рівня обізнаності про важливість харчування для досягнення фітнес-цілей. На попит на продукти спортивного харчування також впливають зміна споживчих уподобань, пандемія COVID-19 та маркетингові стратегії, які використовують компанії. Очікується, що ринок продовжить зростати в найближчі роки завдяки все більшому поширенню здорового способу життя та наявності широкого асортименту продуктів спортивного харчування.

Очікується, що світовий ринок спортивного харчування досягне \$45,27 млрд до 2022 року, зростаючи на 8,4% в середньорічному обчисленні з 2017 по 2022 рік (джерело: MarketsandMarkets).

Протеїновий порошок є найпопулярнішим продуктом спортивного харчування, на який припадає понад 60% частки ринку (джерело: Euromonitor International)

Північноамериканський регіон є найбільшим ринком продуктів спортивного харчування, на який припадає близько 40% частки ринку (джерело: Research and Markets)

Очікується, що Азіатсько-Тихоокеанський регіон матиме найвищий

						ДТЕУ 121 06-10.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			9

середньорічний темп зростання на рівні 10,7% протягом прогнозованого періоду з 2019 по 2025 рік (джерело: Mordor Intelligence)

Онлайн-канали продажів складають значну частину ринку спортивного харчування, при цьому очікується, що в період з 2020 по 2027 рр. темпи зростання продажів через електронну комерцію складуть 12,3% (джерело: Allied Market Research).

Найпоширенішими джерелами білка, що використовуються в продуктах спортивного харчування, є сироватковий білок, казеїновий білок, соєвий білок і гороховий білок (джерело: Nutrition Business Journal)

### **1.1.3 Аналіз конкурентів на ринку спортивного харчування**

Ринок спортивного харчування є висококонкурентним, на ньому працюють численні гравці. Компанії на ринку постійно впроваджують інновації та розробляють нові продукти, щоб задовольнити мінливі потреби споживачів. У цьому розділі ми проаналізуємо ключових конкурентів на ринку спортивного харчування, їхні ринкові позиції, стратегії та продуктові пропозиції.

#### **Glanbia plc:**

Glanbia plc є провідним гравцем на ринку спортивного харчування з часткою ринку близько 20%. Компанія пропонує широкий асортимент спортивного харчування під різними брендами, такими як Optimum Nutrition, BSN та ABB. Glanbia plc має потужну дистриб'юторську мережу та присутня у понад 130 країнах світу. Компанія фокусується на розширенні свого продуктового портфелю шляхом злиття та поглинання. У 2020 році Glanbia plc придбала The Isopure Company, LLC, провідного постачальника продуктів спортивного харчування преміум-класу.

#### **PepsiCo, Inc:**

PepsiCo, Inc - великий гравець на ринку спортивного харчування з часткою ринку близько 14%. Компанія пропонує продукти спортивного

						ДТЕУ 121 06-10.БР	Аркуш
							10
Зм.	Аркуш	№ докум	Підпис	Дата			

Компанія пропонує продукти спортивного харчування під брендом Gatorade, який включає протеїнові батончики, енергетичні напої та протеїнові коктейлі. PepsiCo, Inc зосереджується на розширенні свого продуктового портфеля за рахунок інновацій та стратегічних партнерств. У 2021 році компанія оголосила про партнерство з Beyond Meat для розробки та просування на ринок білкових продуктів на рослинній основі.

**Abbott Laboratories:**

Abbott Laboratories - провідний гравець на ринку спортивного харчування з часткою ринку близько 10%. Компанія пропонує продукти спортивного харчування під брендом EAS, який включає протеїнові порошки, батончики та напої. Abbott Laboratories зосереджується на розширенні свого продуктового портфелю за рахунок інновацій та партнерств. У 2020 році компанія оголосила про партнерство з Національною асоціацією студентського спорту (NCAA) з метою надання продуктів спортивного харчування та освіти для студентів-спортсменів.

**GlaxoSmithKline plc:**

GlaxoSmithKline plc - великий гравець на ринку спортивного харчування з часткою ринку близько 6%. Компанія пропонує продукти спортивного харчування під брендом MaxiNutrition, який включає протеїнові порошки, батончики та снеки. GlaxoSmithKline plc зосереджується на розширенні свого продуктового портфеля за рахунок інновацій та партнерств. У 2021 році компанія оголосила про партнерство з Оксфордським університетом для дослідження впливу харчування на спортивні результати.

**MusclePharm Corporation:**

MusclePharm Corporation - нішевий гравець на ринку спортивного харчування з часткою ринку близько 3%. Компанія пропонує продукти спортивного харчування під брендом MusclePharm, який включає протеїнові порошки, батончики та передтренувальні добавки. Корпорація MusclePharm

						ДТЕУ 121 06-10.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			11



зосереджується на розширенні свого продуктового портфоліо за рахунок інновацій та партнерств. У 2020 році компанія оголосила про партнерство з BioZone Laboratories для розробки та просування на ринок продуктів спортивного харчування.

Отже, ринок спортивного харчування є висококонкурентним, і в галузі працює кілька гравців. Glanbia plc, PepsiCo, Inc, Abbott Laboratories, GlaxoSmithKline plc і MusclePharm Corporation є одними з ключових гравців на ринку. Ці компанії зосереджені на розширенні свого продуктового портфелю за рахунок інновацій та стратегічного партнерства. Очікується, що ринок продовжить зростати в найближчі роки завдяки зростаючому прийняттю здорового способу життя та зростаючій тенденції фітнесу та велнесу.

## 1.2. Огляд програмного додатка торговельного підприємства

У сучасному конкурентному бізнес-середовищі для роздрібних торговців важливо мати ефективні та дієві програмні рішення для управління своїми операціями та покращення якості обслуговування клієнтів. У цьому розділі ми надамо огляд програмних рішень, доступних для підприємств роздрібної торгівлі.

Системи касових терміналів (POS):

POS-системи є одним з найважливіших програмних рішень для роздрібного бізнесу. Вони дозволяють підприємствам управляти операціями з продажу, відстежувати запаси та генерувати звіти. Деякі з популярних POS-систем на ринку включають Square POS, Shopify POS та Lightspeed POS.

Програмне забезпечення для управління взаємовідносинами з клієнтами (CRM):

Програмне забезпечення CRM допомагає ритейлерам управляти даними про клієнтів, відстежувати взаємодію з клієнтами та аналізувати їхню поведінку. Ця інформація може бути використана для персоналізації клієнтського досвіду та покращення утримання клієнтів. Деякі з популярних

						ДТЕУ 121 06-10.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			12

програмних рішень CRM включають HubSpot CRM, Salesforce CRM і Zoho CRM.

Програмне забезпечення для управління запасами:

Програмне забезпечення для управління запасами допомагає ритейлерам відстежувати рівень запасів, контролювати їх рух та автоматизувати поповнення запасів. Це програмне забезпечення також може допомогти бізнесу оптимізувати рівень запасів і зменшити витрати. Серед популярних програмних рішень для управління запасами - TradeGecko, Cin7 та Brightpearl.

Платформи електронної комерції:

Платформи електронної комерції дозволяють ритейлерам продавати свої товари онлайн, керувати інтернет-магазином та обробляти онлайн-транзакції.

Ці платформи також можуть інтегруватися з іншими програмними рішеннями, такими як управління запасами та програмне забезпечення CRM. Деякі з популярних платформ електронної комерції включають Shopify, WooCommerce та BigCommerce.

Програмне забезпечення для бухгалтерії:

Програмне забезпечення для бухгалтерії допомагає ритейлерам керувати фінансовими операціями, відстежувати витрати та створювати фінансові звіти. Це програмне забезпечення також може автоматизувати такі бухгалтерські процеси, як виставлення рахунків та нарахування заробітної плати. Деякі з популярних бухгалтерських програмних рішень включають QuickBooks, Xero та Sage.

Програмне забезпечення для бізнес-аналітики (BI):

Програмне забезпечення для бізнес-аналітики допомагає ритейлерам аналізувати дані про продажі, відстежувати показники ефективності та виявляти тенденції. Ця інформація потім може бути використана для прийняття рішень на основі даних і підвищення ефективності бізнесу.

						Аркуш
						13
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-10.БР	

Деякі з популярних програмних рішень BI включають Tableau, Microsoft Power BI і QlikView.

Підприємства роздрібної торгівлі мають широкий спектр програмних рішень для управління своїми операціями та покращення клієнтського досвіду. POS-системи, програмне забезпечення CRM, програмне забезпечення для управління запасами, платформи електронної комерції, бухгалтерське програмне забезпечення та програмне забезпечення для бізнес-аналітики - ось деякі з популярних програмних рішень, що використовуються ритейлерами. Підприємства повинні ретельно оцінити свої потреби в програмному забезпеченні та обрати програмні рішення, які найкраще відповідають їхнім вимогам та бюджету.

### **1.2.1 Опис та класифікація програмного додатка торговельного підприємства**

Програмні рішення необхідні для управління операціями будь-якого сучасного роздрібного бізнесу. Ці програмні рішення призначені для автоматизації різних бізнес-процесів, підвищення ефективності та покращення загального рівня обслуговування клієнтів. У цьому розділі ми надамо огляд різних типів програмних рішень, доступних для роздрібного бізнесу, їх класифікацію та функції.

Системи касових терміналів (POS):

POS-системи є одними з найважливіших програмних рішень для роздрібного бізнесу. Вони використовуються для управління торговими операціями, відстеження запасів і створення звітів. Ці системи можна розділити на дві основні категорії: традиційні та хмарні POS-системи.

Традиційні POS-системи є апаратними і вимагають встановлення програмного забезпечення на локальному сервері. Ці системи, як правило, мають більше можливостей і краще налаштовуються, але вони також і дорожчі. З іншого боку, хмарні POS-системи - це програмне забезпечення як

						Аркуш
					ДТЕУ 121 06-10.БР	14
Зм.	Аркуш	№ докум	Підпис	Дата		

послуга (SaaS), яке розміщується в хмарі і доступне через Інтернет. Ці системи є більш доступними за ціною, їх легше налаштувати і вони мають менше вимог до апаратного забезпечення.

Програмне забезпечення для управління взаємовідносинами з клієнтами (CRM):

Програмне забезпечення CRM використовується для управління даними про клієнтів, відстеження взаємодії з ними та аналізу їхньої поведінки. Ця інформація може бути використана для персоналізації клієнтського досвіду та покращення утримання клієнтів. Програмне забезпечення CRM можна розділити на дві основні категорії: локальне та хмарне програмне забезпечення CRM.

Локальне програмне забезпечення CRM встановлюється на локальних серверах і вимагає значних початкових інвестицій в апаратне та програмне забезпечення. Ці системи забезпечують більший контроль і кастомізацію, але також є більш дорогими в обслуговуванні. Хмарне програмне забезпечення CRM, з іншого боку, розміщується в хмарі і доступ до нього здійснюється через Інтернет. Ці системи є доступнішими, вимагають менших інвестицій в апаратне забезпечення і доступні звідусіль.

Програмне забезпечення для управління запасами:

Програмне забезпечення для управління запасами використовується для відстеження рівня запасів, моніторингу руху запасів та автоматизації поповнення запасів. Це програмне забезпечення також може допомогти бізнесу оптимізувати рівень запасів та зменшити витрати. Програмне забезпечення для управління запасами можна розділити на дві основні категорії: локальне та хмарне програмне забезпечення для управління запасами.

Локальне програмне забезпечення для управління запасами встановлюється на локальних серверах і вимагає значних початкових

						Аркуш
					ДТЕУ 121 06-10.БР	15
Зм.	Аркуш	№ докум	Підпис	Дата		

інвестицій в апаратне та програмне забезпечення. Ці системи забезпечують більший контроль і кастомізацію, але також є більш дорогими в обслуговуванні. Хмарне програмне забезпечення для управління запасами, з іншого боку, розміщується в хмарі і доступ до нього здійснюється через Інтернет. Ці системи є доступнішими, вимагають менших інвестицій в апаратне забезпечення і доступні звідусіль.

Платформи електронної комерції: Платформи електронної комерції використовуються для продажу товарів онлайн, управління інтернет-магазинами та обробки онлайн-транзакцій. Ці платформи можна розділити на дві основні категорії: хостингові та самостійні платформи електронної комерції.

Хостингові платформи електронної комерції - це програмне забезпечення як послуга (SaaS), яке розміщується в хмарі і доступне через Інтернет. Ці системи є більш доступними і вимагають менших інвестицій в апаратне забезпечення, але можуть мати обмежені можливості налаштування. З іншого боку, самостійні платформи електронної комерції вимагають встановлення на локальних серверах і надають більше можливостей для контролю та налаштування.

Програмне забезпечення для бухгалтерії:

Бухгалтерське програмне забезпечення використовується для управління фінансовими операціями, відстеження витрат і створення фінансових звітів. Це програмне забезпечення також може автоматизувати такі бухгалтерські процеси, як виставлення рахунків та нарахування заробітної плати. Бухгалтерське програмне забезпечення можна розділити на дві основні категорії: локальне та хмарне програмне забезпечення.

Локальне бухгалтерське програмне забезпечення встановлюється на локальних серверах і вимагає значних початкових інвестицій в апаратне та програмне забезпечення. Ці системи забезпечують більший контроль і

						ДТЕУ 121 06-10.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			16

можливість кастомізації, але також є більш дорогими в обслуговуванні. З іншого боку, хмарне бухгалтерське програмне забезпечення розміщується в хмарі, а доступ до нього здійснюється через Інтернет. Ці системи є доступнішими, вимагають менших інвестицій в апаратне забезпечення і доступні звідусіль.

### 1.3. Висновки до розділу 1

В розділі 1 "Аналіз предметної області" було проведено детальний аналіз ринку спортивного харчування та огляд програмного додатка для торговельного підприємства.

Загальною метою аналізу предметної області було отримати глибоке розуміння ринку спортивного харчування та програмного додатка для торговельного підприємства. Цей аналіз надав нам основні знання про стан ринку, попит, конкурентний ландшафт та наявні програмні рішення. Ця інформація є важливою для подальшого розроблення стратегій та рекомендацій, що дозволять ефективно функціонувати у сфері спортивного харчування та використовувати відповідне програмне забезпечення для оптимізації торговельного підприємства.

						ДТЕУ 121 06-10.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			17

## РОЗДІЛ 2

### ПРОЕКТУВАННЯ ПРОГРАМНОГО ДОДАТКА ТОРГІВЕЛЬНОЇ ДІЯЛЬНОСТІ

#### 2.1 Вимоги до програмного додатка

При розробці програмного додатка для роздрібного бізнесу, що спеціалізується на спортивному харчуванні, важливо враховувати вимоги бізнесу. Нижче наведені деякі з ключових вимог, які слід враховувати при розробці програмного додатка:

**Система точки продажу (POS):** Програмний додаток повинен мати POS-систему, яка може бути використана для швидкої та ефективної обробки транзакцій продажу. POS-система також повинна мати можливість управляти запасами і генерувати звіти про продажі, рівні запасів і прибутки.

**Управління продуктами:** Програмний додаток повинен дозволяти бізнесу легко керувати асортиментом продукції. Система повинна відстежувати рівень запасів товарів і генерувати сповіщення, коли рівень запасів опускається нижче певного порогу. Система також повинна вміти керувати цінами на продукцію, акціями та знижками.

**Управління клієнтами:** Програмний додаток повинен дозволяти бізнесу легко керувати базою даних клієнтів. Система повинна мати можливість зберігати інформацію про клієнтів, таку як контактні дані, історія покупок та вподобання.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 06-10.БР</i>			
Зав. каф.		Криворучко О.В.		03.03.23	Програмний додаток підприємства з реалізації спортивного харчування	Стадія	Аркуш	Аркушів
Керівник		Власенко Л.О.		03.03.23		P2	18	50
Гарант		Рзаєва С.Л.		03.03.23		Факультет інформаційних технологій 4 курс, 6 група		
Розробив		Доля О.С.		03.03.23				
					Проектування програмного додатка торговельної діяльності			

Система також повинна вміти генерувати звіти про купівельну поведінку клієнтів і тенденції.

Онлайн-замовлення: Програмний додаток повинен дозволяти клієнтам розміщувати замовлення онлайн. Система повинна інтегруватися з веб-сайтом компанії і дозволяти клієнтам переглядати продукти, ціни та акції, а також розміщувати замовлення. Система також повинна мати можливість керувати онлайн-платежами та доставкою.

Звітність та аналітика: Програмний додаток повинен вміти генерувати звіти про продажі, рівень запасів та прибутки. Система також повинна надавати аналітику щодо купівельних моделей і тенденцій клієнтів. Ця інформація може бути використана для прийняття обґрунтованих бізнес-рішень та покращення прибутку компанії.

Інтеграція з іншими системами: Програмний додаток повинен інтегруватися з іншими системами, такими як бухгалтерське програмне забезпечення, інструменти автоматизації маркетингу та системи управління взаємовідносинами з клієнтами (CRM). Це дозволить бізнесу оптимізувати свою діяльність та підвищити ефективність.

Безпека: Програмний додаток повинен бути розроблений з урахуванням безпеки. Система повинна бути здатна захистити дані клієнтів та бізнес-інформацію від кіберзагроз, таких як хакерські атаки та атаки шкідливих програм.

Враховуючи ці вимоги, програмний додаток може бути спроектований і розроблений відповідно до потреб бізнесу та підвищити його загальну ефективність і прибутковість.

## 2.2 Архітектруа програмного додатка

Архітектура програмного додатка - це дизайн та організація його компонентів, таких як модулі, шари та підсистеми, а також те, як вони взаємодіють один з одним для досягнення цілей програмного додатка. У

						Аркуш
					ДТЕУ 121 06-10.БР	19
Зм.	Аркуш	№ докум	Підпис	Дата		



випадку з програмним додатком для торгового підприємства, що займається реалізацією спортивного харчування, архітектура повинна бути ретельно спланована і спроектована, щоб забезпечити його гнучкість, масштабованість, безпеку та ефективність.

Архітектуру програмного додатка можна розділити на дві основні складові: фронтенд і бекенд. Фронт-енд - це користувацький інтерфейс програмного додатка, з яким взаємодіють кінцеві користувачі, тоді як бек-енд відповідає за обробку та зберігання даних. Фронтенд і бекенд взаємодіють між собою за допомогою інтерфейсу прикладного програмування (API).

Інтерфейс програмного додатка для торговельного підприємства повинен бути розроблений з урахуванням потреб кінцевих користувачів. Він повинен бути інтуїтивно зрозумілим, зручним і простим для навігації. Інтерфейс повинен бути здатним обробляти велику кількість даних і реагувати на різні розміри екранів. Користувацький інтерфейс повинен дозволяти користувачам легко шукати товари, додавати їх до кошика та здійснювати платежі.

Внутрішня частина програмного додатка повинна бути розроблена з урахуванням бізнес-логіки торговельного підприємства. Вона повинна бути масштабованою і здатною обробляти велику кількість транзакцій одночасно. Внутрішня частина програмного додатка також повинна бути спроектована для зберігання та пошуку даних. Вона повинна мати можливість обробляти та зберігати інформацію про клієнтів, інформацію про товари, дані про продажі та інші важливі дані, пов'язані з торговельним підприємством.

Архітектура програмного додатка повинна також включати базу даних для зберігання даних, що використовуються програмним додатком. База даних повинна бути ефективною, безпечною та масштабованою. Вона повинна бути здатна обробляти велику кількість даних і бути розрахована на одночасну

						ДТЕУ 121 06-10.БР	Аркуш
							20
Зм.	Аркуш	№ докум	Підпис	Дата			

роботу декількох користувачів. База даних також повинна бути легкодоступною і забезпечувати швидкий час відгуку.

Ще одним важливим моментом в архітектурі програмного додатку є безпека. Програмний додаток має бути розроблений з урахуванням безпеки, щоб запобігти несанкціонованому доступу до конфіденційних даних. Програмний додаток також має бути спроектований таким чином, щоб запобігати втраті даних і забезпечувати регулярне резервне копіювання даних.

Архітектура програмного додатка для торговельного підприємства, що займається реалізацією спортивного харчування, повинна бути ретельно спланована і спроектована таким чином, щоб забезпечити його гнучкість, масштабованість, безпеку та ефективність. Фронтенд і бекенд компоненти повинні бути спроектовані так, щоб безперешкодно працювати один з одним, а база даних повинна бути спроектована так, щоб обробляти велику кількість даних. Програмний додаток також має бути розроблений з урахуванням вимог безпеки, щоб запобігти несанкціонованому доступу до конфіденційних даних.

### 2.3 Розробка бази даних

Розробка бази даних є важливим аспектом розробки програмного додатка. База даних служить основою програмної системи і відповідає за зберігання та організацію всіх даних, які використовуються системою. Для розробки програмного додатка для торгового підприємства, що займається продажем спортивного харчування, добре спроектована база даних має вирішальне значення для забезпечення ефективної та результативної роботи програмного додатка.

Першим кроком у розробці бази даних є визначення моделі даних. Модель даних описує структуру даних, які будуть зберігатися в базі даних. Сюди входить визначення таблиць, стовпців, зв'язків між таблицями, а також будь-яких обмежень або правил, які необхідно застосувати до даних.

						ДТЕУ 121 06-10.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			21

Для торгового підприємства, що займається продажем спортивного харчування, база даних повинна зберігати інформацію про товари, клієнтів, замовлення та запаси. Кожна з цих сутностей повинна бути представлена таблицею в базі даних. Наприклад, таблиця продуктів повинна зберігати інформацію про кожен продукт, таку як його назва, опис, ціна та кількість.

База даних також повинна містити зв'язки між різними сутностями. Наприклад, кожне замовлення буде пов'язане з клієнтом, а кожне замовлення включатиме один або кілька продуктів. Цей зв'язок може бути представлений зовнішнім ключем у таблиці замовлень, який посилається на таблицю клієнтів, та іншим зовнішнім ключем у таблиці замовлень, який посилається на таблицю продуктів.

Для забезпечення цілісності та узгодженості даних у базі даних можуть бути визначені обмеження. Наприклад, до таблиці товарів можна додати обмеження, яке гарантуватиме, що ціна товару завжди буде більшою за нуль. Аналогічно, обмеження можна додати до таблиці замовлень, щоб гарантувати, що загальна вартість замовлення завжди розраховується правильно на основі цін продуктів і замовленої кількості.

База даних може бути створена за допомогою різних інструментів і технологій. Наприклад, популярними системами управління базами даних є MySQL, Oracle і Microsoft SQL Server. Ці системи надають цілий ряд функцій і можливостей для проектування та управління базами даних, включаючи інструменти моделювання даних, інструменти запитів і звітів, а також функції резервного копіювання та відновлення.

Таким чином, розробка бази даних є критично важливим компонентом розробки програмного додатка для торгового підприємства, що займається продажем спортивного харчування. База даних служить основою для програмної системи і відповідає за зберігання та організацію всіх даних, які використовуються системою. Завдяки розробці добре структурованої бази

						ДТЕУ 121 06-10.БР	Аркуш
							22
Зм.	Аркуш	№ докум	Підпис	Дата			

даних, яка включає відповідні таблиці, взаємозв'язки та обмеження, програмний додаток може працювати ефективно та результативно, допомагаючи бізнесу більш ефективно управляти запасами та продажами.

У нашому випадку для розробки бази даних для програмного додатка було обрано sql та на рисунку нижче приведено структуру бази (Рис 2.1)

Програмний додаток побудований за допомогою мови програмування

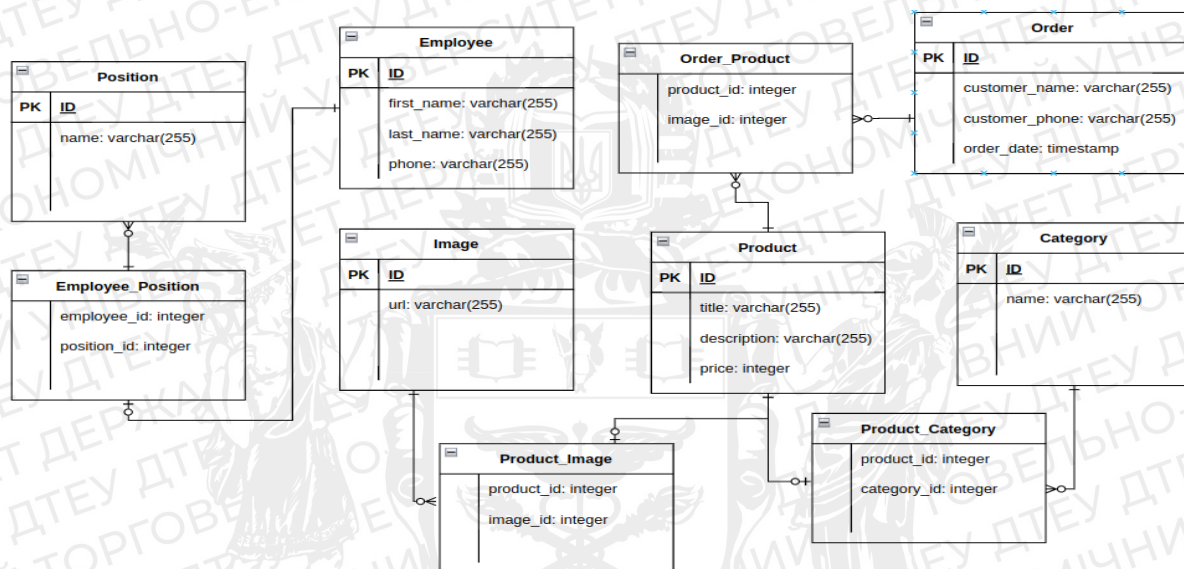


Рисунок 2.1-Модель розробленої бази даних

*Джерело: побудовано автором*

Python з використаннями веб-фреймворку Flask. Програма розроблена на архітектурі Model-View-Controller (MVC) та поєднана із базою даних SQLite

Для побудови даної системи керування клієнтами компанії було визначено певні моделі даних, кожна з яких буде містити певні інші дані.

## 2.4 Розробка інтерфейсу користувача

Інтерфейс користувача (UI) є важливим аспектом розробки програмного додатка, оскільки це основний спосіб, за допомогою якого користувачі взаємодіють з системою. Інтерфейс повинен бути інтуїтивно зрозумілим, зручним і естетично привабливим, щоб заохочувати залучення користувачів і забезпечувати позитивний користувацький досвід. У випадку програмного

дodatка, що розробляється для роздрібної торгівлі спортивним харчуванням, інтерфейс повинен бути спроектований таким чином, щоб клієнти могли легко орієнтуватися і розуміти його.

Інтерфейс для програмного додатка буде розроблений з урахуванням потреб користувача, з акцентом на забезпечення ефективного і безперебійного досвіду для клієнтів. Інтерфейс буде розроблений з використанням сучасних принципів дизайну, включаючи плоский дизайн і дизайн матеріалів, щоб створити елегантний і візуально привабливий інтерфейс, який буде простим у використанні.

Інтерфейс буде розділений на кілька розділів для забезпечення простоти використання. Домашній екран буде першою точкою контакту для клієнтів, і він надасть огляд продуктів, доступних для придбання. На головному екрані буде розміщено рядок пошуку, який дозволить покупцям швидко шукати товари за ключовими словами або назвами продуктів. Пошуковий рядок буде легко доступний з усіх сторінок, і він надаватиме підказки при введенні клієнтом, допомагаючи йому швидше знайти те, що йому потрібно. Сторінка продукту надаватиме детальну інформацію про кожен продукт, включаючи назву, опис, ціну та інформацію про поживну цінність. Сторінка товару також дозволить покупцям переглянути зображення товару, прочитати відгуки інших покупців і додати товар у кошик для покупки. Покупці зможуть легко переміщатися між різними категоріями продуктів за допомогою фільтра категорій, який дозволить їм переглядати продукти за категоріями, наприклад, протеїнові батончики, харчові добавки та замітники їжі.

Сторінка кошика дозволить покупцям переглядати товари, які вони додали до свого кошика, змінювати кількість кожного товару та видаляти товари з кошика. На сторінці кошика також відобразатиметься загальна вартість замовлення, включаючи податки та витрати на доставку. Сторінка оформлення замовлення забезпечить простий і безпечний процес оформлення

						ДТЕУ 121 06-10.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			24

замовлення з можливістю введення інформації про доставку та виставлення рахунку, вибору способу доставки та здійснення платежу за допомогою різних способів оплати, включаючи кредитні картки, PayPal та Apple Pay.

Інтерфейс користувача буде розроблений з урахуванням адаптивності, що гарантує, що він буде оптимізований для використання як на настільних, так і на мобільних пристроях. Інтерфейс буде розроблений з використанням HTML, CSS та JavaScript, з акцентом на створенні швидкого та адаптивного інтерфейсу, який швидко завантажується на всіх пристроях.

Загалом, інтерфейс для програмного додатка, що розробляється для роздрібної торгівлі спортивним харчуванням, буде розроблений з думкою про користувача, з акцентом на створенні безперебійного та ефективного користувацького досвіду. Інтерфейс буде візуально привабливим, простим у використанні та оптимізованим для використання як на настільних, так і на мобільних пристроях, забезпечуючи сучасний та професійний вигляд програмного додатка. MVC (Model-View-Controller) - це популярний архітектурний патерн, який використовується у розробці програмного додатка, зокрема у розробці веб-додатків. Він розділяє додаток на три взаємопов'язані компоненти, кожен з яких виконує певну роль:

**Модель:** Цей компонент відповідає за управління даними та бізнес-логікою програми. Він взаємодіє з базою даних для отримання та зберігання даних, а також містить логіку для маніпулювання цими даними відповідно до правил бізнес-домену.

**Представлення:** Цей компонент відповідає за відображення даних користувачеві у форматі, зручному для розуміння і взаємодії з ними. Це може бути веб-сторінка, екран мобільного додатка або будь-який інший елемент інтерфейсу користувача.

**Контролер:** Цей компонент виступає посередником між моделлю і представленням, отримуючи вхідні дані від користувача, інтерпретуючи їх і

						Аркуш
					ДТЕУ 121 06-10.БР	25
Зм.	Аркуш	№ докум	Підпис	Дата		

визначаючи відповідну реакцію. Він взаємодіє як з моделлю, так і з представленням, оновлюючи модель будь-якими змінами, внесеними користувачем, і оновлюючи представлення для відображення цих змін.

Основна перевага використання патерну MVC полягає в тому, що він розділяє проблеми і сприяє модульному дизайну коду. Кожен компонент може бути розроблений незалежно, що полегшує модифікацію або оновлення однієї частини програми, не впливаючи на інші. Це також сприяє повторному використанню коду і полегшує тестування програми, оскільки кожен компонент виконує певну роль і може бути протестований незалежно.

Нижче наведена абстрактна модель всього додатка в загальному понятті архітектури MVC. (Рис 2.2)

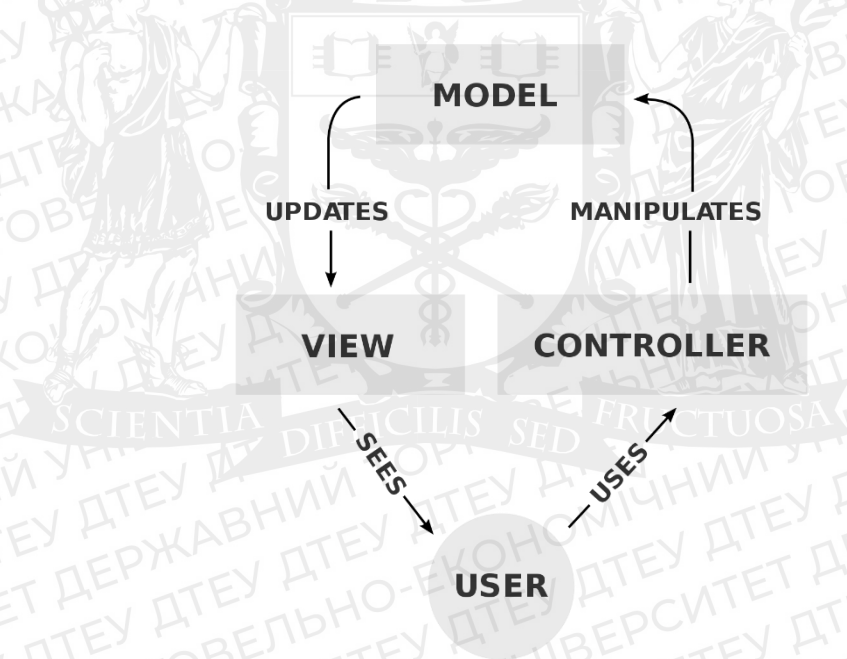


Рисунок 2.2-Абстрактна модель архітектури додатка

*Джерело: побудовано автором*

На діаграмі (рис 2.3) зображено алгоритми роботи програми при роботі з додаванням, редагуванням та видаленням певних сутностей. При добавленні продукту, йде перевірка на те, чи є картинка продукту у базі даних. Якщо да, то до продукту прив'язується ця картинка, якщо ні – то спочатку вона добавляється до БД.

					ДТЕУ 121 06-10.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		26

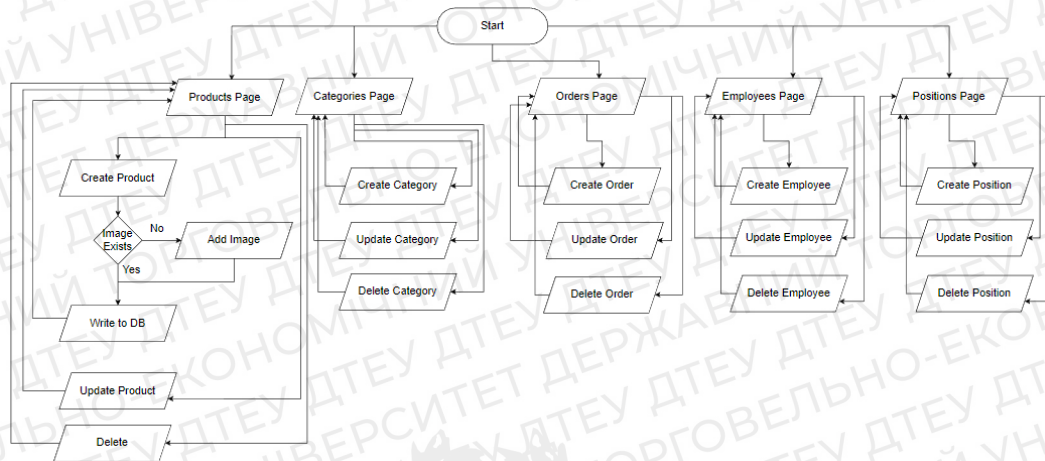


Рисунок 2.3-Діаграма алгоритму програми

Джерело: побудовано автором

На цій діаграмі(рис 2.4) зображено те, хто може користуватися сайтом, та які операції виконувати. Наприклад, існує адміністратор, який може робити все, окрім виготовлення продукції. А от робітник має право лише на цю дію.

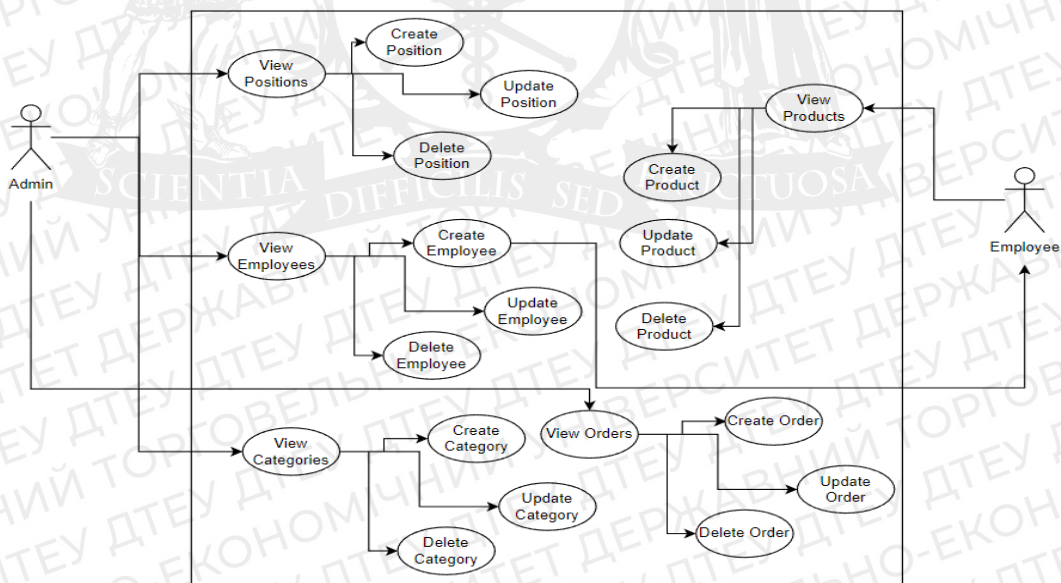


Рисунок 2.4-Діаграма використання

Джерело: побудовано автором

На даній діаграмі(рис 2.5) показано всю архітектуру додатка з виду програмного коду, у функцій та файлів.

						Аркуш
						27
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-10.БР	



Діаграма компонентів - це тип структурної діаграми в уніфікованій мові моделювання (UML), яка ілюструє компоненти системи та залежності між ними. Вона використовується для зображення організації та залежностей між компонентами системи, включаючи програмні модулі, апаратні пристрої та сховища даних.

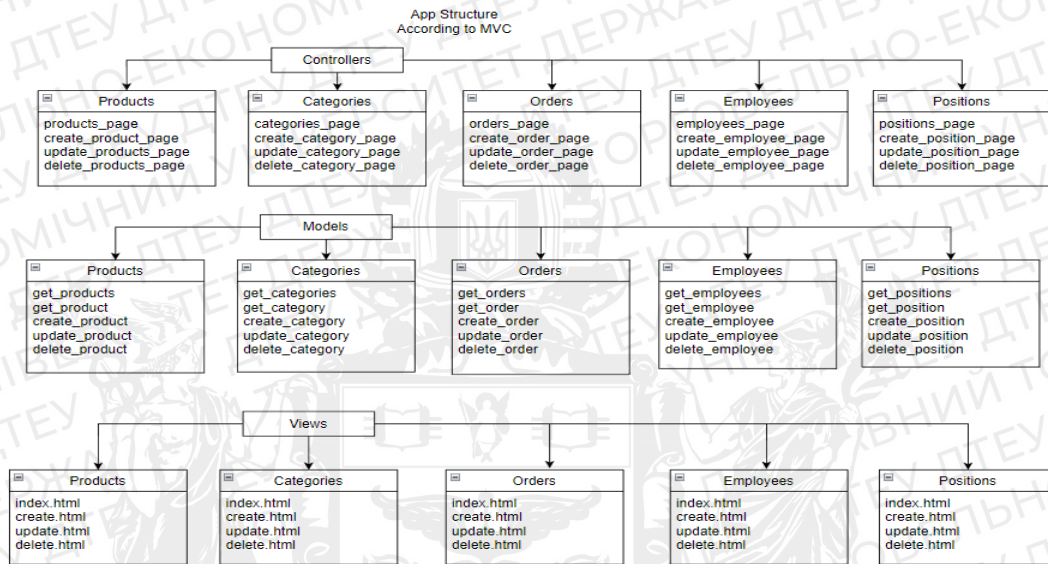


Рисунок 2.5 Діаграма додатка стосовно MVC

*Джерело: побудовано автором*

На діаграмі компонентів компоненти зображуються прямокутниками з назвою компонента всередині. Прямокутники з'єднані лініями, які відображають залежності між компонентами. Ці лінії можуть бути позначені, щоб вказати тип зв'язку, наприклад, "використовує", "реалізує" або "вимагає".

Компонент може бути розбитий на менші підкомпоненти, які представлені прямокутниками, вкладеними в більший прямокутник компонента. Підкомпоненти також можуть бути з'єднані з іншими компонентами, щоб показати їхню залежність.

Діаграми компонентів корисні для визначення ключових компонентів системи та розуміння того, як вони взаємодіють один з одним. Вони часто використовуються на етапі проектування при розробці програмного додатка,

					Аркуш
					28
Зм.	Аркуш	№ докум	Підпис	Дата	

щоб допомогти розробникам ідентифікувати та організувати компоненти системи і переконатися, що вони належним чином з'єднані, а залежності чітко визначені.

Окрім розробки програмного додатка, діаграми компонентів можуть використовуватися в багатьох інших сферах, включаючи проектування апаратного забезпечення, системну архітектуру та моделювання бізнес-процесів. Їх можна використовувати для моделювання складних систем і виявлення потенційних недоліків проектування або областей для оптимізації.

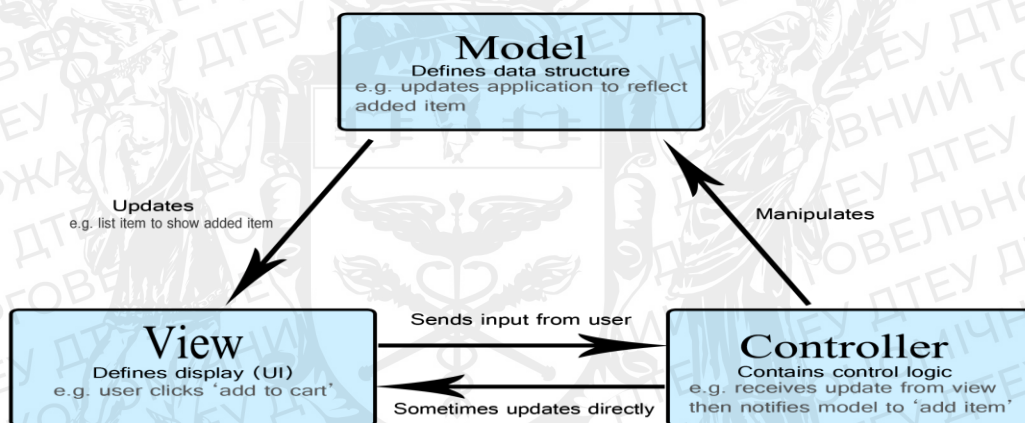


Рисунок 2.6- Діаграма компонентів

Джерело: побудовано автором

На цій діаграмі (рис 2.6)показано точні залежності та взаємозв'язки між компонентами, їх логічними або ж фізичними модулями програми (або ж програмного коду).

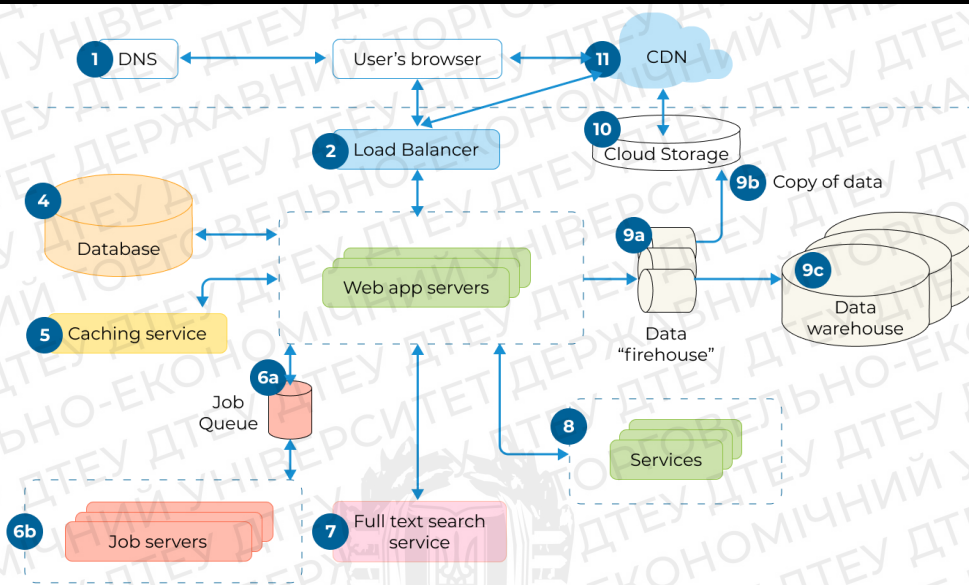


Рисунок 2.7- Діаграма розгортання

Джерело: побудовано автором

На цій (рис 2.7)діаграмі відображено спеціальні фізичні вузли у зв'язках з логічними. Вони виконують певні функції під час роботи програми.

Фізичне розгортання артефактів на цих вузлах, забезпечена вкладеність, захищеність між передачею даних спеціальними протоколами та з'єднанням фізичними вузлами.

Інтерфейс сайту(Рис 2.8) було розроблено з використанням мов HTML та CSS. Також був використаний фреймворк Bootstrap.

Вигляд головної сторінки:

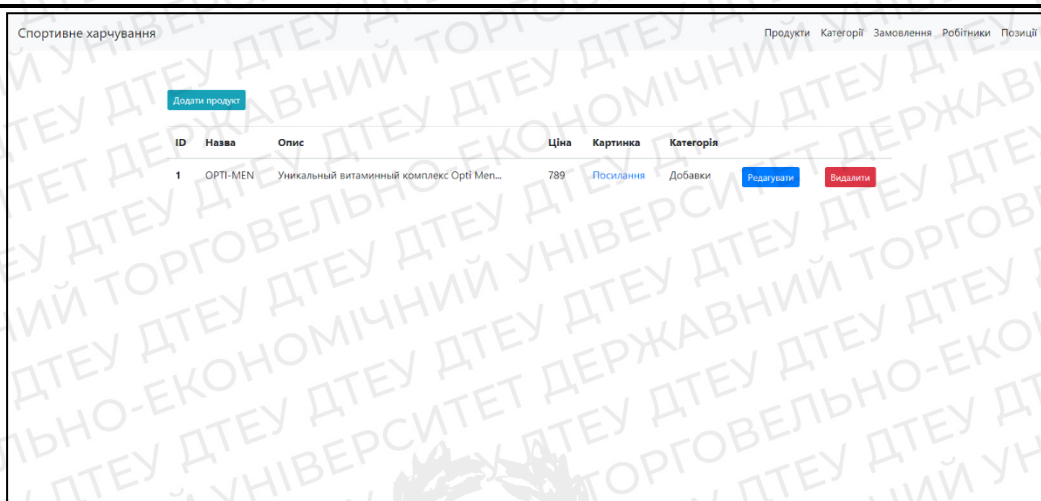


Рисунок 2.8-Інтерфейс ПЗ

*Джерело: побудовано автором*

В меню є 5 пунктів:

- Продукти
- Категорії
- Замовлення
- Робітники
- Позиції

Даний FRONT побудований згідно правил “USER EXPERIENCE”, має кольори, з якими зручно працювати, а розташування теж підлягає правилам.

На кожній сторінці присутня таблиця, то кнопки для:

- Створення сутностей
- Редагування сутностей
- Видалення сутностей

## 2.5. Висновки до розділу 2

Розділ 2 "Проектування програмного додатка для торговельної діяльності" містить розгорнуту інформацію про проектування програмного додатка для торговельного підприємства.

						Аркуш
						31
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-10.БР	

У підрозділі 2.1 "Вимоги до програмного додатка" були визначені вимоги до програмного додатка, необхідного для ефективної торговельної діяльності. Цей підрозділ містить перелік функціональних та нефункціональних вимог, які повинні бути враховані при розробці програмного додатка.

У підрозділі 2.2 "Архітектура програмного додатка" була описана архітектура програмного додатка для торговельного підприємства. Цей підрозділ включає в себе визначення компонентів системи, їх взаємодію та структуру системи в цілому.

У підрозділі 2.3 "Розробка бази даних" була проведена розробка бази даних для програмного додатка. Цей підрозділ включає проектування схеми бази даних, визначення таблиць, полів та зв'язків між ними. База даних є важливою складовою програмного додатка і дозволяє зберігати, організувати та отримувати доступ до інформації, необхідної для торговельної діяльності.

У підрозділі 2.4 "Розробка інтерфейсу користувача" була проведена розробка інтерфейсу користувача програмного додатка. Цей підрозділ включає проектування графічного інтерфейсу, визначення функціональності та навігації, які допоможуть користувачам зручно взаємодіяти з програмним додатком.

Розділ 2 надає основні керівництва та рекомендації щодо проектування програмного додатка для торговельної діяльності, забезпечуючи необхідність створення ефективної та функціональної системи для підприємства.

						ДТЕУ 121 06-10.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			32

## РОЗДІЛ 3

### ТЕСТУВАННЯ ПРОГРАМНОГО ДОДАТКА

#### 3.1 Опис методів тестування програмного додатка

Тестування програмного додатка є важливим процесом у розробці програмного додатка, оскільки воно допомагає гарантувати, що кінцевий продукт відповідає бажаним вимогам до якості та функціональності. Існують різні методи тестування програмного додатка, які можна використовувати залежно від характеру розроблюваного програмного додатка, його призначення та бажаного рівня впевненості в якості програмного додатка. У цьому розділі ми надамо огляд найпоширеніших методів тестування програмного додатка.

#### Модульне тестування

Модульне тестування передбачає тестування окремих блоків або модулів програмного додатка ізольовано від решти системи. Мета полягає в тому, щоб переконатися, що кожен модуль працює так, як очікується, і виявити будь-які дефекти на ранній стадії процесу розробки. Юніт-тести, як правило, автоматизовані і часто запускаються під час розробки, щоб виявити помилки на ранніх стадіях.

#### Інтеграційне тестування

Інтеграційне тестування передбачає перевірку того, як окремі блоки або модулі програмного додатка працюють разом як єдине ціле.

Мета - виявити будь-які дефекти, які можуть виникнути при об'єднанні блоків, і переконатися, що програмний додаток працює так, як очікується, як система. Інтеграційне тестування зазвичай виконується після модульного тестування і, як правило, також автоматизоване.

Зм.	Аркуш	№ докум.	Підпис	Дата			
Зав. каф.	Криворучко О.В.			14.04.23	Програмний додаток підприємства з реалізації спортивного харчування		
Керівник	Власенко Л.О.			14.04.23			
Гарант	Рзасва С.Л.			14.04.23	Факультет інформаційних технологій 4 курс, 6 група		
Розробив	Доля О.С.			14.04.23			
					Тестування програмного додатка		

ДТЕУ 121 06-10.БР

### Системне тестування

Системне тестування передбачає тестування програмного додатка в цілому, щоб переконатися, що воно відповідає бажаним функціональним і нефункціональним вимогам. Цей тип тестування зазвичай виконується після інтеграційного тестування і передбачає тестування програмного додатка в різних середовищах, щоб переконатися, що воно працює у всіх сценаріях.

### Приймальне тестування

Приймальне тестування передбачає тестування програмного додатка, щоб переконатися, що воно відповідає вимогам кінцевих користувачів і зацікавлених сторін. Мета полягає в тому, щоб переконатися, що програмний додаток є прийнятним для випуску і відповідає потребам його цільових користувачів.

### Регресійне тестування

Регресійне тестування передбачає тестування програмного додатка після внесення змін або модифікацій в систему, щоб переконатися, що не з'явилися нові дефекти і що існуюча функціональність все ще працює належним чином. Цей тип тестування важливий для того, щоб переконатися, що зміни в програмному забезпеченні не мають непередбачуваних наслідків для решти системи.

### Тестування продуктивності

Тестування продуктивності передбачає перевірку здатності програмного додатка працювати під різними робочими навантаженнями, наприклад, стрес-тестування, навантажувальне тестування та тестування на витривалість. Мета полягає в тому, щоб переконатися, що програмний додаток може впоратися з очікуваними рівнями використання без уповільнення або збоїв.

### Тестування безпеки

Тестування безпеки передбачає перевірку здатності програмного додатка протистояти зовнішнім загрозам, таким як злом, несанкціонований

						ДТЕУ 121 06-10.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			34

доступ та зловмисні атаки. Мета полягає в тому, щоб переконатися, що програмний додаток є безпечним і що дані користувачів захищені.

**3.2 Тестування розробленого додатка** Після розробки додатка та ознайомившись з процесом тестування переходимо до тестування додатка.

Для цього на головній сторінці потрібно перейти на сторінку Додати продукт(кнопка перед таблицею).(рис 3.1)



Рисунок 3.1-Додавання продукту

*Джерело: побудовано автором*

Ми попадаємо на сторінку, яка має вигляд(рис 3.2)

Рисунок 3.2-Додавання та опис продукту

*Джерело: побудовано автором*



Заповнюємо текстові поля, вибираємо категорію, після чого нажимаємо Зберегти(рис 3.3)

Назва

Назва

Опис

Опис

Ціна

1000

Картинка

https://bodymarket.ua/img/p/4/2/8/4/4284-large\_default.jpg

Категорія

Добавки

Виберіть категорію

Добавки

Рисунок 3.3-Додавання опису та функція збереження

*Джерело: побудовано автором*

Тепер на головній сторінці ми можемо побачити тільки-но доданий нами продукт(Рис 3.4)

Додати продукт

ID	Назва	Опис	Ціна	Картинка	Категорія		
1	OPTI-MEN	Унікальний вітамінний комплекс Opti Men...	789	Посилання	Добавки	Редагувати	Видалити
15	Назва	Опис	1000	Посилання	Добавки	Редагувати	Видалити

Рисунок 3.4-Додані товари

*Джерело: побудовано автором*

Провівши тестування бачимо що програма виконує всі функції які задумані автором та необхідні для виконання завдання

### 3.3. Висновки до розділу 3

Розділ 3 "Тестування програмного додатка" зосереджується на процесі тестування розробленого програмного додатка для торговельного підприємства.

У підрозділі 3.1 "Опис методів тестування програмного додатка" надається огляд методів, які використовуються для тестування програмного додатка. Цей підрозділ містить опис різних підходів до тестування, таких як

						Аркуш
					ДТЕУ 121 06-10.БР	36
Зм.	Аркуш	№ докум	Підпис	Дата		

модульне тестування, інтеграційне тестування, системне тестування та приймальне тестування. Кожен метод має свої переваги та специфіку в застосуванні, що допомагає забезпечити якість та надійність програмного додатка.

У підрозділі 3.2 "Тестування розробленого додатка" проводиться тестування конкретного розробленого додатка для торговельного підприємства. Цей підрозділ включає опис проведених тестів, визначення тестових сценаріїв та результатів тестування. Тестування допомагає виявити помилки, дефекти та недоліки в програмному забезпеченні перед його впровадженням, що дозволяє виправити проблеми та забезпечити належну функціональність та якість продукту.

Розділ 3 надає важливі відомості про тестування програмного додатка, його методи та результати для забезпечення його якості та відповідності вимогам. Тестування є необхідним етапом в процесі розробки програмного додатка, щоб забезпечити коректну та надійну роботу програми перед її використанням.

								Аркуш
								37
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-10.БР			

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В цьому проекті було описано розробку програмного додатка для торгової компанії, що спеціалізується на продажі продуктів спортивного харчування. Метою програмного додатка є автоматизація бізнес-процесів та підвищення ефективності операцій, а також забезпечення кращого клієнтського досвіду.

Аналіз ринку продуктів спортивного харчування показав, що попит на такі продукти зростає, що пов'язано зі збільшенням популярності занять спортом та фітнесом. Ринок є конкурентним, численні компанії пропонують різноманітні продукти, включаючи порошки, батончики та добавки.

Аналіз існуючих програмних рішень для торговельних компаній показав, що існує багато різних типів програмного додатка, починаючи від базових систем для торгових точок і закінчуючи комплексними системами планування ресурсів підприємства. Вибір програмного забезпечення залежить від конкретних потреб компанії та розміру бізнесу.

При розробці програмного додатка використовувалася архітектура Model-View-Controller, яка дозволила розділити завдання і полегшила реалізацію зручного для користувача інтерфейсу. Програмний додаток було розроблено з використанням різних технологій, включаючи python фреймворк Flask, HTML, CSS та JavaScript. Процес розробки включав кілька етапів, зокрема збір вимог, проектування, реалізацію, тестування та розгортання. Для забезпечення якості та надійності програмного додатка було використано декілька методів тестування, включаючи модульне тестування, інтеграційне тестування, системне тестування та приймальне тестування. Процес

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 06-10.БР</i>			
Зав. каф.	Криворучко О.В.			28.04.23	Програмний додаток підприємства з реалізації спортивного харчування	Стадія	Аркуш	Аркуші
Керівник	Власенко Л.О.			28.04.23		ВП	38	50
Гарант	Рзаєва С.Л.			28.04.23		Факультет інформаційних технологій 4 курс, 6 група		
Розробив	Доля О.С.			28.04.23				
					<i>Висновки та пропозиції</i>			

тестування був ретельним і допоміг виявити та вирішити будь-які проблеми або помилки в програмному забезпеченні.

В цілому, розробка програмного додатка для торгової компанії, що спеціалізується на спортивному харчуванні, була успішною, і очікується, що програмний додаток підвищить ефективність і прибутковість бізнесу. Використання сучасних програмних технологій та методологій розробки дозволило створити високоякісний програмний продукт, який відповідає потребам бізнесу та його клієнтів. Оскільки ринок спортивного харчування продовжує зростати, очікується, що програмний додаток відіграватиме важливу роль в успіху торгової компанії.



						Аркуш
					ДТЕУ 121 06-10.БР	39
Зм.	Аркуш	№ докум	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Sommerville, I. (2016). Software Engineering. Pearson Education Limited. (Pages: 800)
2. Pressman, R. S. (2014). Software engineering: a practitioner's approach. McGraw-Hill Education. (Pages: 936)
3. Martin, R. C. (2003). Agile software development: principles, patterns, and practices. Pearson Education. (Pages: 528)
4. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design patterns: elements of reusable object-oriented software. Pearson Education. (Pages: 395)
5. Fowler, M. (2002). UML distilled: a brief guide to the standard object modeling language. Pearson Education. (Pages: 192)
6. Beck, K. (2000). Extreme programming explained: embrace change. Addison-Wesley Professional. (Pages: 224)
7. Shalloway, A., & Trott, J. R. (2004). Design patterns explained: a new perspective on object-oriented design. Pearson Education. (Pages: 480)
8. Larman, C. (2004). Agile and iterative development: a manager's guide. Addison-Wesley Professional. (Pages: 704)
9. Cockburn, A. (2008). Agile software development: the cooperative game. Addison-Wesley Professional. (Pages: 504)
10. McConnell, S. (2004). Code complete. Microsoft Press. (Pages: 960)
11. Ambler, S. W. (2002). Agile modeling: effective practices for extreme programming and the unified process. Wiley. (Pages: 384)
12. Highsmith, J. A. (2002). Agile software development ecosystems. Addison-Wesley Professional. (Pages: 400)

					<i>ДТЕУ 121 06-10.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		23.12.22	Програмний додаток підприємства з реалізації спортивного харчування	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Власенко Л.О.		23.12.22		<i>СВД</i>	<i>40</i>	<i>50</i>
Гарант		Рзаєва С.Л.		23.12.22		<i>Факультет інформаційних технологій 4 курс, 6 група</i>		
Розробив		Доля О.С.		23.12.22				
					<i>Список використаних джерел</i>			

13. Schwaber, K., & Beedle, M. (2001). Agile software development with scrum. Prentice Hall. (Pages: 176)
- Sutherland, J. (2013). Scrum: the art of doing twice the work in half the time. Random House. (Pages: 256)
14. Fowler, M., & Highsmith, J. (2001). The agile manifesto. Agile Alliance. (Pages: 16)
15. Ambler, S. W., & Lines, M. (2006). Disciplined agile delivery: a practitioner's guide to agile software delivery in the enterprise. IBM Press. (Pages: 544)
16. Babar, M. A., & Zhu, L. (Eds.). (2016). Agile software architecture: aligning agile processes and software architectures. Morgan Kaufmann. (Pages: 320)
17. Bass, L., Clements, P., & Kazman, R. (2012). Software architecture in practice. Addison-Wesley Professional. (Pages: 768)
18. Cohn, M. (2005). Agile estimating and planning. Prentice Hall. (Pages: 368)
19. Derby, E., & Larsen, D. (2006). Agile retrospectives: making good teams great. Pragmatic Bookshelf. (Pages: 176)
20. Holzmann, G. J. (2003). The power of 10: rules for developing safety-critical code. ACM Press. (Pages: 144)
21. Jones, C. (2011). Applied software measurement: global analysis of productivity and quality. McGraw Hill Professional. (Pages: 656)
22. Lamsweerde, A. V. (2001). Requirements engineering: from system goals to UML models to software specifications. John Wiley & Sons. (Pages: 704)
23. Larman, C., & Basili, V. R. (2003). Iterative and incremental development: a brief history. IEEE Computer, 36(6), 47-56. (Pages: 10)

					<i>ДТЕУ 121 06-10.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		41

## ДОДАТКИ

### ДОДАТОК А

```
from flask import render_template, request, redirect, url_for

from src import app

from src.models.products import get_products, create_product, update_product, get_product, delete_product

from src.models.categories import get_categories

@app.route("/")
def index():
    return redirect(url_for("products_page"))

@app.route("/products")
def products_page():
    return render_template("products/index.html", products=get_products())

@app.route("/create_product", methods=["GET", "POST"])
def create_product_page():
    if request.method == "POST":
        title = request.form["title"]
        descr = request.form["descr"]
        price = request.form["price"]
        img = request.form["img"]
        cat_id = request.form["category"]

        create_product(title, descr, price, img, cat_id)
    return render_template("products/create.html", categories=get_categories())

@app.route("/update_product/<int:product_id>", methods=["GET", "POST"])
def update_product_page(product_id):
    if request.method == "POST":
        title = request.form["title"]
        descr = request.form["descr"]
        price = request.form["price"]
        img = request.form["img"]
        cat_id = request.form["category"]

        update_product(product_id, title, descr, price, img, cat_id)

    return render_template("products/update.html", product=get_product(product_id), categories=get_categories())

@app.route("/delete_product/<int:product_id>", methods=["GET", "POST"])
def delete_product_page(product_id):
    if request.method == "POST":
        delete_product(product_id)
```

```

return redirect(url_for("products_page"))

return render_template("products/delete.html")
Програмний код контролера категорій
from flask import render_template, request, redirect, url_for
from src import app
from src.models.categories import get_categories, create_category, update_category, get_category, delete_category

@app.route("/categories")
def categories_page():
    return render_template("categories/index.html", categories=get_categories())

@app.route("/create_category", methods=["GET", "POST"])
def create_category_page():
    if request.method == "POST":
        name = request.form["name"]
        create_category(name)

    return render_template("categories/create.html")

@app.route("/update_category/<int:category_id>", methods=["GET", "POST"])
def update_category_page(category_id):
    if request.method == "POST":
        name = request.form["name"]
        update_category(category_id, name)
    return render_template("categories/update.html", category=get_category(category_id))

@app.route("/delete_category/<int:category_id>", methods=["GET", "POST"])
def delete_category_page(category_id):
    if request.method == "POST":
        delete_category(category_id)
        return redirect(url_for("categories_page"))

    return render_template("categories/delete.html")

```

Програмний код контролера замовлень

```

from flask import render_template, request, redirect, url_for

from src import app
from src.models.orders import get_order, get_orders, create_order, update_order, delete_order
from src.models.products import get_products

@app.route("/orders")
def orders_page():
    return render_template("orders/index.html", orders=get_orders())

@app.route("/create_order", methods=["GET", "POST"])
def create_order_page():
    if request.method == "POST":

```



```

name = request.form["name"]
phone = request.form["phone"]
products = list(request.form.listvalues())[2]

create_order(name, phone, products)
return render_template("orders/create.html", products=get_products())

@app.route("/update_order/<int:order_id>", methods=["GET", "POST"])
def update_order_page(order_id):
    if request.method == "POST":
        name = request.form["name"]
        phone = request.form["phone"]
        products = list(request.form.listvalues())[2]

        update_order(order_id, name, phone, products)

    order = get_order(order_id)

    return render_template("orders/update.html", products=get_products(), order=order,
                           selected_products=[int(i) for i in order[4].split(",")])

@app.route("/delete_order/<int:order_id>", methods=["GET", "POST"])
def delete_order_page(order_id):
    if request.method == "POST":
        delete_order(order_id)
        return redirect(url_for("orders_page"))

    return render_template("orders/delete.html")

```

Програмний код контролера робітників

```

from flask import render_template, request, redirect, url_for
from src import app
from src.models.employees import get_employees, create_employee, update_employee, get_employee, delete_employee
from src.models.positions import get_positions

@app.route("/employees")
def employees_page():
    return render_template("employees/index.html", employees=get_employees())

@app.route("/create_employee", methods=["GET", "POST"])
def create_employee_page():
    if request.method == "POST":
        name = request.form["name"]
        surname = request.form["surname"]
        phone = request.form["phone"]
        pos_id = request.form["position"]

```

```

create_employee(name, surname, phone, pos_id)

return render_template("employees/create.html", positions=get_positions())

@app.route("/update_employee/<int:employee_id>", methods=["GET", "POST"])
def update_employee_page(employee_id):
    if request.method == "POST":
        name = request.form["name"]
        surname = request.form["surname"]
        phone = request.form["phone"]
        pos_id = request.form["position"]

        update_employee(employee_id, name, surname, phone, pos_id)

    return render_template("employees/update.html", positions=get_positions(), employee=get_employee(employee_id))

@app.route("/delete_employee/<int:employee_id>", methods=["GET", "POST"])
def delete_employee_page(employee_id):
    if request.method == "POST":
        delete_employee(employee_id)
        return redirect(url_for("employees_page"))

    return render_template("employees/delete.html")

```

Програмний код контролера позицій

```

from flask import render_template, request, redirect, url_for

from src import app

from src.models.positions import get_positions, get_position, create_position, update_position, delete_position

@app.route("/positions")
def positions_page():
    return render_template("positions/index.html", positions=get_positions())

@app.route("/create_position", methods=["GET", "POST"])
def create_position_page():
    if request.method == "POST":
        name = request.form["name"]
        create_position(name)

    return render_template("positions/create.html")

@app.route("/update_position/<int:position_id>", methods=["GET", "POST"])
def update_position_page(position_id):
    if request.method == "POST":
        name = request.form["name"]
        update_position(position_id, name)

    return render_template("positions/update.html", position=get_position(position_id))

```

```

@app.route("/delete_position/<int:position_id>", methods=["GET", "POST"])
def delete_position_page(position_id):
    if request.method == "POST":
        delete_position(position_id)
        return redirect(url_for("positions_page"))

    return render_template("positions/delete.html")

```

Програмний код моделі продуктів

```

from src import get_db
def create_product(title, descr, price, img, cat_id):
    get_db().execute("""
        INSERT INTO products(title, description, price) VALUES(?, ?, ?)""", (title, descr, price))

    img_cur = get_db().execute("""
        SELECT * FROM images
        WHERE url = ?""", (img,))

    if len(img_cur.fetchall()) < 1:
        get_db().execute("""
            INSERT INTO images(url) VALUES(?)"", (img,))
        get_db().commit()

    img_cur = get_db().execute("""
        SELECT * FROM images
        WHERE url = ?""", (img,))

    img_id = img_cur.fetchall()[0][0]
    img_cur.close()

    prod_cur = get_db().execute("""
        SELECT id FROM products ORDER BY id DESC LIMIT 1""")
    product_id = prod_cur.fetchall()[0][0]
    prod_cur.close()

    get_db().execute("""
        INSERT INTO products_images
        (product_id, image_id)
        VALUES(?, ?)""", (product_id, img_id))

    get_db().execute("""
        INSERT INTO products_categories
        (category_id, product_id)
        VALUES(?, ?)""", (cat_id, product_id))

    get_db().commit()

```

```

def update_product(product_id, title, descr, price, img, cat_id):
    get_db().execute("""
        UPDATE products
        SET title = ?,
            description = ?,
            price = ?
        WHERE id = ?""", (title, descr, price, product_id))

    img_cur = get_db().execute("""
        SELECT * FROM images
        WHERE url = ?""", (img,))

    if len(img_cur.fetchall()) < 1:
        get_db().execute("""
            INSERT INTO images(url) VALUES(?)"", (img,))
        get_db().commit()

    img_cur = get_db().execute("""
        SELECT * FROM images
        WHERE url = ?""", (img,))
    img_id = img_cur.fetchall()[0][0]

    get_db().execute("""
        UPDATE products_images
        SET image_id = ?
        WHERE product_id = ?""", (img_id, product_id))

    get_db().execute("""
        UPDATE products_categories
        SET category_id = ?
        WHERE product_id = ?""", (cat_id, product_id))

    get_db().commit()

def delete_product(product_id):
    get_db().execute("DELETE FROM products WHERE id = ?", (product_id,))
    get_db().commit()

def get_products():
    products_cur = get_db().execute("""
        SELECT
        products.id,
        products.title,
        products.description,
        products.price,
    """)

```

```

images.url,
categories.name
FROM products
LEFT JOIN products_images
    ON products_images.product_id = products.id
LEFT JOIN products_categories
    ON products_categories.product_id = products.id
LEFT JOIN images
    ON images.id = products_images.image_id
LEFT JOIN categories
    ON categories.id = products_categories.category_id"""

products = products_cur.fetchall()
products_cur.close()

return products

def get_product(product_id):
    product_cur = get_db().execute("""
SELECT
products.id,
products.title,
products.description,
products.price,
images.url,
categories.id
FROM products
LEFT JOIN products_images
    ON products_images.product_id = products.id
LEFT JOIN products_categories
    ON products_categories.product_id = products.id
LEFT JOIN images
    ON images.id = products_images.image_id
LEFT JOIN categories
    ON categories.id = products_categories.category_id
WHERE products.id = ?""", (product_id,))

    product = product_cur.fetchall()
    product_cur.close()

    return product[0]

```

Програмний код моделі категорій

```

from src import get_db

def create_category(name):
    get_db().execute("""
INSERT INTO categories(name) VALUES(?)"", (name,))

```

```

get_db().commit()

def update_category(category_id, name):
    get_db().execute("""
        UPDATE categories SET name = ? WHERE id = ?""", (name, category_id))

    get_db().commit()

def delete_category(category_id):
    get_db().execute("DELETE FROM categories WHERE id = ?", (category_id,))
    get_db().commit()

def get_categories():
    categories_cur = get_db().execute("SELECT * FROM categories")
    categories = categories_cur.fetchall()
    categories_cur.close()

    return categories

def get_category(category_id):
    category_cur = get_db().execute("SELECT * FROM categories WHERE id = ?", (category_id,))
    category = category_cur.fetchall()
    category_cur.close()

    return category[0]

```

Програмний код моделі замовлень

```

from src import get_db

def create_order(name, phone, products):
    get_db().execute("""
        INSERT INTO orders(customer_name, customer_phone) VALUES(?, ?)""", (name, phone))

    order_id_cur = get_db().execute("""
        SELECT * FROM orders ORDER by id DESC LIMIT 1""")

    order_id = order_id_cur.fetchall()[0][0]
    order_id_cur.close()

    for product_id in products:
        get_db().execute("INSERT INTO orders_products VALUES (?, ?)", (order_id, product_id))

    get_db().commit()

def update_order(order_id, name, phone, products):
    get_db().execute("""

```

```
UPDATE orders
```

```
SET customer_name = ?,
```

```
customer_phone = ?""", (name, phone))
```

```
get_db().execute("DELETE FROM orders_products WHERE order_id = ?", (order_id,))
```

```
for product_id in products:
```

```
get_db().execute("INSERT INTO orders_products VALUES (?, ?)", (order_id, product_id))
```

```
get_db().commit()
```

```
def delete_order(order_id):
```

```
get_db().execute("DELETE FROM orders WHERE id = ?", (order_id,)) get_db().commit()
```

```
def get_orders():
```

```
orders_cur = get_db().execute("""
```

```
SELECT
```

```
orders.id,
```

```
orders.customer_name,
```

```
orders.customer_phone,
```

```
orders.order_date,
```

```
group_concat(products.title) as "products"
```

```
FROM orders
```

```
JOIN orders_products
```

```
ON orders_products.order_id = orders.id
```

```
JOIN products
```

```
ON products.id = orders_products.product_id
```

```
GROUP BY orders.id""")
```

```
orders = orders_cur.fetchall()
```

```
orders_cur.close()
```

```
return orders
```

```
def get_order(order_id):
```

```
orders_cur = get_db().execute("""
```

```
SELECT
```

```
orders.id,
```

```
orders.customer_name,
```

```
orders.customer_phone,
```

```
orders.order_date,
```

```
group_concat(products.id) as "products"
```

```
FROM orders
```

```
JOIN orders_products
```

```
ON orders_products.order_id = orders.id
```

```

JOIN products
ON products.id = orders_products.product_id
WHERE orders.id = ?
GROUP BY orders.id""", (order_id,))

order = orders_cur.fetchall()[0]
orders_cur.close()

return order
Програмний код моделі робітників
from src import get_db

def create_employee(name, surname, phone, pos_id):
    get_db().execute("""
        INSERT INTO employees(first_name, last_name, phone) VALUES(?, ?, ?)""", (name, surname, phone))

    emp_cur = get_db().execute("""
        SELECT id FROM employees ORDER BY id DESC LIMIT 1""")
    emp_id = emp_cur.fetchall()[0][0]
    emp_cur.close()

    get_db().execute("""
        INSERT INTO employees_positions
        VALUES(?, ?)""", (emp_id, pos_id))

    get_db().commit()

def update_employee(employee_id, name, surname, phone, pos_id):
    get_db().execute("""
        UPDATE employees
        SET first_name = ?,
            last_name = ?,
            phone = ?
        WHERE id = ?""", (name, surname, phone, employee_id))

    get_db().execute("""
        UPDATE employees_positions
        SET position_id = ? WHERE employee_id = ?""", (pos_id, employee_id))

    get_db().commit()

def delete_employee(employee_id):
    get_db().execute("DELETE FROM employees WHERE id = ?", (employee_id,))
    get_db().commit()

```



```

def get_employees():
    employees_cur = get_db().execute("""
    SELECT
        employees.id,
        employees.first_name,
        employees.last_name,
        employees.phone,
        positions.name
    FROM employees
    LEFT JOIN employees_positions
        ON employees_positions.employee_id = employees.id
    LEFT JOIN positions
        ON positions.id = employees_positions.position_id""")

    employees = employees_cur.fetchall()
    employees_cur.close()

    return employees

```

```

def get_employee(employee_id):
    employees_cur = get_db().execute("""
    SELECT
        employees.id,
        employees.first_name,
        employees.last_name,
        employees.phone,
        positions.id
    FROM employees
    LEFT JOIN employees_positions
        ON employees_positions.employee_id = employees.id
    LEFT JOIN positions
        ON positions.id = employees_positions.position_id
    WHERE employees.id = ?""", (employee_id,))

    employees = employees_cur.fetchall()
    employees_cur.close()

    return employees[0]

```

Програмний код моделі позицій  
from src import get\_db

```

def create_position(name):
    get_db().execute("""
        INSERT INTO positions(name) VALUES(?,?)", (name,))

    get_db().commit()

```

```
def update_position(position_id, name):
    get_db().execute("""
        UPDATE positions SET name = ? WHERE id = ?""", (name, position_id))
    get_db().commit()

def delete_position(position_id):
    get_db().execute("DELETE FROM positions WHERE id = ?", (position_id,))
    get_db().commit()

def get_positions():
    positions_cur = get_db().execute("SELECT * FROM positions")
    positions = positions_cur.fetchall()
    positions_cur.close()

    return positions

def get_position(position_id):
    position_cur = get_db().execute("SELECT * FROM positions WHERE id = ?", (position_id,))
    position = position_cur.fetchall()
    position_cur.close()

    return position[0]
```