

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра

системного аналізу та управління

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи ступеню магістра

(бакалавра, магістра)

студента Владимирова Якіма Дмитровича

(ПІБ)

академічної групи 124М-21-1

(шифр)

спеціальності 124 - Системний аналіз

(код і назва спеціальності)

на тему «Аналіз способів перевезень товарів меблевого маркетплейсу»

(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	к.ф.-м.н., доц Хом'як Т.В.			
розділів:				
Інформаційно-аналітичний	к.ф.-м.н., доц Хом'як Т.В.			
Спеціальний	к.ф.-м.н., доц Хом'як Т.В.			

Рецензент	д.т.н., проф. Алексєєв М.О.			
-----------	-----------------------------	--	--	--

Нормоконтролер	к.ф.-м.н., доц Хом'як Т.В.			
----------------	----------------------------	--	--	--

Дніпро

2022

ЗАТВЕРДЖЕНО:

завідувач кафедри

Системного аналізу та

управління

(повна назва)

Желдак Т.А.

(підпис)

(прізвище, ініціали)

« _____ » _____ 20__ року

Завдання**на кваліфікаційну роботу****ступеня магістра**

(бакалавра, магістра)

студенту Владимирову Я.Д. академічної групи 124М-21-1
(прізвище та ініціали) (шифр)спеціальності 124 - Системний аналізна тему «Аналіз способів перевезень товарів меблевого маркетплейсу»Затверджена наказом ректора НТУ «Дніпровська політехніка» від 31.10.2022 р. № 1200-с

Розділ	Зміст	Термін виконання
Інформаційно-аналітичний	Вибір, аналіз та обґрунтування методів системного аналізу, а також сервісів та технологій розробки програмного забезпечення.	28.10.2022 – 17.10.2022
Спеціальний	Аналіз меблевих перевезень та веб-сайтів електронної комерції з ціллю розробки програмного забезпечення, вибору серверу і прогнозування статистики для реалізації бізнес-проекту з організації системи доставок з альтернативами на сайті з оптимізацією маршрутів і детальною схематизацією.	17.10.2022 – 05.10.2022

Завдання видано

(підпис керівника)

Хом'як Т.В.

(прізвище, ініціали)

Дата видачі 31.10.2022 р.Дата подання до екзаменаційної комісії 16.12.2022 р.**Прийнято до виконання**

(підпис студента)

Владимиров Я.Д.

(прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка 102 с., 19 рисунків, 23 таблиці, 4 додатки, 63 джерела.

Об'єкт дослідження: електронно-комерційна і транспортна діяльність меблевої галузі.

Предмет дослідження: методи системного аналізу з прийняття рішень на основі багатьох критеріїв, рішень щодо вибору стратегії, рішень конфліктних ситуацій, технології розробки програмного забезпечення, методи прогнозування даних, лінійне програмування та нотації побудови схем.

Мета дослідження: аналіз транспортних систем та поштових служб з метою реалізації інтернет-маркетплейсу та системи доставок, вибору серверу на основі багатьох критеріїв та стратегії його управління, а також вибору методу прогнозування статистики та оптимізації маршрутів перевезень.

В *інформаційно-аналітичному розділі* проведено вибір, обґрунтування та опис необхідних методів системного аналізу, а також технологій та сервісів, необхідних для процесу розробки веб-додатку.

В *спеціальному розділі* змодельовані та розв'язані задачі на основі аналізу меблевої промисловості, транспортних систем та веб-сайтів електронної комерції, а саме: побудовано бізнес-проект, спроектовано базу даних, обрано сервер, обрано стратегію управління сервером, розроблено програмне забезпечення, оптимізовано маршрути перевезень, схематизовано процеси доставки, а також виконано аналіз методів прогнозування даних.

Практична цінність отриманих у роботі результатів полягає в оптимізації та автоматизації процесу продажу товарів меблевої промисловості, а також розширення його функціоналу, що призведе до збільшення масштабів компанії та популярності (і прибутковості) сервісу.

Ключові слова: СИСТЕМНИЙ АНАЛІЗ, ПРОГРАМНЕ ЗАБЕСПЕЧЕННЯ, КЛІЄНТ-СЕРВЕР, МЕТОД АНАЛІЗУ ІЕРАРХІЙ, МЕТОДИ ПРИЙНЯТТЯ РІШЕНЬ, НЕЧІТКІ МОДЕЛІ, БАЗИ ДАНИХ,

НОРМАЛІЗАЦІЯ, ВЕБ-ТЕХНОЛОГІЇ, ПРОГНОЗУВАННЯ ДАНИХ,
МАТРИЧНІ ІГРИ, ТРАНСПОРТНА ЗАДАЧА, UML, EPC, USE-CASE.

ABSTRACT

Explanatory note 102 pp., 19 figures, 23 tables, 4 appendices, 63 sources.

Object of research: e-commerce and transport activities of the furniture industry.

Subject of research: systems analysis methods for multi-criteria decision-making, strategy selection decisions, conflict resolution, software development technologies, data forecasting methods, linear programming, and notations of circuit construction.

Purpose of research: the analysis of transport systems and postal services for the purpose of implementing an Internet marketplace and delivery system, choosing a server based on many criteria and its management strategy, as well as choosing a method of forecasting statistics and optimizing transportation routes.

In the *information and analytical section*, the selection, justification and description of the necessary methods of system analysis, as well as technologies and services necessary for the process of developing a web application, were carried out.

In the *special section*, problems based on the analysis of the furniture industry, transportation systems and e-commerce websites are modeled and solved, namely: a business project is built, a database is designed, a server is selected, a server management strategy is selected, software is developed, routes are optimized transportation, delivery processes are diagrammed, and data forecasting methods are analyzed.

The *practical value* of the results obtained in the work consists in optimizing and automating the process of selling furniture industry goods, as well as expanding its functionality, which will lead to an increase in the scale of the company and the popularity (and profitability) of the service.

Keywords: SYSTEM ANALYSIS, SOFTWARE, CLIENT-SERVER, HIERARCHY ANALYSIS METHOD, DECISION-MAKING METHODS, FUZZY MODELS, DATA BASES, NORMALIZATION, WEB

TECHNOLOGIES, DATA FORECASTING, MATRIX GAMES,
TRANSPORTATION PROBLEM, EPC, UML, USE-CASE.

ЗМІСТ

ВСТУП	10
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	12
1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ РОЗДІЛ	13
1.1 Системний аналіз	13
1.2 Прийняття рішень в умовах невизначеності	16
1.3 Метод аналізу ієрархій.....	17
1.4 Методи прийняття рішень в умовах невизначеності.....	19
1.5 Прогнозування даних.....	20
1.6 Бази даних.....	21
1.7 Нормалізація реляційних баз даних	24
1.8 Системи управління базами даних.....	25
1.9 Вибір технологій, сервісів та бібліотек	26
1.10 Обґрунтування обраних технологій, сервісів та бібліотек	27
1.11 Програмна структура веб-додатків Django	29
1.12 Презентація бізнес-проекту	31
1.13 Теорія ігор у конфліктній ситуації	33
1.14 Транспортні задачі	35
1.15 Нотація UML	37
2 СПЕЦІАЛЬНИЙ РОЗДІЛ	40
2.1 Проектування реляційної моделі бази даних.....	40
2.1.1 ER-проектування.....	40
2.1.2 Опис результуючої моделі	41
2.1.3 Проектування реляційної структури.....	42
2.1.4 Моделювання SQL-запитів	46
Висновок	48
2.2 Проектування та розробка back-end.....	50
2.2.1 Структура серверної частини веб-додатку.....	50
2.2.2 З'єднання з клієнтською частиною та розгортання на сервері	51
Висновок	51

2.3 Вибір алгоритмів аналізу та прогнозування даних	52
2.3.1 Обчислення і аналіз показників популярності і рейтингу.....	52
2.3.2 Вибір алгоритму прогнозування статистики товару	53
2.3.3 Реалізація прогнозування у python для автоматизації процесу	55
Висновок	55
2.4 Вибір серверу та управління	57
2.4.1 Змістова модель задачі	57
2.4.2 Концептуальна модель задачі	57
2.4.3 Математична модель задачі.....	57
2.4.4 Вибір серверу методом аналізу ієрархій.....	59
2.4.5 Вибір стратегії методами прийняття рішень за невизначеності	61
2.4.6 Аналіз отриманих результатів	63
Висновок	63
2.5. Презентація бізнес-проекту з розширення функціоналу сервісу	64
2.5.1. Визначення проекту.....	64
2.5.2. Обґрунтування проекту	65
2.5.3. Визначення альтернатив рішень: поштові служби	65
2.5.4. Визначення альтернатив рішень: відділ перевезень	69
2.5.5. Презентація проекту	70
Висновок	73
2.6. Теорія ігор у конфліктній ситуації сервісу і користувача	74
2.6.1. Постанова задачі	74
2.6.2. Обґрунтування застосування методу матричних ігор	75
2.6.3. Побудова моделі та визначення вхідних даних	76
2.6.4. Обчислення матричної гри	79
2.6.5. Аналіз результатів матричної гри	80
Висновок	81
2.7 Транспортна задача для оптимізації перевезень.....	83
2.7.1 Постановка задачі	83
2.7.2 Обґрунтування вибору методу для рішення даної задачі.....	84

2.7.3 Побудова моделі та визначення вхідних даних.....	84
2.7.4 Обчислення транспортної задачі.....	86
2.7.5 Аналіз отриманих результатів	91
Висновок.....	91
2.8 Схематизація та побудова сценаріїв роботи служби доставки	93
2.8.1 Вибір типів схем та обґрунтування.....	93
2.8.2 Нотація EPC. Система доставки та її процеси.....	94
2.8.3 Нотація UML. Випадки використання процесів доставки	97
2.8.4 Нотація UML. Діаграми діяльності процесів доставки	99
2.8.5 Use-Case Сценарії замовлення через веб-додаток.....	100
Висновок.....	104
ВИСНОВКИ.....	105
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	107
ДОДАТКИ.....	112

ВСТУП

У час активного розвитку інтернет-технологій онлайн-транзакції швидко набирають свою популярність. Інтернет-магазини майже замінили торгові центри, і їх кількість продовжує рости. Одним з популярних видів інтернет-магазинів є інтернет-маркетплейси, такі як Amazon, AliExpress, тощо.

Маркетплейс – це великий інтернет-магазин, який дає можливість іншим продавцям реалізовувати свої товари за певну винагороду. Найчастіше, маркетплейси вкладають величезні гроші в просування свого майданчика, завдяки чому у них багато трафіку, а значить - клієнтів і партнерів, які хочуть скористатися можливостями маркетплейсу в свою користь.

Все маркетплейси різні, але є кілька обов'язкових вимог, яким вони повинні відповідати, щоб так називатися:

- є основна організація, яка контролює всі процеси;
- на майданчику представлені товари інших продавців і продуманий алгоритм, завдяки якому вони починають торгувати;
- прописані умови співпраці: комісія, вимоги до контенту та якості обслуговування, тощо.

Структури більшості маркетплейсів схожі. У них обов'язково повинен бути каталог товарів, форма для реєстрації продавців, особисті кабінети продавців і покупців, наявність статистики продажів та ін., можливість вказати способи оплати та доставки, а також підтримка відгуків [1].

У рамках даної роботи розглянуто серверну частину меблевого маркетплейсу і його систему доставок, та пов'язані з ними процеси та задачі.

Економічний ефект від реалізації результатів роботи очікується за рахунок надходжень від використання системи доставок на платформі.

Соціальний ефект очікується за рахунок підвищення популярності платформи за рахунок підвищення якості та зручності.

Результати мають відповідати вимогам стандарту НТУ «ДП» «Методичні рекомендації до виконання кваліфікаційної роботи магістра

галузі знань 12 Інформаційні технології спеціальності 124 Системний аналіз» [2]. Результати досліджень мають дозволяти їх безпосереднє використання.

Апробація результатів досліджень:

– VI Міжнародна Науково-Технічна Конференція (04-06 Листопада 2020 р., м. Дніпро, ДВНЗ УДХТУ);

– VI Всеукраїнська Інтернет-конференція (17-19 Березня 2021 р., м. Дніпро);

– Міжнародна наукова інтернет-конференція (випуск 70), (м. Тернопіль, Україна – м. Переворськ, Польща, 22-23 вересня 2022 р.);

– XI Міжнародна науково-практична конференція (12-14 грудня 2022 р., м. Запоріжжя).

Отримані наукові результати відображаються у публікаціях:

1. Владимиров Я.Д., Козир С.В. Імітаційне моделювання процесу ціноутворення // VI Міжнародної науково-технічної конференції «Комп'ютерне Моделювання та Оптимізація Складних Систем», 04-06 Листопада 2020 р., м. Дніпро, ДВНЗ УДХТУ – 29-30 с.

2. Владимиров Я.Д., Хом'як Т.В. Вибір та управління серверним обладнанням для роботи з веб-додатком // IV Всеукраїнська Інтернет-конференція здобувачів вищої освіти і молодих учених «Інформаційні технології: теорія і практика» (17-19 березня 2021 р.). Збірник тез. Дніпро. НТУ "ДП". – 18-21 с.

3. Владимиров Я.Д., Хом'як Т.В. Організація доставки товару інтернет-маркетплейсу // Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 70): матеріали Міжнародної наукової інтернет-конференції, (м. Тернопіль, Україна – м. Переворськ, Польща, 22-23 вересня 2022 р.); ГО “Наукова спільнота” – 18-21 с.

4. Владимиров Я.Д., Хом'як Т.В. Схематизація системи доставок товару інтернет-маркетплейсу // XI Міжнародна науково-практична конференція «Сучасні проблеми та досягнення радіотехніки, телекомунікацій та інформаційних технологій», 12-14 грудня 2022 р., м. Запоріжжя.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

ТПР – теорія прийняття рішень

ОПР – особа, що приймає рішення

ПЗ – програмне забезпечення

СУБД – система управління базами даних

1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ РОЗДІЛ

1.1 Системний аналіз

Системний аналіз – науковий метод пізнання, що являє собою послідовність дій з установлення структурних зв'язків між змінними або елементами досліджуваної системи. Спирається на комплекс загальнонаукових, експериментальних, природничих, статистичних, математичних методів.

Системний аналіз виник в епоху застосування комп'ютерної техніки у бізнесі та інших сферах діяльності. Успіх його застосування у вирішенні складних завдань багато в чому визначають можливості сучасних інформаційних технологій. Результатом системних досліджень є, як правило, вибір цілком певної альтернативи плану розвитку регіону, параметрів конструкції тощо. Тому витoki системного аналізу, його методичні концепції лежать в тих дисциплінах, які займаються проблемами прийняття рішень: теорії операцій і загальної теорії управління [3].

Системний аналіз передбачає розгляд об'єктів як систем, переважно цілеспрямованих. Основні методологічні засади системного аналізу базуються на принципах системного підходу.

Системний аналіз спрямований на розв'язання складних проблем. Проблема виникає тоді, коли є розходження між бажаним та дійсним, тобто це абстрактна категорія, що відображає розуміння людьми мотивів своєї діяльності. Проблеми породжуються та розв'язуються людьми, а тому поняття «проблема» має людські риси сприйняття, що породжує наступні труднощі:

- неясність розуміння проблеми;
- складнощі постановки проблем на віддалену перспективу;
- складність класифікації проблем і, як наслідок, вибір неадекватних засобів їх розв'язання;

- спотворена оцінка проблем (близькі, але дрібні проблеми затуляють великі, але віддалені);
- неправильна оцінка значимості проблем внаслідок вузькопрофесійної точки зору;
- змішування цілей, які необхідно досягнути, з засобами їх досягнення.

Метою застосування системного аналізу до конкретної проблеми є підвищення ступеня обґрунтованості рішення, що приймається. Для системного аналізу важливими є наступні методологічні принципи:

- органічна єдність суб'єктивного та об'єктивного;
- структурність системи, що визначає цілісність та стійкість характеристик системи;
- динамізм систем;
- міждисциплінарний характер системних досліджень;
- органічна єдність формального та неформального при проведенні системного аналізу.

На відміну від доволі широкої системної методології системний аналіз обмежують дві наступні особливості:

- системні аналітики вивчають лише штучно створені системи, в яких людині належить надзвичайно важлива, а в багатьох випадках і вирішальна роль;
- головна задача системного аналізу – прийняття рішень і управління.

Системний аналіз застосовується для розв'язання складних проблем, що пов'язані з діяльністю людей. Людську діяльність умовно можна поділити на дві області: рутинна діяльність, розв'язання регулярних, щоденних завдань, розв'язання нових задач, які виникають вперше.

Окрім того, проблеми розрізняються за ступенем їх структурованості:

- добре структуровані та сформульовані кількісно;
- слабо структуровані, в яких зустрічаються кількісні і якісні оцінки;
- неструктуровані, якісні проблеми.

Основною областю застосування методів системного аналізу є слабо структуровані проблеми, а для розв'язання неструктурованих проблем в більшості застосовуються евристичні методи.

Потреба в системному аналізі виникає в тому випадку, коли виникають наступні ситуації:

- розв'язується нова проблема, і за допомогою системного аналізу вона формулюється, визначається, що та про що потрібно дізнатися, і хто повинен знати;
- розв'язання проблеми передбачає координацію цілей з множиною засобів їх досягнення;
- проблема має розгалужені зв'язки, що викликають віддалені наслідки в різних галузях, і прийняття рішення в таких випадках потребує врахування сукупної ефективності та повних затрат;
- існують варіанти розв'язання проблеми або досягнення взаємно пов'язаного комплексу цілей, які важко порівняти;
- створюються нові складні системи;
- здійснюється вдосконалення, реконструювання виробництва, необхідна реінженерія бізнес-процесів;
- при створенні інформаційних систем та систем керування;
- коли важливі рішення повинні прийматися за наявності невизначеності та ризику та (або) на достатньо віддалену перспективу.

Для забезпечення успіху системного аналізу потрібно:

- застосовувати його у тих випадках, для яких він призначений;
- наявність потреби, зрозумілої мети та (або) призначення;
- відповідальне ставлення як аналітиків, так і організації-замовника;
- наявність будь-якої інформації, досвіду, ідей та уявлень про предмет дослідження;
- відображення в результатах аналізу реального стану справ та реальних шляхів розв'язання проблем, а не «обґрунтування» суб'єктивних рішень;
- наявність ресурсів – експертів, обладнання, грошових засобів;

– аналіз можливого впливу сторонніх побічних факторів (прогноз наукових відкриттів, винаходів, політичної ситуації) [4].

1.2 Прийняття рішень в умовах невизначеності

У процесі прийняття рішень виникають ситуації, які мають той чи інший ступінь невизначеності, а тому якість рішення залежить від повноти врахування всіх чинників, що впливають на його наслідки. Нерідко ці чинники носять суб'єктивний характер, і це стосується як ОПР, так і самого процесу прийняття рішень. Крім того, орган управління не завжди має у своєму розпорядженні всю інформацію, яка необхідна для його обґрунтованих дій.

Таким чином, можна виділити основні труднощі, що виникають у процесі прийняття рішень, а саме:

- наявність великого числа критеріїв, які не завжди погоджені між собою;
- висока міра невизначеності, зумовлена недостатньою інформацією для обґрунтованого прийняття рішення.

Такі ситуації потребують для свого опису спеціального математичного апарату, який забезпечував би можливість врахування такої невизначеності.

Це можуть бути методи теорії ймовірностей, теорії ігор, статистичних рішень, нечітких множин або якісні методи системного аналізу [5].

Процес прийняття рішень в умовах невизначеності повинен бути ітераційним, адаптивним, спрямованим на зменшення середньої похибки оптимального значення характеристичної функції системи.

Для селективного опрацювання даних про ефективність рішень за відсутності математичної моделі системи використовується її числова ідентифікація у просторі стан-дія. Для формування числової моделі системи використовуються різноманітні методи марківського навчання, серед яких за критеріями ефективності, простоти реалізації та популярності застосування

варто виділити групу методів під узагальнюючою назвою “заохочувальне” або “підкріплене” навчання.

Застосування нечіткого логічного виведення дає змогу використати експертну апроксимацію у моделях заохочувального навчання і тим самим до деякої міри компенсувати невизначеність системи прийняття рішень.

Незважаючи на широкий фронт дослідження, системи прийняття рішень з нечіткою логікою мають низку невирішених проблем, пов'язаних з особливостями структурно-функціональної організації агентів, урахування експертних знань у моделях заохочувального навчання, визначення умов збіжності та забезпечення точності методів адаптивного нечіткого логічного виведення [6].

1.3 Метод аналізу ієрархій

Ієрархія – система, що складається з підсистем, що функціонують як ціле на одному рівні і що є складовими системи більш високого рівня, стаючи підсистемами цієї системи.

У найбільш елементарному вигляді ієрархія будується з вершини (цілей – з погляду управління), через проміжні рівні (критерії, від яких залежать наступні рівні) до найнижчого рівня (який зазвичай є переліком альтернатив). Наприклад, при виборі житла слід врахувати наявність певних критеріїв вибору квартири, необхідних для вибору (вартість, площа, комфортність, інфраструктура району, транспорт, стан житла, сусіди, престиж та інші). Цьому прикладу відповідає ієрархія (рисунок 1.1), на першому рівні який знаходиться ціль – «Житло», на другому чинники, що уточнюють ціль, і на кінець, на останньому рівні три різні квартири, які мають бути оцінені по відношенню до критеріїв другого рівня. Існує декілька видів ієрархій:

- домінантні ієрархії – ієрархія з основою у вершині.
- холархії – домінантні ієрархії із зворотним зв'язком.
- китайський ящик (або модулярні ієрархії) – ієрархія, зростаюча в розмірах від простих елементів до усе більш крупних сукупностей.

Ієрархія є повною, якщо кожен елемент заданого рівня функціонує як критерій для всіх елементів нижчого рівня, інакше ієрархія є неповною.

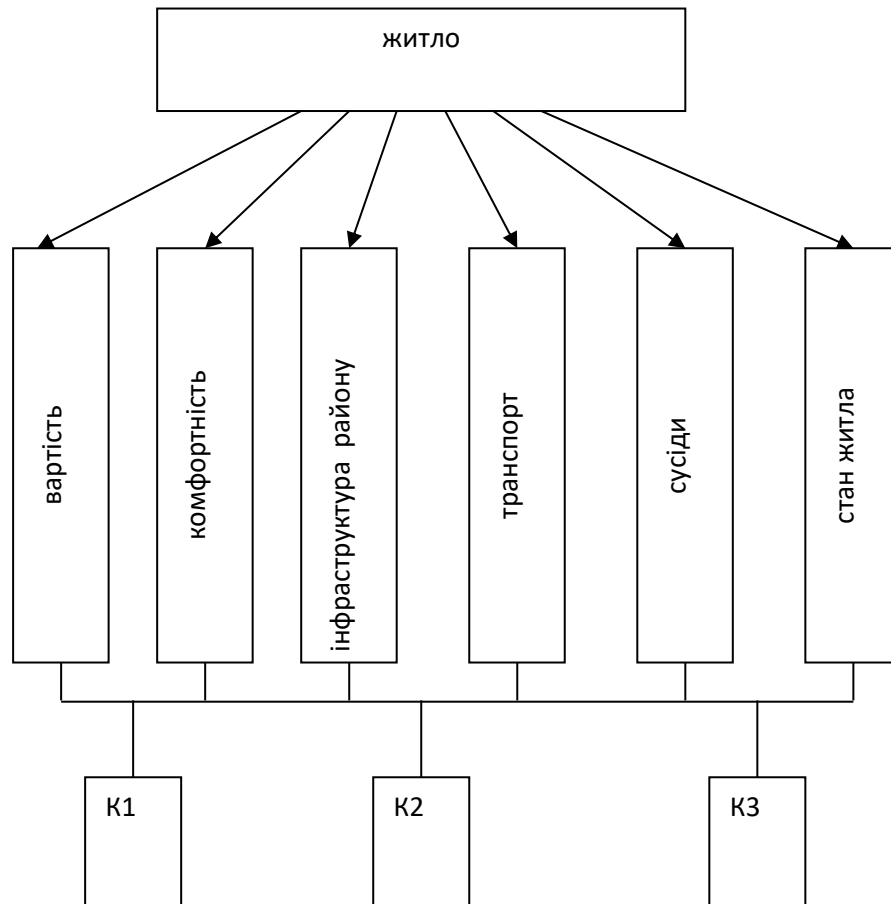


Рисунок 1.1 – Приклад ієрархії

Побудова ієрархії виходить з природної здібності людей думати логічно і творчо, визначати події і встановлювати відношення і спиратися, таким чином, на принцип ідентичності і декомпозиції. На практиці не існує встановленої процедури генерування цілей, критеріїв і видів діяльності для включення в ієрархію. Для вирішення складніших проблем, ієрархія яких не може бути зведена до 3-х або 4-х рівневої структури, можлива наступна їх декомпозиція.

У вершині ієрархії встановлюється єдиний елемент – фокус – формулювання досліджуваної проблеми.

В другий (не обов'язковий) рівень слід включати різні економічні, політичні і соціальні сили, що впливають на результат.

Третій рівень – актори (дійові особи), які реально впливають на ситуацію шляхом маніпулювання цими силами.

Четвертий рівень – переслідувані цілі кожного актора.

П'ятій рівень (не обов'язковий) включає політики акторів, за допомогою яких вони намагаються досягти своїх цілей.

Шостий рівень – альтернативні можливі сценарії або результати, за які береться кожен актор заради досягнення своїх цілей.

Сьомий рівень – узагальнений результат, як результат реалізації і взаємодії можливих альтернативних сценаріїв розвитку проблеми.

При побудові ієрархії слід пам'ятати, що основні цілі існують на вершині ієрархії, їх підцілі безпосередньо нижче за вершину, сили, що обмежують акторів ще нижче. Сили домінують над рівнем самих акторів, які – над рівнем своїх цілей, далі їх можливі дії, а в самому низі лежать можливі результати [7].

1.4 Методи прийняття рішень в умовах невизначеності

При вирішенні статичних моделей прийняття рішень дуже часто застосовують декілька методів прийняття рішень для отримання більш точної відповіді. Необхідно пам'ятати що результати можуть значно відрізнятись в залежності від того, чи відомі вірогідності перебування системи у тому чи іншому стані. У даній роботі будуть розглянуті всі поширені методи прийняття рішень, як з використанням вірогідностей, так і без.

Критерій Байеса (середнього значення). Сенс цього критерію полягає в максимізації математичного сподівання оцінного функціонала. Згідно з критерієм Байеса, оптимальними рішеннями вважають такі, для яких математичне сподівання оцінного функціонала набуває найбільшого (або навпаки) можливого значення. Даний критерій доцільно застосовувати якщо ситуація повторюється багато разів, оскільки за таких умов максимізується середнє значення корисності (або мінімізується середній ризик).

Критерій мінімуму дисперсії оцінного функціонала. Для кожного рішення $f_k \in \Phi$ знайдемо середнє значення V оцінного функціонала й дисперсію $2\sigma_k$. Дисперсія описує розсіювання випадкових значень оцінного функціонала для рішення f_k відносно його середнього значення V .

Основним недоліком цього критерію є те, що дисперсія на одному рішенні може виявитися меншою, ніж на іншому рішенні. Інакше кажучи, критерій мінімуму дисперсії з одного боку є допоміжним, а з іншого – його прийняття потребує довизначення й невеликої зміни вигляду дисперсії.

Критерій максимізації ймовірності розподілу оцінного функціонала. Фіксуємо величину α , яка задовольняє таку умову: $\alpha_1 \leq \alpha \leq \alpha_2$. Для кожного рішення $f_k \in \Phi$ визначають імовірність того, що значення оцінного функціонала буде не меншим за величину α , коли обрано рішення f_k . Сенс критерію полягає у виборі рішення $f_k \in \Phi$, для якого ця ймовірність буде максимальною. При використанні цього критерію орган управління виходить із необхідності прийняття конкретного рівня величини α [8].

1.5 Прогнозування даних

Прогнозування – процес передбачення майбутнього стану предмета чи явища на основі аналізу його минулого і сучасного, систематично оцінювана інформація про якісні й кількісні характеристики розвитку обраного предмета чи явища в перспективі. Результатом прогнозування є прогноз – знання про ймовірний розвиток сьогочасних тенденцій конкретного явища-об'єкту в майбутньому. Прогнозування застосовується в наступних ситуаціях:

- ланцюг постачання – для забезпечення клієнтів компанії правильним продуктом у правильний час;
- бізнес-планування – частина підготовки та розробки бізнес-планів;
- прогноз погоди, метеорологія;
- планування транспорту;
- економічне прогнозування;

- технологічне прогнозування;
- передбачення землетрусів;
- політичне прогнозування;
- прогнозування педагогічне.

Існують два підходи до прогнозування: якісний та кількісний.

Кількісний підхід базується на математичних моделях й історичних даних. Якісний підхід покладається на освічену думку, інтуїцію й досвід професіоналів. Серед його різновидів є консенсус керівництва, Делфі-метод, оцінка торговими працівниками – кожного за своїм регіоном, опитування клієнтів. Кількісні методи діляться на два види: причинно-наслідкові й моделі часових рядів. Часові ряди діляться на:

- моделі з декомпозицією: виділення сезонності й тренду;
- моделі згладжування.

Надзвичайно важливо оцінювати прогноз. Серед безлічі надійних методів є декілька, які найпростішими, але, поєднані разом, дають безпомилкову оцінку якості поданого плану продажу та його фактичного виконання. Це:

- середнє абсолютне відхилення / Mean Average Deviation (MAD);
- сума помилок прогнозу зростаючим підсумком / Running Sum of Forecast Error (RSFE);
- сигнал відслідковування / Tracking Signal (TS) [9-10].

1.6 Бази даних

База даних (далі БД) – це організована структура, яка призначена для зберігання, зміни та обробки взаємозалежної інформації, переважно великих обсягів. БД використовують для динамічних сайтів з великими обсягами (інтернет-магазин, портал, корпоративний сайт).

Кожна таблиця БД – це певний набір даних, які пов'язані між собою спільною ознакою або властивістю, та впорядковані, наприклад, за алфавітом.

Об'єднання великої кількості даних в єдину базу дає змогу для формування безлічі варіації групування інформації – особисті дані клієнта, історія замовлень, каталог товарів та будь-що інше.

Головною перевагою БД є швидкість внесення та використання потрібної інформації. Завдяки спеціальним алгоритмам можна легко знаходити необхідні дані всього за декілька секунд. Також в базі даних існує певний взаємозв'язок інформації: зміна в одному рядку може спричинити зміни в інших рядках – це допомагає працювати з інформацією простіше і швидше. БД для сайтів дають змогу зберігати інформацію, що виглядає як зв'язані між собою таблиці. Саме в БД зберігаються вся необхідна та корисна інформація для функціонування сайту (клієнтські дані, прайс-лист, список товарів).

Щоб створити запит до бази даних часто використовують Structured Query Language. SQL дає змогу додавати, редагувати та видаляти інформацію, що міститься у таблицях [11].

Загалом мову SQL поділяють на дві частини: мову маніпуляції даними (Data Manipulation Language (DML)), та мову опису даних (Data Definition Language (DDL)). Мова SQL не чутлива до регістру, і оператори пишуть великими буквами тільки для зручності.

Мова опису даних дозволяє нам описати структуру даних.

Для створення таблиці використовується команда CREATE TABLE:

```
CREATE TABLE persons (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    surname VARCHAR(50)
);
```

Щоб змінити структуру таблиці використовують команду ALTER:

```
ALTER TABLE persons ADD COLUMN email VARCHAR(100);
```

Крім цього таблиці можна перейменовувати:

```
ALTER TABLE аж RENAME AS new_table;
```

А також видаляти поля:

```
ALTER TABLE persons DROP COLUMN tel;
```

Мова маніпуляції даними дозволяє виконувати операції з даними.

Оператор SELECT дозволяє вибирати дані з бази. Загалом оператор SELECT виглядає так:

```
SELECT імена_полів
FROM імена_таблиць
WHERE умова;
```

Імена полів записуються через кому. Якщо потрібно вибрати всі поля, пишуть зірочку ("*"). При потребі можна уточнити з якої таблиці брати поле, додавши перед його іменем.

Також до запиту SELECT можна додати команду ORDER BY, що дозволяє впорядкувати результат за заданими стовпцями. Щоб сортувати в зворотньому порядку після стовпців за якими сортують пишуть DESC:

```
SELECT name FROM students ORDER BY name DESC;
```

Варто також зауважити, що ORDER BY не може стояти перед WHERE.

З'єднання використовуються для запитів з кількох таблиць, що базуються на зв'язках між певними стовпцями таблиць. З'єднання INNER JOIN (теж саме що і JOIN), де є хоч одне співпадіння в стовпцях таблиці, пишуть так:

```
SELECT стовпці FROM таблиця1 INNER JOIN таблиця2
ON таблиця1.назва_стовпця=таблиця2.назва_стовпця
```

І це буде те ж саме що і:

```
SELECT стовпці FROM таблиця1, таблиця2
WHERE таблиця1.назва_стовпця=таблиця2.назва_стовпця
```

LEFT JOIN працює та пишеться майже так само, але повертає таблицю, в яку входять всі записи лівої таблиці (недостаючі записи з правої заповнюються NULL-ами). RIGHT JOIN відповідно навпаки.

FULL JOIN повертає об'єднання результатів RIGHT та LEFT JOIN.

Оператор INSERT додає до таблиці рядок. Має такий синтаксис:

```
INSERT INTO назва_таблиці VALUES (список_значень);
```

Значення мають йти в такому ж порядку, як і стовпці таблиці. При необхідності можна задати конкретні стовпці, та конкретні значення:

```
INSERT INTO students(name) VALUES ('Іван');
```

Всі інші поля отримують значення за замовчуванням.

Команда UPDATE змінює значення полів в уже існуючих записах:

```
UPDATE таблиця SET стовпець1=значення1, ... , ... WHERE умова;
```

Оператор DELETE: DELETE FROM назва_таблиці WHERE умова;

Мова SQL не передбачає наявності команди скасування, тому такі операції рекомендується виконувати лише при абсолютній необхідності [12].

1.7 Нормалізація реляційних баз даних

Нормалізація – це процес ефективної організації даних у базі даних. Є дві цілі процесу нормалізації: усунення зайвих даних (наприклад, зберігання одних і тих же даних у більш ніж одній таблиці) та забезпечення залежності даних (має сенс лише зберігання пов'язаних даних у таблиці).

Спільнота баз даних розробила низку вказівок щодо забезпечення нормалізації баз даних. Вони називаються нормальними формами і нумеруються від однієї (найнижча форма нормалізації, називається першою нормальною формою, 1NF) через п'ять (п'ята нормальна форма, 5NF). У практичних програмах ви часто бачите 1-3NF, а також епізодичні 4NF.

Іноді виникає необхідність відхилятися від нормальних форм, щоб задовольнити практичні вимоги бізнесу. Однак коли виникають зміни, надзвичайно важливо оцінити всі можливі наслідки, які вони могли б мати у системі, та врахувати можливі невідповідності.

Перша нормальна форма (1NF) встановлює дуже основні правила для організованої бази даних:

- відсутність дублікатів стовпців в тій ж таблиці.
- окремі таблиці для кожної групи пов'язаних даних та визначте кожен рядок унікальним стовпцем або набором стовпців (первинний ключ).

Друга нормальна форма (2NF) стосується концепції видалення дублюючих даних:

- відповідність всім вимогам першої нормальної форми.
- видалення підмножини даних, які застосовуються до декількох рядків таблиці, і помістити їх в окремі таблиці.
- створення зв'язків між цими новими таблицями та їх попередниками за допомогою використання зовнішніх ключів.

Третя нормальна форма (3NF):

- відповідність всім вимогам другої нормальної форми.
- видалення стовпців, які не залежать від первинного ключа.

Норми щодо нормалізації є сукупними. Щоб база даних знаходилася в 2NF, вона повинна спочатку виконати всі критерії бази 1NF.

Хоча нормалізація бази даних часто є хорошою ідеєю, це не є вимогою. Є випадки, коли порушення правил нормалізації є хорошою практикою [13].

1.8 Системи управління базами даних

СУБД – це програмні засоби, які виступають посередником між БД та її користувачами. Завдяки сукупності мовних та програмних засобів, СУБД сприяють створенню, веденню та спільного використання БД різними користувачами.

Більшість сучасних СУБД складаються з ядра, процесору мови БД, підсистеми підтримки часу виконання та сервісних програм, які надають додаткові можливості для обслуговування інформаційних систем.

До основних СУБД відносять:

- об'єктно-реляційна система управління базами даних Oracle Database;
- вільна система управління базами даних PostgreSQL;
- система керування базами даних Microsoft SQL Server;
- вільна система управління базами даних MySQL.

Такі системи управління відрізняються централізованою обробкою запитів, забезпечують надійність, доступність та безпеку БД [11].

У даному проекті використовується СУБД PostgreSQL.

PostgreSQL – це потужна об'єктно-реляційна система баз даних з відкритим кодом, яка використовує та розширює мову SQL у поєднанні з багатьма функціями, які безпечно зберігають та масштабують найскладніші робочі навантаження даних.

PostgreSQL працює на всіх основних операційних системах, сумісний з ACID з 2001 року та має потужні доповнення, такі як популярний розширювач геопросторових баз даних PostGIS.

На додаток до того, що PostgreSQL є безкоштовним та відкритим кодом, він дуже розширюваний. Наприклад, ви можете визначати власні типи даних, створювати власні функції, навіть писати код з різних мов програмування, не перекомпілюючи базу даних!

PostgreSQL намагається відповідати стандарту SQL, коли така відповідність не суперечить традиційним особливостям або може призвести до неправильних архітектурних рішень. Багато функцій, які вимагає стандарт SQL, підтримуються, хоча іноді мають різний синтаксис або функції [14].

1.9 Вибір технологій, сервісів та бібліотек

Серверна частина веб-додатку необхідна для його роботи та для роботи перевезень в майбутньому, і складається з трьох компонентів:

- програмна логіка (далі бек-енд);
- база даних;
- веб-сервер.

Для реалізації програмної логіки було використано наступні технології:

- мова програмування Python [15];
- фреймворк розробки веб-додатків Django [16];
- бібліотеки на мові Python, призначені для роботи з Django. Повний список використаних бібліотек зазначено в коді бек-енду.

Для організації проекту Django було обрано бібліотеку Poetry – модуль, що реалізує зручну роботу з віртуальним середовищем та управлінням версіями Python [17].

Для підтримки відповідності програмного коду стандартам PEP [18] використано бібліотеку Flake8 [19].

База даних веб-додатку побудована та працює за допомогою технології PostgreSQL [14]. У якості веб-серверу було використано Nginx [20]. Даний вибір обґрунтовано легким встановленням та значною гнучкістю налаштування у порівнянні із іншими варіантами. Для організації WSGI-рішення для Django було обрано Gunicorn з причини підтримки Nginx [21]. Docker [22] відповідає за зручне розгортання веб-додатку на сервері, а надбудова Docker-compose [23] – за автоматизацію даного процесу. Інші використані технології, сервіси та мови збереження даних: Git [24], Bash [25], Yaml [26], Toml [27], Json [28], DigitalOcean [29], GitHub [30].

1.10 Обґрунтування обраних технологій, сервісів та бібліотек

Вибір Django було зроблено спираючись на наступні переваги:

– даний фреймворк має найширший набір інструментів, надбудов та компонентів на момент розробки додатку. Він включає як і базовий функціонал фреймворку для створення веб-додатків, тобто роботу з html запитами та веб-сервером, так і вбудовану підтримку найпоширеніших СУБД, роботу з файлами, проміжними шарами, систему авторизації, стандартні засоби безпеки, такі як CSRF, а також конструктори візуальних шаблонів та панелі адміністрування базою даних. Треба взяти до уваги, що в рамках даного проекту замість вбудованих технологій Django, таких як CSRF, авторизація та візуальні шаблони, буде використано окремо розроблені альтернативи;

– Django є одним з найпопулярніших фреймворків для Python та активно розвивається, присутня підтримка найновіших технологій;

– присутня детальна документація кожної частини фреймворку та активна підтримка фреймворку розробниками, що значно полегшує процес розробки та ознайомлення з його інструментарієм.

Django як фреймворк задає структуру проекту. Вона допомагає розробникам розуміти, де і як додавати нову функціональність. Альтернативні легкі фреймворки типу Flask, хоч і дозволяють бути вільніше відносно Django в плані екосистеми і конфігурації, можуть вимагати додатковий час на пошук та створення додаткових бібліотек і функціональних можливостей в довгостроковій перспективі.

За допомогою сторонніх додатків адміністративну панель Django можна вдосконалити і адаптувати під потреби свого проекту. Крім того, даний фреймворк дозволяє налаштовувати її інтерфейс [31].

Вигляд головної сторінки адміністративної панелі адміністрування зображено на рисунку 1.2.

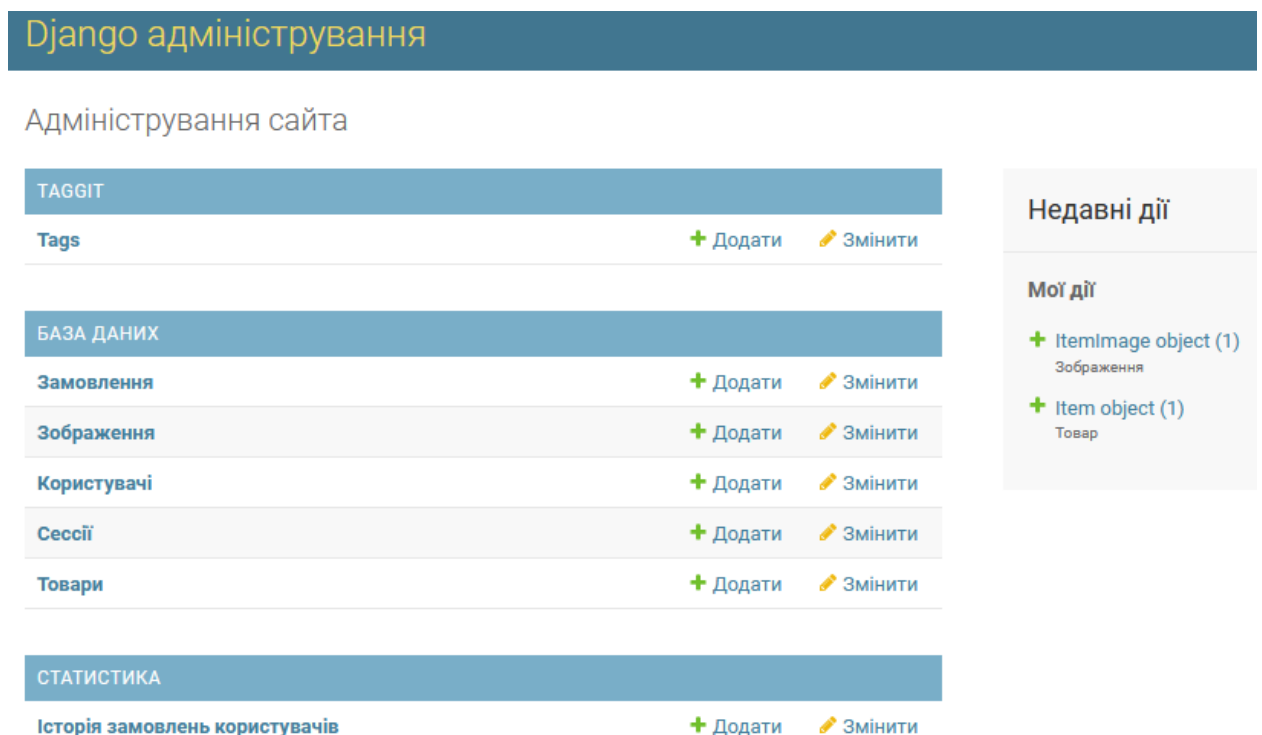


Рисунок 1.2 – Приклад адміністративної панелі

Для використання фреймворку Django необхідно також обрати СУБД.

У розроблюваному додатку одним з головних факторів є цілісність даних, тому що операцій на читання буде значно більше ніж на запис, в такому випадку система керування базами даних PostgreSQL буде, мабуть, найкращим вибором. Вона містить повну підтримку надійних транзакцій, виконуються умови атомарності, послідовності, ізоляваності та міцності (Atomicity, Consistency, Isolation, Durability) [32].

Крім того, завдяки своїм потужним технологіям, PostgreSQL є дуже продуктивною системою: паралельність у ній досягнута не за рахунок блокування операцій читання (як це можна побачити у MySQL), а завдяки реалізації управління різноманітним паралелізмом MVCC, де кожному користувачеві надається «знімок» бази даних, що володіє тими же властивостями. Цей спосіб управління дозволяє пишущим транзакціям не блокувати читаючі, а читаючим – не блокувати пишучі [33].

1.11 Програмна структура веб-додатків Django

Кожний Django веб-додаток має схожу структуру проекту: він складається з директорії самого проекту, директорій окремих модулів-додатків та файлу `manage.py`. Даний файл використовується для запуску проекту [34]. У директорії проекту визначаються налаштування всієї серверної частини, а також повний список кінцевих точок (`url-адрес`) бек-енду.

Основна частина логіки проекту знаходиться у додатках – модульних частинах проекту Django [35]. Так, логіка додатку зазвичай поділяється на три види – логіку представлення, модель та бізнес-логіку.

Документація Django визначає модель (`model`) як «джерело інформації про дані, в яких містяться ключові поля і поведінка даних». Зазвичай одна модель вказує на одну таблицю в базі даних. Django підтримує бази даних PostgreSQL, MongoDB, MySQL, SQLite і Oracle.

Моделі містять інформацію про об'єкти. Ці об'єкти представлені атрибутами або полями. Оскільки модель являє собою простий клас, вона

нічого не знає про інші складові Django. Модель відповідає за бізнес-логіку, методи, властивості і елементи, пов'язані з маніпуляцією об'єктом. Моделі дозволяють створювати, читати, оновлювати та видаляти об'єкти у БД [36].

Представлення (view) вирішує три завдання: приймає HTTP-запити, реалізує поведінку додатку і відправляє різноманітні об'єкти у відповідь [37].

Приклад моделі зображено на рисунку 1.3, приклад представлення – на рисунку 1.4. Зазвичай, відправлений об'єкт є стандартним HTTP-шаблоном Django, що відображає результат представлення [38].

Проміжні шари – функції, що перехоплюють HTTP-запит до того, як він дійде до представлення, та модифікують його, або перевіряють на безпечність [39]. На рисунку 1.5. зображено приклад такого шару.

```

48 class User(AbstractBaseUser, PermissionsMixin):
49     """Модель користувача. Дана модель представляє собою базового
50     користувача - з ім'ям, прізвищем, ел. поштою та паролем.
51     Дана модель також базується на PermissionsMixin, тому має
52     інтегровану систему прав користувача:
53     - `is_superuser`
54     - `groups`
55     - `user_permissions`
56     """
57
58     class Meta:
59         verbose_name = ("користувач")
60         verbose_name_plural = ("користувачі")
61
62     USERNAME_FIELD = 'email'
63
64     REQUIRED_FIELDS = ['first_name', 'last_name']
65
66     email = models.EmailField(
67         verbose_name="ел. пошта"), unique=True,
68         error_messages={'unique': ("Користувач з такою поштою вже існує"), },
69     )
70     first_name = models.CharField(
71         max_length=30, verbose_name="ім'я"),
72     )
73     last_name = models.CharField(
74         max_length=30, verbose_name="прізвище"),
75     )

```

Рисунок 1.3 – Приклад моделі

```

furniture-marketplace > backend > core > views.py > ...
114 @protected_view
115 def add_item_view(request):
116     """Дане представлення додає новий товар до бази даних."""
117     if errors := Forms.new_item(request.args):
118         return fail_validation(errors)
119     if ex := Users.add_item(request.user, request.args):
120         return fail_validation(Forms.fail(Forms.IMAGE_MSG, 'image'), ex)
121     return ok()

```

Рисунок 1.4 – Приклад представлення

```
furniture-marketplace > backend > core > utils > middleware.py > args_middleware > middleware
21 def args_middleware(get_response):
22     """Даний проміжний слой декодує дані тіла запиту та записує
23     результат як частину запиту - у аргумент args типу dict.
24
25     Дані можна отримати або напряму (request.args.get(key) або
26     request.args[key]), або через метод get() (request.get(key)),
27     де key - це назва ключу, що шукається."""
28
29     def middleware(request):
30
31         def get(*keys):
32             """Метод, що приймає назву ключу на вхід та шукає даний
33             ключ у списку переданих у запиті аргументів. Якщо ключа
34             немає, повертаю None. Якщо виникла помилка із запитом,
35             повертає пусту строку ('')."""
36             result = []
37             for key in keys:
38                 try:
39                     value = request.args.get(key)
40                 except (TypeError, AttributeError):
41                     value = ''
42                 result.append(value)
43             return result[0] if len(result) == 1 else tuple(result)
```

Рисунок 1.5 – Приклад проміжного шару

1.12 Презентація бізнес-проекту

Для реалізації значних змін або розширень в роботі підприємства або сервісу необхідно чітко сформулювати, описати та обґрунтувати всі пов'язані з такими змінами дії, використані ресурси, можливі наслідки, варіанти та прогнози.

Сукупність даних показників представляє собою бізнес-проект – ініціативу, яка прагне вирішити ділову проблему, яка не була повністю вирішена існуючою пропозицією. Проект залучає людей, процеси, бюджети та мету [40].

Хоча у даному випадку метою проекту є розширення функціоналу онлайн-сервісу через надання інструментів та підтримки продавцям та покупцям щодо перевезень товару, він (проект) спрямований в першу чергу на покращення робочих процесів та якості вже існуючих частин сервісу.

Метою проекту може бути також покращення поточного становища на ринку, і в цьому сенсі бізнес-проект прагне розвинути нові можливості в компанії [40].

Належне створення, а в подальшому функціонування та розвиток кожного суб'єкта підприємницької діяльності, потребує детального

планування, дієвим інструментом якого виступає процес бізнес-планування. Особливо при започаткуванні нового напрямку, виду економічної діяльності чи створенні стратегічного підрозділу суб'єкта бізнесу важливе значення відводиться розробці бізнес-плану як детального попереднього аналізу та прогнозу перспектив підприємницького проекту [41].

З цього приводу, для визначення та зручної презентації даних показників необхідно провести поверхнєве бізнес-планування. Це дозволить:

- промодельовати систему управління бізнес-проектом. Досвід та практика засвідчують, що успіх підприємницької діяльності визначається трьома факторами: правильним розумінням реальної ситуації в даний момент часу; чіткою постановкою цілей, яких хоче досягнути підприємство; якісним плануванням процесів переходу з одного стану в інший. Тому бізнес-план з його конкретними комерційними цілями та програмою дій для їх досягнення виступає в ролі засобу, здатного керувати бізнесом;

- передбачити ускладнення / ризики функціонування підприємства (його окремого стратегічного підрозділу чи реалізації окремого підприємницького проекту);

- залучити позичковий капітал, необхідний для створення підприємства чи реалізації окремого підприємницького проекту. Бізнес-план розкриває не лише виробничу, ринкову і організаційну, але і фінансову інформацію про підприємство. Ця інформація дозволяє робити висновки про те, якою є потреба підприємства у стартовому капіталі, як будуть витрачатися кошти та погашатися кредити, який рівень ризику господарювання і т. ін. У цьому контексті бізнес-план є основою конструктивних відносин з потенційними інвесторами, банком і т. ін.;

- забезпечити процес „комунікації” між підприємством та інституціями зовнішнього середовища його функціонування (постачальниками, контрагентами, покупцями, потенційними працівниками);

- розвинути управлінські навички керівництва та іншого управлінського персоналу підприємства [42].

1.13 Теорія ігор у конфліктній ситуації

Конфлікт – це ситуація з кількома учасниками, цілі яких не збігаються і дії яких не є абсолютно незалежними. До конфліктних ситуацій належать, наприклад, взаємини між банком і клієнтом, покупцем та продавцем, начальником та підлеглим і т. п.

Теорія ігор – це розділ математики, в якому досліджуються математичні моделі прийняття рішень в умовах конфлікту, тобто в умовах зіткнення сторін, кожна з яких прагне впливати на розвиток конфлікту у власних інтересах.

Математична модель конфліктної ситуації називається грою, сторони, що беруть участь у конфлікті, – гравцями.

Вибір і застосування однієї з передбачених правилами дій називається ходом гравця. Ходи можуть бути особистими і випадковими.

Особистий хід – це свідомий вибір гравцем однієї з можливих дій (наприклад, хід у шаховій грі).

Випадковий хід – це випадково вибрана дія (наприклад, вибір карти з перетасованої колоди). Далі розглядатимемо тільки особисті ходи гравців.

Природно, що гравець приймає рішення по ходу гри. Проте теоретично можна припустити, що всі ці рішення прийняті гравцем заздалегідь. Сукупність цих рішень становить його стратегію. Стратегією гравця називається деякий план або сукупність правил, за якими він обирає рішення під час кожного особистого ходу залежно від ситуації, що склалася у процесі гри [43].

Для теорії ігор фундаментальними є три поняття:

- конфлікт і його сторони;
- прийняття рішення в конфлікті;
- оптимальність прийнятого рішення.

Ці поняття утворюють логічну основу теорії та входять у її визначення. Формалізація понять відповідає змістовним уявленням про відповідні об'єкти.

Звісно, конфліктом можна назвати будь-яке явище, для якого в свою чергу можна визначити його учасників, їхні дії, результати явищ, до яких призводять дії. Також часто говорять про сторони конфлікту, які тією чи іншою мірою зацікавлені в певних результатах і про сутність цієї зацікавленості [44].

Очевидно, якщо перший гравець має всього m стратегій, а другий – n , то таку гру можна описати, задавши виграші гравців двома матрицями. Така пара матриць також називається нормальною формою гри.

Оскільки в описі гри двох осіб фігурують дві матриці, такі ігри часто називають біматричними.

Біматрична гра двох гравців така, що для будь-яких стратегій i, j виконується: $a_{ij} + b_{ij} = \text{const}$, може бути задана за допомогою однієї матриці і тому такі ігри називають матричними іграми з постійною сумою або просто матричними.

Матричні ігри з нульовою сумою називають антагоністичними.

Нормальна форма матричної гри з матрицею A є самою платіжною матрицею A .

Матрична гра двох гравців з нульовою сумою може розглядатися як подана нижче абстрактна гра двох гравців. Перший гравець має m стратегій $i = 1, 2, \dots, m$, другий – n стратегій $j = 1, 2, \dots, n$. Кожній парі стратегій (i, j) поставлено у відповідність число a_{ij} , що виражає виграш гравця 1 за рахунок гравця 2, якщо перший гравець вибере свою i -ту стратегію, а гравець 2 – свою j -ту стратегію.

Кожен із гравців робить один хід: гравець 1 вибирає свою i -ту стратегію ($i=1, m$), гравець 2 – свою j -ту стратегію ($j=1, n$), після чого гравець 1 отримує виграш a_{ij} за рахунок гравця 2 (якщо $a_{ij} < 0$, то це означає, що перший гравець платить другому суму $|a_{ij}|$). На цьому гра закінчується.

Головним у дослідженні ігор є поняття оптимальних стратегій гравців.

У це поняття інтуїтивно вкладається такий сенс: стратегія гравця є оптимальною, якщо: а) застосування цієї стратегії забезпечує йому найбільший гарантований виграш при всіляких стратегіях іншого гравця і б) відхилення від цієї стратегії не вигідне.

Виходячи з цих міркувань, гравець 1 досліджує матрицю виграшів A таким чином: для кожного значення i ($i = 1, m$) визначається мінімальне значення виграшу залежно від вживаних стратегій гравця 2: $\min_j a_{ij}$ ($i = 1, m$), тобто визначається мінімальний виграш для гравця 1 за умови, що він прийме свою i -ту чисту стратегію, потім з цих мінімальних виграшів відшукується така стратегія $i = i_0$, за якої цей мінімальний виграш буде максимальним, тобто знаходиться $\max_i \min_j a_{ij} = a_{i_0 j_0} = \alpha$ [45].

При виборі гравцем 1 i -ї стратегії, а гравцем 2 – j -ї стратегії їх виграші перебувають в матрицях виплат на перетині i -х рядків і j -х стовпців: в матриці A це елемент a_{ij} , а в матриці B – елемент b_{ij} .

Таким чином, в разі, коли інтереси гравців різні (але не обов'язково протилежні), виходять дві платіжні матриці: одна – матриця виплат гравцеві A , інша – матриця виплат гравцеві B . Для біматричних ігор також розроблена теорія оптимальної поведінки гравців.

Безперервною вважається гра, в якій функція виграшів кожного гравця є безперервною в залежності від стратегій. У теорії математики доведено, що ігри цього класу мають рішення, проте поки не розроблено практично прийнятних методів їх знаходження [46].

1.14 Транспортні задачі

Транспортна задача – це задача лінійного програмування, яка ставиться як задача знаходження оптимального плану перевезення вантажів із M пунктів відправлення в N пунктів призначення, коли вартості перевезення із пункту відправлення i в пункт призначення j відомі.

Існують два види транспортної задачі: транспортна задача по критерію вартості – треба знайти план перевезень з мінімальною вартістю. Транспортна задача по критерію часу – в якій більш важливим є час перевезення вантажів. Транспортна задача по критерію вартості є випадком задачі лінійного програмування і розв’язується симплекс методом [47].

Існує ряд методів розв’язання транспортних задач:

Метод потенціалів. Задача ТЗ може бути розв’язана за допомогою симплекс-методу, але її структура дозволяє використати більш ефективніший метод потенціалів. Також застосовується до ТЗ в закритій формі. Тому спочатку слід визначити тип транспортної задачі (відкрита чи замкнена) і збалансувати її. Алгоритм методу складається з наступних етапів.

1. Побудова початкового опорного плану транспортної задачі.
2. Перевірка плану транспортної задачі на оптимальність.
3. Якщо умова оптимальності виконується, то отримуємо оптимальний розв’язок транспортної задачі. Якщо ж умова не виконується, то визначається наступний опорний план (перехід до кроку 2).

Метод мінімальної вартості. Даний метод полягає в тому, що на кожному кроці заповнюють клітинку таблиці, яка містить величину найменшої вартості перевезення одиниці продукції, за алгоритмом, що наведений вище, доти, поки вся продукція не буде розподілена між споживачами.

Метод подвійної переваги. Перед початком заповнення таблиці необхідно позначити клітинки, які містять найменшу величину вартості с_{ij} у рядках і стовпчиках. Таблицю опорного плану починають заповнювати з клітинок, позначених двічі (як мінімальні і в рядку, і в стовпчику). Далі заповнюють клітинки, позначені один раз (як мінімальні або в рядку, або в стовпчику), а вже потім – за методом мінімальної вартості.

Метод апроксимації Фогеля. За його допомогою на кроках визначають різниці між двома найменшими вартостями в кожному рядку і стовпчику транспортної таблиці. Їх записують у спеціально відведених місцях (поряд з

відповідними рядками або стовпчиками). Серед усіх різниць вибирають найбільшу і у відповідному рядку чи стовпчику заповнюють клітинку з найменшою величиною вартості перевезення. Коли залишається незаповненим лише один рядок або стовпчик, таблицю продовжують заповнювати за методом мінімальної вартості.

Після побудови початкового опорного плану за допомогою одного із розглянутих методів у таблиці має бути заповнено $(m + n - 1)$ клітинок (далі будемо називати їх базисними), де m – кількість постачальників; n – кількість споживачів у задачі, у тому числі фіктивних. Такий план називають невинородженим.

Якщо кількість заповнених клітинок перевищує $(n + m - 1)$, то початковий план побудовано неправильно і він не є опорним. Ознакою опорності плану транспортної задачі є його ациклічність. Циклом у транспортній задачі називають замкнену ламану лінію, вершини якої розміщуються в заповнених клітинках таблиці, а сторони проходять уздовж рядків і стовпчиків таблиці [48].

1.15 Нотація UML

Універсальна мова моделювання (Unified Modelling Language або UML) – це мова позначень або побудови діаграм, призначена для визначення, візуалізації і документування моделей зорієнтованих на об'єкти систем програмного забезпечення. UML не є методом розробки, іншими словами, у конструкціях цієї мови не повідомляється про те, що робити першим, а що останнім, і не надається інструкцій щодо побудови вашої системи, але ця мова допомагає вам наочно переглядати компонування системи і полегшує співпрацю з іншими її розробниками. Розробкою UML керує Object Management Group (OMG). Ця мова є загальноприйнятим стандартом графічного опису програмного забезпечення.

UML розроблено для розробки структури зорієнтованого на об'єкти програмного забезпечення, ця мова має дуже обмежену користь для програмування на основі інших парадигм.

Конструкції UML створюються з багатьох модельних елементів, які позначають різні частини системи програмного забезпечення. Елементи UML використовуються для побудови діаграм, які відповідають певній частині системи або точці зору на систему. Існують такі типи діаграм:

- *Діаграма випадків використання* показує дієвих осіб (людей або інших користувачів системи), випадки використання (сценарії використання системи) та їх взаємодію;

- *Діаграми класів*, на яких буде показано класи та зв'язки між ними;

- *Діаграми послідовності*, на яких показано об'єкти і послідовність методів, якими ці об'єкти викликають інші об'єкти;

- *Діаграми співпраці*, на яких буде показано об'єкти та їх взаємозв'язок з наголосом на об'єкти, які беруть участь у обміні повідомленнями;

- *Діаграми стану*, на яких буде показано стани, зміну станів і події у об'єкті або частині системи;

- *Діаграми діяльності*, на яких буде показано дії та зміни однієї дії іншою, які є наслідком подій, що сталися у певній частині системи;

- *Діаграми компонентів*, на яких буде показано програмні компоненти високого рівня (на зразок KParts або Java Beans);

- *Діаграми впровадження*, на яких буде показано екземпляри компонентів та їх взаємодію;

- *Діаграми взаємозв'язку сутностей*, на яких буде показано дані, взаємозв'язки і умови обмеження зв'язків між даними [49].

Одне із завдань UML – служити засобом комунікації всередині команди та при спілкуванні з замовником. Розглянемо можливі варіанти використання діаграм:

- *Проектування*. UML-діаграми стануть у пригоді при моделюванні архітектури великих проектів, в якій можна зібрати як великі, так і дрібніші

деталі і намалювати каркас (схему) програми. По ньому пізніше буде будуватись код;

– Реверс-інжиніринг – створення UML-моделі з існуючого коду додатку, зворотна побудова. Може застосовуватися, наприклад, на проектах підтримки, де є написаний код, але документація неповна або відсутня.

– З моделей можна витягувати текстову інформацію і генерувати відносно читабельні тексти – документувати. Текст і графіка будуть доповнювати один одного [50].

Як будь-яка інша мова, UML має власні правила оформлення моделей і синтаксис. За допомогою графічної нотації UML можна візуалізувати систему, об'єднати всі компоненти в єдину структуру, уточнювати і покращувати модель у процесі роботи. На загальному рівні графічна нотація UML містить 4 основні типи елементів:

- фігури;
- лінії;
- значки;
- написи.

UML-нотація де-факто є галузевим стандартом у сфері розробки програмного забезпечення, IT-інфраструктури і бізнес-систем [50].

2 СПЕЦІАЛЬНИЙ РОЗДІЛ

2.1 Проектування реляційної моделі бази даних

У даному розділі розглядається процес проектування бази даних, з якою буде взаємодіяти серверна частина веб-додатку.

2.1.1 ER-проектування

Діаграма сутність-зв'язок є результатом систематичного процесу, який описує та визначає деяку предметну область. Вона не визначає сам процес, а лише візуалізує його. Дані представлені у вигляді сутностей, які пов'язані між собою певними зв'язками, що виражають залежності і вимоги між ними.

Первинний вигляд бази даних зображений на рисунку 2.1.



Рисунок 2.1 – Первинний вигляд ER

У діаграмі використовуються три основні сутності: користувач, магазин та виробник.

Дана версія моделі є неоптимізованою. Це означає, що вона недосконало описує набір функцій системи та неповністю відображає взаємодію між її компонентами. Для усунення таких недоліків необхідно провести ряд оптимізаційних операцій, результатом яких буде нова ER-схема.

Таким чином, модель Відгук тепер пов'язана блоком Характеризує із моделями Товар та Магазин.

Моделі Користувач, Виробник та Магазин мають майже однаковий набір ф-ій та зв'язків, що призводить до необхідності поєднання їх у одну модель для полегшення функціонування всієї системи.

Отже, створимо модель Транзакція та поєднаємо з нею моделі Користувач і Товар. Вигляд моделі після оптимізації зображений на рисунку 2.2.

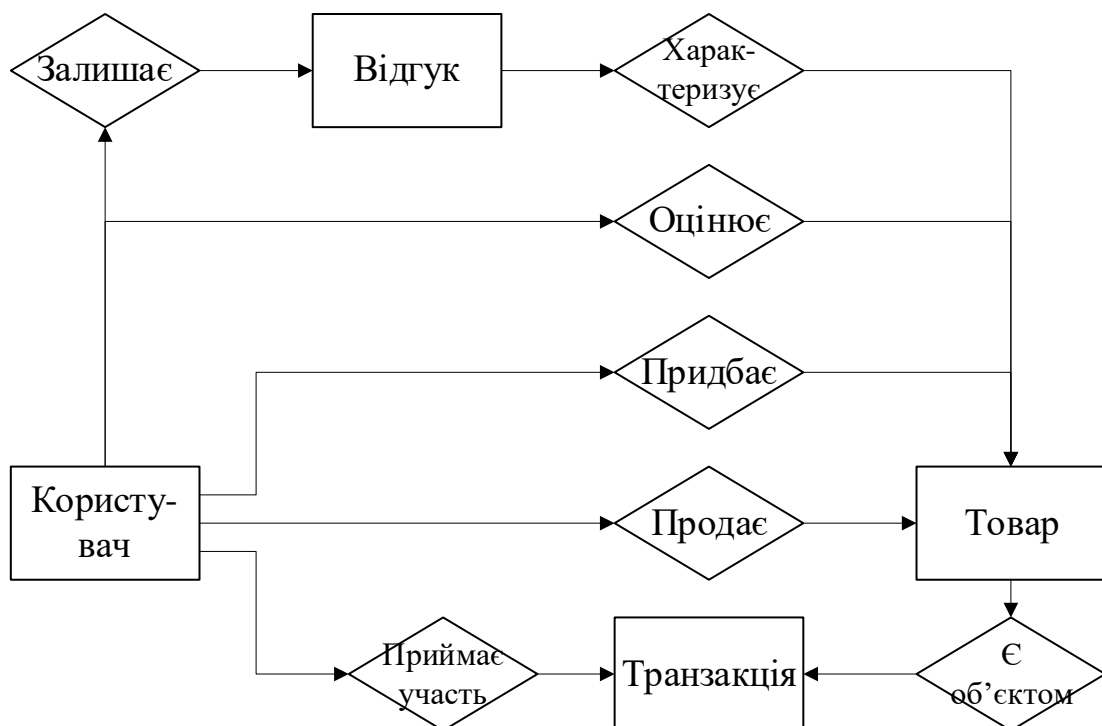


Рисунок 2.2 – Вигляд ER-моделі після оптимізації

2.1.2 Опис результуючої моделі

Користувач – сутність, яка працює з одним або групою оголошень на платформі. Він може придбати або здійснювати продаж товару, а також залишати відгук, який, у свою чергу, характеризує товар.

Товар і транзакція є сутностями, що хоч і не мають великого набору дій, але займають важливе місце у схемі, тому що більшість операцій проводяться саме над ними. Слід розуміти, що така діаграма може не

відповідати кінцевій структурі бази даних, тому що не відображає повну інформацію про модель транзакції, користувача тощо, але допомагає інтуїтивно зрозуміти зв'язки у логіці.

2.1.3 Проектування реляційної структури

На рисунку 2.3 приведена схема реляційної структури бази даних:

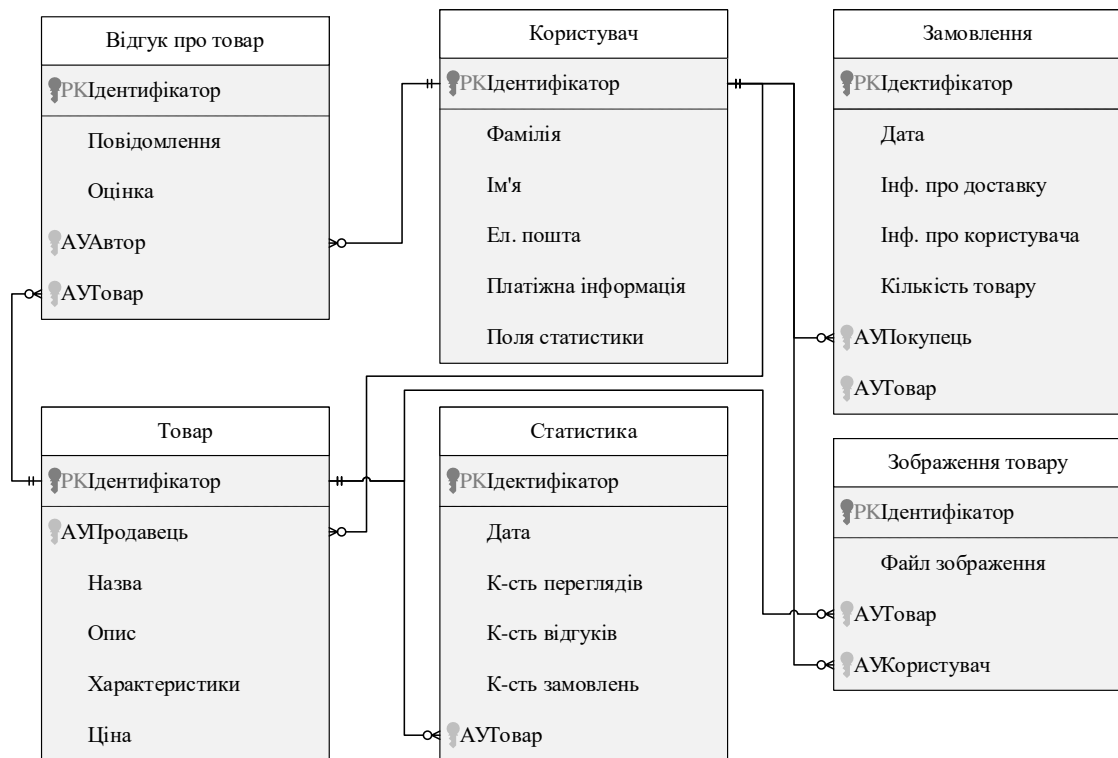


Рисунок 2.3 – Реляційна структура бази даних

В даному випадку було обрано залишити атрибут «Платіжна інформація» цілісним з причини невизначеності інформації, необхідної для роботи із різноманітними платіжними системами. Поля окремих характеристик товару та статистики користувача були також об'єднані у одне.

Дана реляційна система не включає атрибутів, що відповідають за безпеку користувачів – такі атрибути як хеш паролю та інші. Це зроблено для того, щоб систему захисту можна було обрати окремо та інтегрувати у додаток.

Список параметрів всіх таблиць і полів зображений у таблицях 2.1-2.6.

Таблиця 2.1

Користувач (User)

Поле	Атрибут	Тип
Ідентифікатор	id	Integer, PK
Ім'я	first_name	String (30)
Фамілія	last_name	String (30)
Ел. пошта	email	Special String
Є адміністратором	is_staff	Boolean
Є активованим	is_active	Boolean
Дата реєстрації	date_joined	DateTime

Таблиця 2.2

Зображення товару (ItemImage)

Поле	Атрибут	Тип
Ідентифікатор	id	Integer, PK
Файл зображення	image	Special String
Користувач	user__id	Integer, FK (User.id)
Товар	item__id	Integer, FK (Item.id)

Таблиця 2.3

Замовлення (Order)

Поле	Атрибут	Тип
Ідентифікатор	id	Integer, PK
Дата	date_created	DateTime
Ім'я	first_name	String (30)
Фамілія	last_name	String (30)
Ел. пошта	email	Special String
Телефон	phone	String (16)

Кількість	amount	Integer (>0)
Інф. про доставку	delivery	Text
Товар	item__id	Integer, FK (Item.id)

Таблиця 2.4

Товар (Item)

Поле	Атрибут	Тип
Ідентифікатор	id	Integer, PK
Назва	name	String (32)
Рейтинг	rate	Decimal (2)
Теги	tags	Special String, Null
Довжина	length	Decimal (2)
Ширина	width	Decimal (2)
Колір	color	String (50)
Матеріал	material	String (50)
Інші хар-ки	params	Text, Null
Ціна	price	Decimal (2)
Продавець	seller__id	Integer, FK (User.id)
Дата створення	date_created	DateTime

Таблиця 2.5

Статистика товару (ItemStatistic)

Поле	Атрибут	Тип
Ідентифікатор	id	Integer, PK
К-сть переглядів	clicks	Integer
К-сть відгуків	purchases	Integer
К-сть замовлень	reviews	Integer
Дата	date	Date
Товар	item__id	Integer, FK (Item.id)

Поле товару «Характеристики» є цілісним полем і представляє собою мультистрокове текстове поле. Зважаючи на те, що різні товари мають мати різний набір характеристик, для універсальності даного поля було прийняте рішення зробити його текстовим.

Таблиця 2.6

Відгук (ItemReview)

Поле	Атрибут	Тип
Ідентифікатор	id	Integer, PK
Повідомлення	body	Text, Null
Оцінка	rate	Decimal (2)
Автор	name	String (50)
Товар	item__id	Integer, FK (Item)
Дата	date_created	DateTime

В приведених вище таблицях використовуються такі типи даних:

- Integer – ціле число;
- Float – дійсне число;
- Decimal – дійсне число з фіксованою точністю (вказана в скобках);
- String – строка з фіксованою макс. довжиною (вказана в скобках);
- Text – мультистрокове поле з необмеженою максимальною довжиною;
- Date/DateTime – спеціальне поле з датою.

Інші позначення:

- PK – первинний ключ;
- FK – зовнішній ключ (таблиця-ціль вказана в скобках);
- Null – поле може не мати значення (може бути пустим).

Таблиці знаходяться в першій нормальній формі, тому що усі атрибути мають атомарний стан. В другій нормальній формі – тому що усі поля в кожній таблиці залежать від єдиного первинного ключа та повністю описують її, а також задовольняють правилам першої нормальної форми.

Таблиці знаходяться в третій нормальній формі, тому що немає залежності одних неключових атрибутів від інших, усі атрибути залежать тільки від первинного ключа, а також повністю задовольняють правилам першої та другої нормальних форм.

2.1.4 Моделювання SQL-запитів

Для спрощення адміністрування на сервері PostgreSQL до базового комплекту після установки входить такий інструмент як pgAdmin. Він являє собою графічний клієнт для роботи з сервером, через який можна створювати, видаляти, змінювати бази даних і управляти ними.

Запити для створення базової схеми БД:

```
DROP TABLE IF EXISTS "api_user" CASCADE;
DROP TABLE IF EXISTS "api_item" CASCADE;
DROP TABLE IF EXISTS "api_order" CASCADE;
DROP TABLE IF EXISTS "api_itemreview" CASCADE;
DROP TABLE IF EXISTS "api_itemimage" CASCADE;
DROP TABLE IF EXISTS "stats_itemstatistic" CASCADE;
CREATE TABLE "api_user" (
    "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
    "password" varchar(128) NOT NULL,
    "last_login" datetime NULL,
    "is_superuser" bool NOT NULL,
    "email" varchar(254) NOT NULL UNIQUE,
    "first_name" varchar(30) NOT NULL,
    "last_name" varchar(30) NOT NULL,
    "is_staff" bool NOT NULL,
    "is_active" bool NOT NULL,
    "date_joined" datetime NOT NULL );
CREATE TABLE "api_item" (
    "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
```

```

"name" varchar(50) NOT NULL,
"rate" decimal NOT NULL,
"date_created" datetime NOT NULL,
"price" decimal NOT NULL,
"length" decimal NOT NULL,
"width" decimal NOT NULL,
"color" varchar(50) NOT NULL,
"material" varchar(50) NOT NULL,
"custom" text NOT NULL,
"seller_id" bigint NOT NULL REFERENCES "api_user" ("id")
DEFERRABLE INITIALLY DEFERRED );

CREATE TABLE "api_order" (
  "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
  "delivery" text NOT NULL,
  "email" varchar(254) NOT NULL,
  "first_name" varchar(30) NOT NULL,
  "last_name" varchar(30) NOT NULL,
  "phone" varchar(16) NOT NULL,
  "amount" integer NOT NULL,
  "date_created" datetime NOT NULL,
  "item_id" bigint NOT NULL REFERENCES "api_item" ("id")
DEFERRABLE INITIALLY DEFERRED );

CREATE TABLE "api_itemreview" (
  "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
  "name" varchar(50) NOT NULL,
  "rate" decimal NOT NULL,
  "date_created" datetime NOT NULL,
  "body" text NOT NULL,
  "item_id" bigint NOT NULL REFERENCES "api_item" ("id")
DEFERRABLE INITIALLY DEFERRED );

```

```

CREATE TABLE "api_itemimage" (
    "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
    "image" varchar(100) NOT NULL,
    "date_created" datetime NOT NULL,
    "item_id" bigint NULL REFERENCES "api_item" ("id") DEFERRABLE
INITIALLY DEFERRED,
    "user_id" bigint NOT NULL REFERENCES "api_user" ("id")
DEFERRABLE INITIALLY DEFERRED );

CREATE TABLE "stats_itemstatistic" (
    "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
    "clicks" integer NOT NULL,
    "purchases" integer NOT NULL,
    "reviews" integer NOT NULL,
    "date" date NOT NULL,
    "item_id" bigint NOT NULL REFERENCES "api_item" ("id")
DEFERRABLE INITIALLY DEFERRED );

```

Додаткові запити які допоможуть маніпулювати даними:

Отримати всі відгуки товарів у діапазоні оцінки між 4 та 5:

```
SELECT * FROM "api_itemreview" WHERE rate BETWEEN 4 AND 5
```

Отримати всі унікальні товари:

```
SELECT DISTINCT (name) FROM "api_item"
```

Отримати всю статистику товарів після заданої дати:

```
SELECT * FROM "stats_itemstatistic" WHERE date>' ' ORDER BY date DESC
```

Висновок

Результуюча схема бази даних має наступні корисні властивості:

- сумісність із будь-якими видами меблів за рахунок універсальних та користувацького полів для збереження характеристик;
- наявність змодельованих наперед SQL-запитів для взаємодії з БД.

Схема передбачає та є сумісною із будь-якими змінами до її структури. Крім того, використовувана для взаємодії з БД технологія Django виконує зміни автоматично, що впроваджує швидкість та зручність подальших змін. Слід зауважити, що у схему не увійшли технічні таблиці, необхідні для збереження певних технічних даних, такі як, наприклад, токени сесій.

2.2 Проектування та розробка back-end

У даному розділі описується результат проектування та розробки серверної частини веб-додатку, а саме: проект Django, налаштування Docker та Nginx.

2.2.1 Структура серверної частини веб-додатку

У якості бази бек-енду використовується Django, мова програмування – Python. Регулювання версій всіх компонентів відбувається за допомогою Poetry. Окрім обов'язкового файлу взаємодії з Django під назвою `manage.py`, коренева директорія містить файл `pyproject.toml`. Даний файл зазвичай використовується Poetry для збереження даних про вимоги щодо встановлюваних залежностей та бібліотек, але у даному випадку в ньому також зберігатимуться налаштування Django, які в свою чергу зазвичай знаходяться у файлі `settings.py` в директорії проекту.

Сама директорія проекту називається `core` та містить оголошення WSGI-об'єкту, перелік допустимих адрес бек-енду (`urls.py`) та всі представлення (`views.py`).

Нарешті, коренева директорія містить директорії-додатки `api` та `stats`, а також директорію тестування `tests`.

Директорія Django-додатку `api` містить специфікацію бази даних, панелі адміністрування та бізнес-логіки, пов'язаної з основним функціоналом сайту – користувач, товар, замовлення та інші.

Крім того, у даному додатку реалізована робота з сесіями за допомогою JWT.

У додатку `stats` реалізована робота із моделями статистики сайту. В цій частині додатку також реалізоване прогнозування певних частин статистики, а саме – прогнозування статистики товарів, такої як перегляди, відгуки та замовлення, на тиждень вперед.

2.2.2 З'єднання з клієнтською частиною та розгортання на сервері

Для того, щоб проект зміг приймати запити від користувачів, йому необхідний веб-сервер. У даному проекті у якості веб-серверу було використано Nginx. Код з його конфігурацією завжди знаходиться у файлі під назвою `nginx.conf`.

Nginx перехоплює запити на сервер, оброблює їх та перенаправляє у правильні частини проекту.

Для того, щоб розгорнути та встановити веб-додаток на сервері, необхідна стратегія розгортання. У даному проекті вона реалізована за допомогою Docker, а так як проект складається з декількох частин (фронт-енд, бек-енд, веб-сервер, база даних), для зв'язку їх контейнерів використано `docker-compose`. Конфігурація зв'язку контейнерів знаходиться у файлах під назвою `docker-compose.ci.yml` та `docker-compose.prod.yml`. Внутрішні налаштування кожного контейнеру знаходяться у файлах під назвою `Dockerfile` у відповідних директоріях.

Сервіс `docker-compose` створює систему контейнерів, пов'язаних між собою, а Docker встановлює в кожному з них відповідну частину проекту та програмне забезпечення, необхідне для її роботи.

Висновок

Розділення проекту на окремі взаємодіючі між собою частини призвело до модульності додатку, що робить оновлення окремих частин набагато зручнішим, тому що при оновленні однієї частини немає необхідності вимикати або перезавантажувати всі інші частини. Це не тільки підвищує зручність використання сайту, але і підвищує швидкість заміни контейнерів на нові.

Треба зауважити, що зв'язки контейнерів мають чітку структуру, а тому і чіткий порядок завантаження – бек-енд не може працювати без бази даних, а веб-сервер не може працювати без фронт-енду та бек-енду.

2.3 Вибір алгоритмів аналізу та прогнозування даних

У даному розділі описується процес вибору методів формування рейтингу товарів та прогнозування їх статистики, а також послідує обґрунтування зробленого вибору.

2.3.1 Обчислення і аналіз показників популярності і рейтингу

Показник рейтингу товару відповідає за його популярність на сайті та за його якість – чим більше користувач задоволений товаром, тим більше оцінку він дасть цьому товару. Чим більше оцінок від користувачів – тим популярнішим можна вважати товар.

Таким чином, популярність товару можна розраховувати на основі статистики відгуків про товар – чим більше відгуків на товар за останній час та чим вище оцінка кожного відгуку, тим вище поточна популярність товару. Для зменшення часу обчислення та зменшення навантаження на сервер популярність товарів можна кешувати.

Обчислення рейтингу товару можна проводити простим методом – обчисленням середнього значення всіх відгуків кожного разу коли товар отримує новий відгук.

Такий метод хоч і є зручним та передбачає видалення та зміну оцінок відгуків, але є дуже ресурсомістким, особливо при великій кількості відгуків товару: для обчислення середнього значення буде необхідно брати список всіх відгуків товару, брати с кожного оцінку, додавати та ділити на загальну кількість відгуків.

З метою усунення даної проблеми було використане альтернативне рішення – додавати оцінку нового відгуку до оцінки товару, помножену на стару кількість відгуків. Далі отримана сума ділиться на нову кількість оцінок.

Треба зауважити, що за таким методом неможливо встановити, які саме відгуки існували до нового. Томи при виникненні необхідності будь-якої

подальшої маніпуляції відгуками потрібно буде перейти на використання першого способу.

Формула даного методу виглядає так:

$$R_n = \frac{(R+R_o \cdot N_{R_o})}{N_{R_o}+1}, \quad (2.1)$$

де:

- R_n – новий рейтинг товару;
- R_o – старий рейтинг товару;
- R – оцінка відгуку;
- N_{R_o} стара кількість відгуків товару.

Такий метод дозволить звертатись лише до рейтингу товару та загальної кількості відгуків.

2.3.2 Вибір алгоритму прогнозування статистики товару

Кожен товар на платформі має свою статистику переглядів, відгуків та замовлень. Ця статистика оновлюється кожний раз коли відбувається одна з вище перелічених дій та відображає поденне число кожної з них.

Таким чином, присутній набір числових даних, який можна спрогнозувати. У такому випадку прогнозування не тільки допоможе власнику товару побачити, що приблизно чекає на його товар у майбутньому, але і допоможе приблизно визначити актуальність даного товару.

Що треба взяти до уваги:

- статистика переглядів кожного товару оновлюється дуже часто, тому числа кількості переглядів можуть бути великими;
- статистика замовлень оновлюється набагато рідше, тому що не кожний користувач, відкривши сторінку товару, замовить його;
- статистика відгуків оновлюється ще рідше, тому що не кожний користувач залишає відгук на товар після замовлення;
- нарешті, власник товару може у будь-який час завантажити статистику з прогнозом.

Зважаючи на частоту оновлень статистики та кількість можливих запитів до серверу, розглядатися будуть прості методи прогнозування та побудови трендів, що не потребують великого часу виконання та не призведуть до перенавантаження серверу.

Отже, для порівняння були обрані наступні методи: ковзне середнє, зважене ковзне середнє, експоненціальне згладжування, подвійне експоненціальне згладжування, лінійний тренд, квадратичний тренд, поліноміальний тренд, експоненціальний, показниковий тренд, логарифмічний тренд. Для методів згладжування були використані різні налаштування вхідних параметрів. Порівнюються методи будуть за коефіцієнтом Тейла на випадкових наборах даних.

Перш за все, відкинуто методи з найбільшим коефіцієнтом Тейла та методи, що завжди мають однаковий коефіцієнт.

Результати першого експерименту відображені на рисунку 2.4.

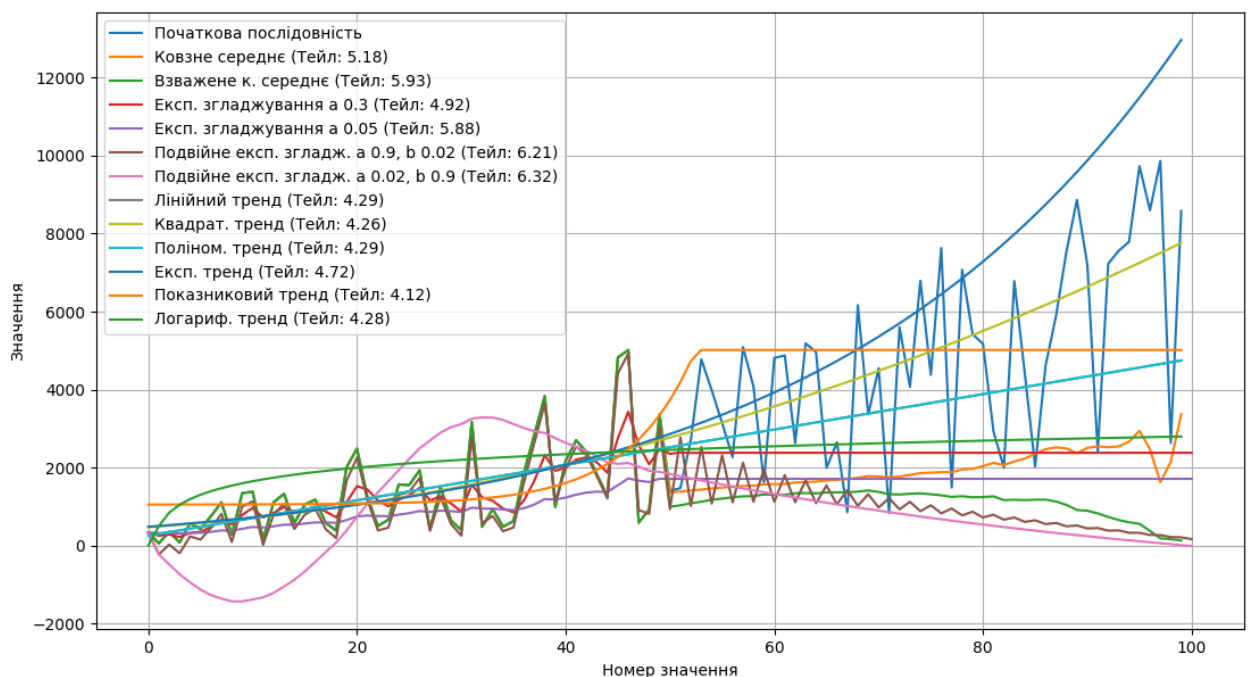


Рисунок 2.4 – Експеримент при 50 вхідних та 50 прогнозованих значеннях

Можна побачити, що коефіцієнт Тейла майже співпадає для лінійного, квадратичного та поліноміального трендів. Ця властивість

повторюватиметься при послідуєчих експериментах, тому має сенс залишити лише один з них. Квадратичний мав найменший коефіцієнт, тому залишимо його.

Подвійне експ. згладжування із будь-яким налаштуванням дає найгірший результат з великим відривом, тому приберемо також і його.

Подальші експерименти проведені відповідно із 50/50 та 90/10 відношенням тестові/прогнозовані дані із ростом даних та без.

За результатами експериментів найкраще себе показали експон. та квадратичний тренди. Їх середнє можна використати як результат.

Отже, дані для статистики товару було вирішено прогнозувати середнім значенням квадратичного та експоненціального трендів.

2.3.3 Реалізація прогнозування у python для автоматизації процесу

Реалізація обраних методів прогнозування, що використовувалась при їх порівнянні, потребує встановлених бібліотек numpy та scipy. Бібліотека numpy реалізує методи обробки наборів числових значень та є залежністю бібліотеки scipy, що містить необхідні методи оптимізації. Так, при завантаженні статистики товару, вона буде доповнена прогноз. даними на 7 днів вперед.

Реалізація методів прогнозування знаходиться у файлі stats/utils.py.

Висновок

Обраний алгоритм обчислення рейтингу товару дозволяє не зберігати всю історію оцінок, яку було би необхідно використовувати при повторному обчисленні рейтингу, а зберігати лише результат та кількість відгуків на товар.

Недоліком методу є неможливість видалення відгуку, тому що видалення відгуку не призведе до виключення оцінки цього відгуку з рейтингу товару.

Вибір простих методів прогнозування дозволяє зберегти простоту реалізації та запобігти перенавантаженню серверу. Очікується, що обчислення середнього значення результату двох методів дасть більш точну оцінку прогнозування. Дане очікування підкріплене тим фактом, що під час експериментів тренд одного з методів знаходився над приблизною середньою лінією значень, а другого – під нею.

2.4 Вибір серверу та управління

У даному розділі розглядається процес вибору серверу для хостингу веб-додатку на основі багатьох критеріїв та вибір подальшої стратегії управління сервером з урахуванням грошових та часових обмежень.

2.4.1 Змістова модель задачі

Змістова модель задачі формулюється на основі зібраної інформації про об'єкт моделювання. Модель як правило не є остаточною і може корегуватись і конкретизуватись в процесі розробки моделі. Але важливо виділити що бідь-яке корегування змістовної постановки повинне бути лише частковим [51].

Для поточної задачі змістова модель повністю наведена у вигляді тез у додатку Г.

2.4.2 Концептуальна модель задачі

Під концептуальною моделлю мається на увазі абстрактна модель предметної області, що складається з переліку взаємопов'язаних понять, що використовуються для опису цієї області, разом з властивостями й характеристиками, класифікацією цих понять, за типами, ситуацій, ознаками в даній області і законів протікання процесів в ній [52].

Повна концептуальна модель даної задачі наведена у тезах у додатку Г.

2.4.3 Математична модель задачі

Математична модель представляє собою більш конкретизовану модель (порівняно із концептуальною моделлю), та може бути представлена різними способами – це може бути як і набір графіків, так і набір таблиць або система рівнянь і нерівностей. Важливо, щоб при використанні моделі була можливість за відомими значеннями одних змінних отримати значення інших, невідомих змінних [53].

У випадку поточної предметної галузі математична модель є комплексною, тобто поєднує в собі дві менші задачі. Математичні моделі поточних задач наведені у відповідних тезах у додатку Г. Контекст моделі має наступний вигляд:

Присутній повний список серверів, представлений онлайн сервісом Digital Ocean з переліком їх параметрів та цін. Необхідно обрати такі сервери (або такий сервер), щоб остаточний вибір можна білу вважати оптимальним.

Модель другої задачі представляє собою набір станів серверу та експертних стратегій поведінки у даних ситуаціях.

Можливі стани серверу:

- θ_1 – навантаження є нормальним (очікуваним);
- θ_2 – навантаження занизьке;
- θ_3 – навантаження завелике;
- θ_4 – сервер не працює.

Можливі стратегії:

- f_1 – замінити сервер на сервер з іншою потужністю (з перенесенням бази даних та самого веб-додатку на новий сервер);
- f_2 – нічого не робити;
- f_3 – розділити навантаження (докупити додатковий сервер та перенести на нього одну з складових системи);
- f_4 – перезавантажити сервер та провести технічні роботи (із збереженням даних).

Числові значення рішень мають додатній інгредієнт у діапазоні від 0 до 10 та були призначені експертним шляхом в залежності від того, наскільки логічним є прийняття певної дії у певній ситуації.

Наприклад, нічого не робити є більш логічним рішенням якщо сервер працює нормально, ніж якщо він не працює.

Всього сума рішень для кожного стану дорівнює 10.

Всі числові значення відображені у додатку Г.

2.4.4 Вибір серверу методом аналізу ієрархій

Для початку необхідно звернути увагу на вхідний список серверів. Проблема такого списку полягає у тому, що завелика кількість опцій можуть сильно ускладнити процес застосування методу аналізу ієрархій.

На щастя задача містить грошове та часове обмеження, що дозволяє нам обмежити кількість доступних опцій.

Так, вибір серверу відбувається з урахуванням однорічного плану, у рамках якого на роботу з сервером виділяється 10000 грн, що приблизно дорівнює 500 доларам. Зважаючи на те, що кожен сервер потребує помісячної оплати, необхідно визначити максимальну доступну ціну серверу. Для цього необхідно розділити загальний капітал на 12 місяців.

Таким чином, максимальна ціна серверу = $500/12 = 41,65$ доларів.

Така сума знижує кількість доступних опцій з 56 до 6.

Тепер треба відзначити важливі та неважливі характеристики. Так як зараз кожен сервер має 7 характеристик, проводити розв'язання все ще вважається заважким.

На щастя, характеристики “\$ в годину” та “\$ на місяць” відображають однакову інформацію, що означає, що один з них можна не враховувати.

Іншим неважливим показником можна вважати варіант SSD.

Так, фінальна вибірка даних зображена у таблиці 2.7.

Для зрозумілості кожний сервер було пронумеровано від 1 до 6.

Таблиця 2.7

Вхідні дані задачі

Назва	Пам'ять	vCPU	Потік	Об'єм	\$ на місяць
Опція 1	1GB	1vCPU	1TB	25GB	\$5
Опція 2	2GB	1vCPU	2TB	50GB	\$10
Опція 3	2GB	2vCPUs	3TB	60GB	\$15
Опція 4	4GB	2vCPUs	4TB	80GB	\$20
Опція 5	8GB	4vCPUs	5TB	160GB	\$40
Опція 6	4GB	2vCPUs	4TB	25GB	\$40

Для початку необхідно побудувати таблицю ефективності критеріїв та вичислити оцінку компонентів власного вектору (далі власна оцінка) і нормалізовані оцінки вектору пріоритету (далі оцінка пріоритету) для кожного критерію, а також індекс узгодженості та відношення узгодженості.

Всі дані, а також обчислені параметри, наведені у таблиці 2.8.

Таблиця 2.8

Ефективність критеріїв

Критерії	Пам'ять	vCPU	Потік	Об'єм	\$ на місяць	Власна оцінка	Норм. оцінка
Пам'ять	1	2	1/3	1/7	1/9	0,402637	0,058226
vCPU	1/2	1	1/6	1/9	1/9	0,252619	0,036532
Потік	3	6	1	1/7	1/3	0,96964	0,140221
Об'єм	7	9	2	1	1/2	2,290172	0,331186
\$ на місяць	9	9	3	1	1	3	0,433835
Разом	20,5	27	6,5	2,3968254	2,0555556	6,9150687	1
S * N	1,1936340,986357 0,911439 0,793794 0,891772						
Lmax	ВУ	Індекс узгодженості		Відношення узгодженості		(< 10%)	
4,776996	1,12	-0,055751086		-116,71%			

Випадкова узгодженість дорівнює 1,12 (табличне значення) тому, що розглядається 5 критеріїв. Так, відношення узгодженості є меншим ніж 10%, що свідчить про адекватність та допустимість обраних пріоритетів.

Тепер необхідно вичислити оцінку всіх опцій за кожним критерієм для подальшого порівняння. ВС буде дорівнювати 1,24 (табличне значення).

Обчислення оцінок та відношень узгодженості всіх п'яти критеріїв наведені у відповідних тезах у додатку Г.

Оцінки всім критеріям відносно один одного було призначено за системою Саати. Більшість значень були призначені експертним шляхом на основі порівняння загальної потужності серверів. Останнім кроком є визначення глобального пріоритету для кожної опції. Обчислені глобальні пріоритети наведені у додатку Г. Таким чином, оптимальним вибором є опція 5. Слід також відмітити, що другим по оптимальності вибором є опція 1.

2.4.5 Вибір стратегії методами прийняття рішень за невизначеності

З причини відсутності вирогідностей наявності системи у певному стані, необхідно застосувати методи Вальда, Севіджа, Гурвіца та Бернуллі (макс). При застосуванні метода Севіджа необхідно побудувати відповідну таблицю ризиків. Так, розрахунки за цими методами приведені у таблиці 2.9.

Таблиця 2.9

Методи прийняття рішень без вирогідностей

1. Метод Вальда (Максимін)						f(max(A[i])-A[i,j])			
Ответ (max)						7	0	7	8
1	0	0	0	f	1	3	0	5	2
2. Метод Севеджа (Мінімакс)						1	2	0	3
Ответ (min)						3	6	5	0
7	6	7	8	f	2				
3. Метод Гурвіца						λ			
Ответ (max)						0,4			
2,2	4,8	2,4	3,6	f	2				
4. Метод Бернуллі									
Ответ (max)									
2,25	3,75	1,5	2,5	f	2				

Наступним кроком необхідно ввести приблизні вирогідності p . Зважаючи на приблизне навантаження на сервер та обрану у розділі 2.4.4 опцію вважається що обраний сервер більшість часу буде знаходитись у нормальному стані (θ_1). Взавши до уваги можливий нестаток активності на сервері, стан θ_2 має більшу вірогідність відносно станів θ_3 та θ_4 . Зважаючи на результат розділу 2.4.4., стану θ_4 слід надати найменший шанс відносно усіх інших станів. Таким чином, маємо приблизні вирогідності p :

$$p_1 = 0,5, p_2 = 0,3, p_3 = 0,15, p_4 = 0,05, p_1 + p_2 + p_3 + p_4 = 1$$

Зважаючи що отримані вирогідності є приблизними, можна використати метод Ходжеса-Лемана. Якщо вони є точними, можна використати методи Байєса (макс), Мінімуму дисперсії, Максимізації ймовірності, Модальний (макс), Мінімуму ентропії та комбінованого критерію. Проведені обчислення наведені у таблицях 2.10 та 2.11.

Таблиця 2.10

Методи прийняття рішень із неточними вірогідностями

					Am = (max(A)-A[i,j])				Мінімум	p
maxθ1	maxθ2	maxθ3	maxθ4		7	0	7	8	7	0,5
7	8	8	8		6	3	8	5	Дозволений	0,3
B1	B2	B3	B4	Ответ (min)	5	6	4	7	риск	0,15
6,3	2,2	6,85	6,65	f 2	5	8	7	2	8	0,05

Таблиця 2.11

Методи прийняття рішень із точними вірогідностями

1. Метод Байеса								
B1	B2	B3	B4	Ответ (max)	Середнє Макс Мін			
1,7	5,8	1,15	1,35	f 2	2,5	5,8	1,15	
2. Метод Мінімуму дисперсії								
σ1	σ2	σ3	σ4	Ответ (min)				
0,61	6,46	1,6275	2,8275	f 1				
2.1. Модифікація 1 (серед)								
σ1	σ2	σ3	σ4	Ответ (min)				
1,25	17,35	3,45	4,15	f 1				
2.2. Модифікація 2 (макс)								
σ1	σ2	σ3	σ4	Ответ (min)				
17,42	6,46	23,25	22,63	f 2				
2.3. Модифікація 3 (мін)								
σ1	σ2	σ3	σ4	Ответ (min)				
0,9125	28,0825	1,6275	2,8675	f 1				
Метод Максимізації ймовірності розподілу оцінного функціонала								
φ1	φ2	φ3	φ4	Ответ (max)	Альфа 1 льфаАльф:			
0	0,8	0,15	0,05	f 2	0	8	4	
4. Метод Модальний								
m1	m2	m3	m4	Ответ (max)	Макс. p			
1	8	1	0	f 2	0,5			
5. Метод Мінімуму ентропії								
entro1	entro2	entro3	entro4	Ответ (min)				
1,293549	0,759206	0,837897	0,848686	f 2				
6. Метод Комбінованого критерію								
k1	k2	k3	k4	minλ	maxλ	λ1	λ2	λ3
0,825714	0,838903	0,448305	0,391935	0,391935	0,838903	0	1	0,5
6.1. λ1								
1*λ	k2*λ	k3*λ	k4*λ	Ответ (max)				
2,89	33,64	1,3225	1,8225	f 2				
6.1. λ2								
1*λ	k2*λ	k3*λ	k4*λ	Ответ (max)				
-0,61	-6,46	-1,6275	-2,8275	f 1				
6.1. λ3								
1*λ	k2*λ	k3*λ	k4*λ	Ответ (max)				
1,14	13,59	-0,1525	-0,5025	f 2				

Таким чином: серед методів із невідомими вирогідностями стратегії зустрічались: $f_1 - 1$, $f_2 - 3$, $f_3 - 0$, $f_4 - 0$ разів.

Серед методів із приблизними: $f_1 - 0$, $f_2 - 1$, $f_3 - 0$, $f_4 - 0$ разів.

Серед методів із точними: $f_1 - 4$, $f_2 - 7$, $f_3 - 0$, $f_4 - 0$ разів.

Всього стратегії зустрічались: $f_1 - 5$, $f_2 - 10$, $f_3 - 0$, $f_4 - 0$ разів.

Підсумки результатів по всім методам наведені у відповідних тезах у додатку Г. Згідно з результатами, стратегія f_2 є більш ефективною.

2.4.6 Аналіз отриманих результатів

На основі результатів, отриманих у розділі 2.4.4, оптимальним рішенням є придбання серверу із найбільшим об'ємом дискового простору.

Перш за все, такий результат свідчить про перевагу ефективних характеристик серверу над його ціною. Але такий вибір серверу залишив лише 396 грн на обслуговування купленого серверу.

У ході пошуку оптимальної стратегії виявилось, що хоч знайдена оптимальна стратегія (нічого не робити) і не використовує залишений капітал, друга по ефективності стратегія (замінити сервер) передбачає витрати, що можуть не задовольнити обмеження, виставлене залишковим капіталом.

Таким чином, має сенс переглянути вибір серверу та звернути увагу на сервер, що виявився другим по оптимальності – сервер з найнижчою ціною.

Висновок

Під час підрахування ефективності критеріїв серверів біло отримано одразу два оптимальні результати – сервери з мінімальною та максимальною цінами. Проаналізувавши можливі ризики можна сказати, що перший вибір хоч і є найбільш ефективним, він також є найбільш небезпечним, тому що не передбачає можливості фінансових витрат у випадку їх необхідності.

А вирішення другої частини задачі додатково показало значну можливість такого ризику.

2.5. Презентація бізнес-проекту з розширення функціоналу сервісу

У даному розділі розглядатиметься організація та презентація бізнес-проекту, альтернативи його впровадження та інші залежні показники.

Презентація бізнес-проекту дозволить визначити наступні кроки та задачі, а також обґрунтує впровадження самого проекту.

Слід зауважити, що розглядувані у даному розділі альтернативи та можливості є обмеженими задля спрощення обчислень та міркувань щодо окремих задач з наступних розділів, а також для спрощення бізнес-проекту в цілому.

2.5.1. Визначення проекту

Платформа, що розглядається у даній науковій роботі, є інтернет-маркетплейсом, спрямованим на продаж меблів. Насамперед така платформа може передбачати доставку куплених товарів користувачеві.

Сучасні торгівельні платформи передбачають відстежування товару на дорозі від продавця до покупця як мінімум, та інтеграції з поштовими службами, власну систему перевезень і гарантії як максимум.

Тому, було виділено можливе розширення поточного функціоналу веб-додатку та сервісу загалом у даному плані. Це може означати як лише зміни у роботі веб-додатку сервісу (бек-енд, фронт-енд та база даних), так і додаткові зміни у структурі всього підприємства, наприклад:

- додаткове відділення з перевезень;
- співпраця або певні контракти з іншими компаніями або сервісами;
- побудова додаткових філіалів підприємства;
- програмна (та платна) інтеграція із поштовими службами;
- додаткові робочі місця для відділу підтримки користувачів.

З приводу наявності великої кількості рішень даної задачі, необхідно спочатку обґрунтувати мету задачі, а потім розглянути та порівняти можливі альтернативи.

2.5.2. Обґрунтування проекту

Головним аргументом впровадження системи доставок, як вже було зазначено раніше, є актуальність та популярність даної ідеї серед сучасних онлайн торгівельних платформ.

Кожний інтернет-магазин фізичних товарів передбачає способи отримати куплені речі. Частіше за все покупці або самі забирають товар з рук продавця або з точок видачі, або використовують поштовий сервіс для доставки товару у найближче поштове відділення або поштомат. Онлайн-сервіси з продажу ліків взагалі дозволяють забирати заздалегідь оплачені препарати з обраних лікарень.

З найпопулярніших типів отримання товару на сьогоднішній день виділяють [54]:

- доставку кур'єром;
- транспортні компанії;
- поштові сервіси;
- самовивіз.

У рамках даної роботи розглядатимуться поштові сервіси, а також впровадження власного відділення перевезень у якості транспортної компанії.

2.5.3. Визначення альтернатив рішень: поштові служби

Ринковий аналіз поштових служб станом на 2020 рік показує «позитивні прогнози розвитку української економіки 3-4% на рік разом з продовженням активного зростання споживання, а також активне зростання електронної комерції на 16-18% на рік», що є основними факторами для зростання всього ринку експрес-доставки [55].

Даний аналіз також наведений у вигляді тез у додатку Г.

Хоча у випадку інтернет-маркетплейсу імплементація даних методів не є прямою необхідністю, має сенс впровадження певного рівню інтеграції з

ними для полегшення взаємодії як покупця з продавцем, так і обох типів користувачів із сайтом та, у окремих випадках, із сервісом доставки.

Це також полегшує комунікацію стосовно статусу замовлень, тому що додатково дозволяє покупцю відстежувати процес доставки товару. Така функція може контролюватися як продавцем, так і самим сервісом доставки. Чудовим прикладом першого варіанту є українська торгівельна платформа rrom.ua [56], прикладом другої, більш складної імплементації, є інтернет-маркетплейс одягу shafa.ua [57].

Так, імплементація систем взаємодії з поштовими сервісами доставки потребує часу та ресурсів, особливо зважаючи на те що кожен такий сервіс працює по-різному і є неможливим побудова однієї системи однакової взаємодії з будь-яким сервісом.

З цього приводу має сенс обмежити список сервісів, з якими буде будуватись взаємодія. Вибір можна провести на основі популярності сервісу та його рівню цифровізації. Так, згідно з The Page, найпопулярнішим поштовим сервісом за останній час є Нова Пошта [58, 55]. Повний список найпопулярніших поштових сервісів та їх порівняння приведено на рисунках 2.5-2.6 і в таблиці 2.12 [59].

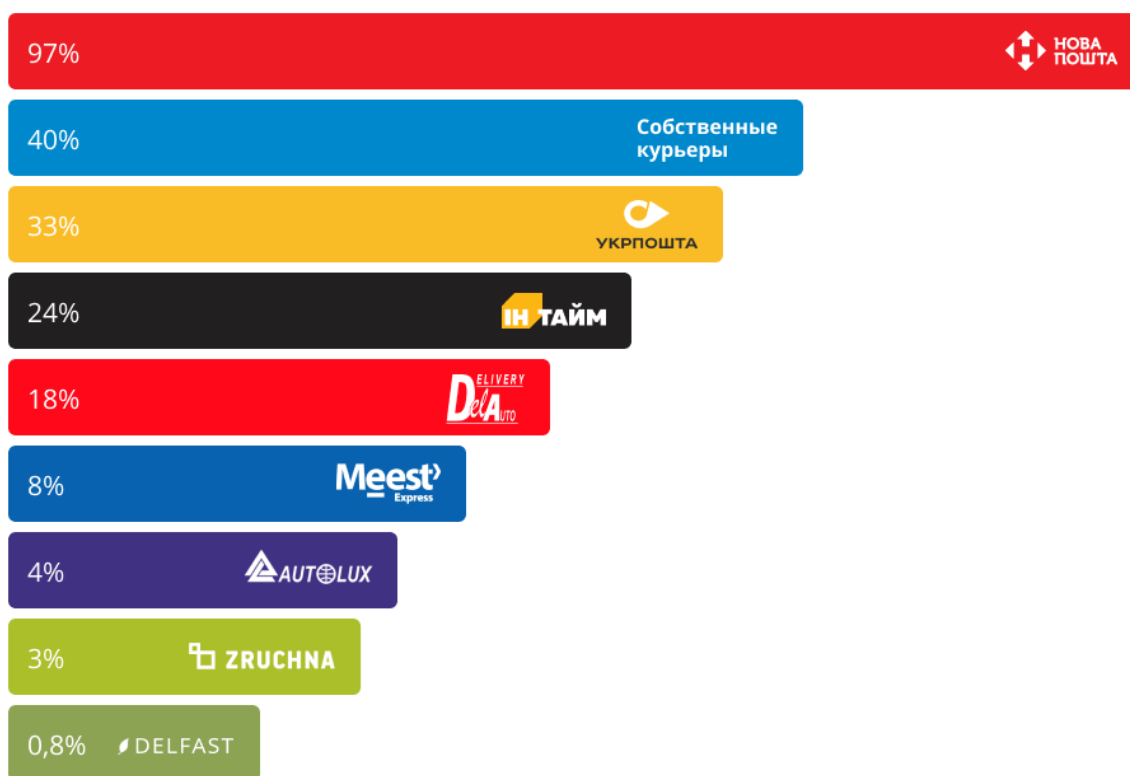


Рисунок 2.5 – Популярність служб доставки серед інтернет-магазинів

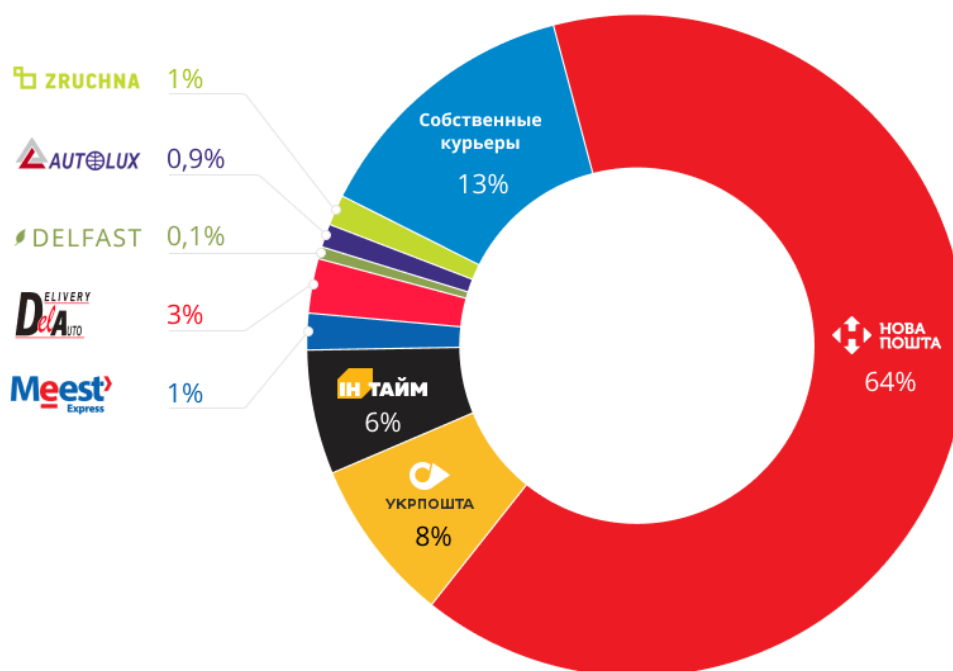


Рисунок 2.6 – Відсоток замовлень, доставлених окремою службою

Таблиця 2.12

Особливості служб доставки

	Новая Почта	УкрПочта	Интайм	Мист Экспресс	Деливери	Делфаст	Автолюкс	Zruchna
Количество отделений	2440	свыше 12800	645	2578	1800	6	200	3
Сроки доставки по Украине	1-2 дня	1-6 дней	1-2 дня	1-3 дня	1-3 дня	несколько часов	1-3 дня	1 день
Наличие мобильного приложения	+	+	+	+	+	+	+	-
API	+	+	+	+	+	+	+	+
Программа лояльности	+	+	-	-	+	-	+	-
Международная доставка	+	+	+	+	+	-	+	-
Фулфилмент	+	-	+	-	-	-	-	+
Отправка нескольких товаров на выбор	-	-	+	-	-	-	-	+
Доставка по временным интервалам	+	-	+	-	+	+	+	+
Перевозка крупногабаритных грузов	+	+	+	+	+	-	+	-

Можна побачити, що найпопулярнішими методами доставки є як і вказана раніше Нова Пошта, так і поштова служба ІнТайм, Укрпошта, а також власні кур'єри.

З приводу того, що інтеграції з останнім варіантом реалізувати майже неможливо, залишаються три кандидати – Нова Пошта, Укрпошта та ІнТайм.

На щастя, для полегшення інтеграції дані сервіси мають власні API, призначені для взаємодії з веб-додатками.

Так, Нова Пошта має власний портал для розробників з детальною документацією щодо використання їх API [60]. Укрпошта має схожу інтеграцією з повною документацією – Укрпошта API [61].

Нажаль, на даний момент ІнТайм не має повноцінної документації свого API, але присутні майже готові plug-in інтеграції, як наприклад очікувана інтеграція від APIxDrive [62].

З цього приводу, оптимальним рішенням буде зачекати на готову інтеграції з ІнТайм, та сфокусуватись на інших варіантах.

Так, у разі використання лише поштових служб при доставці можна було б обмежитись лише програмними змінами у роботі веб-додатку.

Але продаж меблів, серед іншого, вказує на можливу необхідність спеціалізованих способів перевезень, при яких використання звичайних поштових служб не є можливим.

2.5.4. Визначення альтернатив рішень: відділ перевезень

Деяка цільова продукція даного інтернет-маркетплейсу – меблі – зазвичай має розміри та вагу занадто великі для звичайних поштових служб. Це насамперед означає, що окрім як давати користувачеві самому обирати та зв'язуватися з необхідною транспортною компанією, або забирати товар власноруч, даний сервіс може надавати опцію власної послуги перевезень, яка може бути як корисною для користувача, так і для сервісу:

За наявності даної опції, користувач отримує:

- додаткову зручність;
- гарантовану надійність та збереженість товару;
- легший процес оформлення без необхідності додаткових операцій.

Сервіс, в той же час, отримує:

- додаткове джерело прибутку;
- ріст компанії та робочих місць;
- ріст популярності сервісу а також його зручності та якості в очах користувачів.

користувачів.

Слід зазначити, що введення такого відділу передбачає і певні затрати:

Одноразові та семі-регулярні:

- організація нового відділення;
- добудова гаражів під вантажівки;
- самі вантажівки та інший службовий транспорт;
- інструменти обслуговування нової техніки;
- пошук нового, кваліфікованого персоналу.

Регулярні:

- зарплата новим працівникам;
- паливо, ремонт та обслуговування транспорту.

У результаті можна побачити, що прибутковість даного рішення та його підтримка будуть тісно пов'язані насамперед з приводу кількості можливих фінансових затрат.

2.5.5. Презентація проекту

У якості презентації бізнес-проекту використовуватиметься структурований список питань та відповідей, що адресують його кожний аспект.

Список даних питань приведений нижче:

- проблема, на розв’язання якої проект спрямований;
- актуальність;
- мета проекту;
- альтернативи розв’язання проблеми;
- критерії вибору альтернативи та її обґрунтування;
- цільові групи (на яку категорію населення спрямовані результати проекту);
- залучені організації (інвестори, громадські організації, змі, навчальні заклади, підрядники тощо);
- продукт проекту та очікувані результати проекту (короткострокові, довгострокові);
- основні етапи проекту (назва етапу, відповідальна особа, термін виконання);
- опис необхідних ресурсів (фінансові, матеріальні, технічні, людські, інші).

Так, у рамках наукової роботи використовуватимуться припущення та узагальнення щодо точних фінансових та людських ресурсів. См. табл. 2.13.

Таблиця 2.13

Презентація бізнес-проекту по впровадженню системи доставок товару

Проблема, на розв’язання якої проект спрямований	Відсутність будь-яких систем доставки та відстежування товару меблевого інтернет-маркетплейсу.
Актуальність	Весь час подальшої роботи компанії та сервісу. Дані зміни закликані назавжди підвищити зручність, якість та функціональність сервісу.
Мета проекту	Впровадити системи доставки або

	<p>відстежування доставки товару. Таким чином купівля та продаж товарів стануть швидшими та надійнішими, що також підвищить відсоток успішних продажів. Це не тільки підвищить популярність сервісу та його якість і зручність в очах користувачів, але й забезпечить ріст та розвиток самої компанії.</p>
Альтернативи розв'язання проблеми	<ol style="list-style-type: none"> 1. Впровадження системи відстеження товару; 2. Інтеграція з API Нової Пошти; 3. Інтеграція з API Укрпошти; 4. Інтеграція з API ІнТайм; 5. Впровадження відділу з перевезень; 6. Залишення опцій самовивозу/власного вибору транспортних компаній та поштових служб без інтеграції з ними.
Критерії вибору альтернативи та її обґрунтування	<p>Головними критеріями вибору альтернативи є зручність імплементації, зручність використання, грошові витрати, популярність, необхідність та задоволеність клієнтів.</p>
Цільові групи (на яку категорію населення спрямовані результати проекту)	<ol style="list-style-type: none"> 1. Безпосередні клієнти компанії (продавці та покупці що користуються сервісом); 2. Потенційні робітники нових відділень по перевезенням та підтримки; 3. Сама компанія.
Залучені організації (інвестори, громадські організації, ЗМІ, навчальні заклади, підрядники тощо)	<ol style="list-style-type: none"> 1. Інвестори компанії (фінансування робіт); 2. Відділ кадрів компанії (пошук фахівців); 3. Технічний відділ (впровадження змін до веб-додатку сервісу); 4. ЗМІ (реклама робочих місць); 5. Робочі підрядники (планування маршрутів).
Продукт проекту. Очікувані результати проекту (короткострокові, довгострокові)	<p>Короткострокові:</p> <ol style="list-style-type: none"> 1. Маршрути перевезень між всіма сусідніми філіалами компанії; 2. Відділення, відповідні за такі перевезення, у кожному охопленому філіалі. <p>Довгострокові:</p> <ol style="list-style-type: none"> 1. Додаткові фінансові прибутки; 2. Зріст популярності сервісу.
Основні етапи проекту (назва етапу, відповідальна особа, термін виконання)	<ol style="list-style-type: none"> 1. Загальні планування та координація проекту, 1 особа, 1 тиждень; 2. Прокладання маршрутів, 2-3 особи, 3 тижні; 3. Формування відділів та пошук персоналу, 2-4 особи, 4 місяці; 4. Організація перевезень, 1-2 особи, 1 місяць.

Опис необхідних ресурсів (фінансові, матеріальні, технічні, людські, інші)	Фінансові – грошове забезпечення фахівців компанії та перевезень. Матеріальні – робочі офіси відділень, вантажівки для перевезень. Людські – фахівці, що займатимуться перевезеннями та іншими обов'язками відділу.
Орієнтовний бюджет проект за джерелами фінансування (загальна сума, у т.ч. власні кошти, грант, очікування від інших організацій)	Загальна сума проекту складається із коштів закуплених необхідних матеріальних ресурсів (наприклад, вантажівок), а також коштів, витрачених на забезпечення залучених фахівців заробітною платою. Очікування від інвесторів – приблизно 50%, щонайменше 35% загальної суми проекту.

Треба зауважити, що з перелічених у презентації альтернатив не обов'язково вибирати лише одну – більш оптимальним рішенням може бути певна комбінація даних рішень.

Крім того, немалу роль грають додаткові аспекти та варіанти імплементації кожної з альтернатив.

Висновок

У даному розділі було визначено та обґрунтовано бізнес-проект, що розглядатиметься в наступних розділах.

Після проведення ринкового аналізу поштових служб, виявлення їх основних характеристик з подальшим їх порівнянням, а також фільтрування списку опцій за даними характеристиками та можливістю інтеграції, було вирішено реалізувати інтеграцію лише з Новою Поштою та Укрпоштою. Взаємодію з із поштовою службою ІнТайм можна також провести пізніше, по факту наявності документації або готових інтеграцій. Реалізації інтеграції з іншими поштовими службами також можна реалізувати в майбутньому у випадку підвищення попиту на ці служби.

Також було розглянуто можливість впровадження власного відділу для перевезень великого вантажу, а також наведені його переваги (як для користувачів так і для сервісу) та недоліки (у вигляді списку можливих фінансових затрат на організацію та підтримку такого відділу). Дане рішення було протипоставлено незалежним транспортним компаніям у вигляді власного, більш доступного аналогу.

При побудові таблиці презентації бізнес-проекту було визначено та наведено остаточний перелік альтернатив рішення його задачі. Точні фінансові та людські показники було опущено.

Вибір даних альтернатив, а також певних аспектів та стратегій щодо впровадження оптимальних рішень реалізації будь-яких змін у роботі сервісу обумовлений **рішенням конфліктної ситуації**, де стратегії сторін та результат обчислень представляє собою можливі альтернативи та їх наслідки як з боку підприємства та сервісу, так і з боку користувачів.

2.6. Теорія ігор у конфліктній ситуації сервісу і користувача

Даний розділ розглядатиме вибір набору альтернатив з попереднього розділу, а також ряд додаткових деталей щодо впровадження систем доставки та відстеження товару на сайті, у вигляді конфліктної ситуації, де сторонами конфлікту є компанія та користувачі, а розв'язком задачі є оптимальний набір рішень, що максимізує як прибуток компанії, так і привабливість сервісу для користувачів. Аналіз результату вкаже на наступні дії щодо впровадження обраних альтернатив.

2.6.1. Постанова задачі

Необхідно визначити можливі деталі імплементації альтернатив, наведених у попередньому розділі. Кожна деталь буде вважатись окремою стратегією щодо реалізації рішення і матиме свої переваги та недоліки. Слід зауважити, що список цих переваг та недоліків є різним для компанії та її клієнтів.

Отже, присутня компанія (сторона А), що планує впровадити систему доставки товару на сайт її сервісу для підвищення прибутку та популярності сервісу, а також для збільшення його зручності.

Користувачі сервісу виступають у якості сторони Б.

Присутній список стратегій, які компанія може реалізувати для створення такої системи, і які мають різну користь для обох сторін, що беруть участь у задачі.

Розв'язком задачі є оптимальний набір рішень, що намагається максимізувати всі показники передбачені нею.

Логічно буде припустити, що знайдений розв'язок не є об'єктивно найкращим, загалом тому що важкою задачею є визначити, нормалізувати та прирівняти корисність кожної стратегії. У зв'язку з цим, обрані значення корисності можуть бути приблизними або неточними.

2.6.2. Обґрунтування застосування методу матричних ігор

Конфлікт – це ситуація з кількома учасниками, цілі яких не збігаються і дії яких не є абсолютно незалежними [43].

Наведену у постанові задачі ситуацію можна вважати конфліктною з наступних причин:

1. Дана ситуація представляє собою взаємини між компанією і клієнтом, де кожне рішення може мати різну користь для кожної сторони;
2. Ціллю компанії є підвищення прибутку (мінімізація втрат), ціллю клієнта (користувача) – отримати швидке та зручне обслуговування при використанні сервісу;
3. Дії обох сторін у даній ситуації не є абсолютно незалежними – вони обумовлені остаточними рішеннями компанії та залежать від реакції користувачів – набір стратегій буде програшним, якщо користувачі відмовляться користуватись новим функціоналом або, у найгіршому випадку, сервісом взагалі – таким чином або просто не надавши шуканого прибутку компанії, але й знизивши його.

Розглянемо ситуацію з боку фундаментальних понять теорії ігор [43]:

- конфлікт і його сторони: компанія (сторона А) впроваджує функціонал для користувачів (сторона Б);
- прийняття рішення в конфлікті: сторона А впроваджує рішення щодо сервісу, користувачі реагують на дані зміни;
- оптимальність прийнятого рішення: прибутковість рішень та задоволеність користувачів не є незалежними показниками.

Нарешті, можна скласти нормальну форму гри, описавши сторони конфлікту у вигляді матриць виграшів [43]: перший гравець (сторона А) теоретично має всього m стратегій, а другий (сторона Б) – відповідно n стратегій, тому таку гру можна описати, задавши виграші гравців двома матрицями.

Таким чином, дана задача не тільки відповідає визначенню конфліктної ситуації у теорії ігор, а також здатна бути підставленою у її фундаментальні поняття і відображена у вигляді матриць виграшів.

Так, дану ситуацію можна вважати матричною грою з двома гравцями, кожний з яких має набір варіантів «ходів». Залишається побудувати модель задачі та визначити її вхідні дані.

2.6.3. Побудова моделі та визначення вхідних даних

Математична модель конфліктної ситуації називається грою, сторони, що беруть участь у конфлікті, – гравцями.

Гравцем А у даній моделі, як вже біло зазначено раніше, є компанія.

Гравцем В є користувач сервісу, що впроваджує компанія.

Для початку розглянемо стратегії обох сторін. Особистостями стратегій гравця А є:

1. Вплив на процес комунікації між продавцем та покупцем – наскільки обрана стратегія полегшує або поскладнює взаємодію, скільки додаткових кроків додає. Ця особистість цілком впливає на виграш (зацікавленість) гравця 2, тобто користувача;

2. Складність обробки або праці із обраною стратегією, що є небажаним фактором або таким, який необхідно звести до мінімуму;

3. Складність підтримки та моніторингу процесу хоч і ускладнює роботу для підприємства, поліпшує якість сервісу та зручність, що сподобається користувачам;

4. Навантаження на сервери – аспект що майже завжди спершу негативно впливає на роботу сервісу, а потім – і на користувачів. Слід завжди враховувати серверну спроможність сервісу та кількість і частоту обчислень, необхідних для роботи стратегії;

5. Автоматизація – аспект що значно полегшує процеси як підприємству, так і користувачам. Але, так як відсутність автоматизації при ідеальному

виконанні стратегії також майже не заважає користувачам, даний аспект можна вважати виграшем лише для підприємства;

б. Фінансові втрати – аспект, що є програшним для підприємства, тому бажано звести його до мінімуму.

Так, для усунення проблеми навантаження на сервери, необхідно вибрати сервер (або сервери), що спроможні справлятися із прогнозованим навантаженням. Якщо таких серверів немає можливості встановити, необхідно прибрати стратегії що можуть перевищити допустиме навантаження.

Більше про вибір серверу та стратегії його управління наведено у розділі 2.4.

Сервер, обраний для сервісу методом ієрархій, та стратегія його управління, що передбачає неактивність у більшості випадків, можуть бути серйозним обмеженням при фільтруванні опцій.

Також слід мати на увазі модель бази даних, побудованої та нормалізованої у розділі 2.1 – кількість звертань до неї необхідно знизити до мінімуму, а особливості її структури також можуть стати обмеженням при виборі стратегій.

Нарешті, показник автоматизації можна вирішити програмним методом, створив власний API для організації перевезень або під'єднавши вже існуючий для роботи із певним поштовим сервісом, що обумовлюють деякі з наведених у попередньому розділі альтернатив.

Так, остаточний список стратегій гравця А включає:

- A1: Власна система відстежування товару;
- A2: Інтеграція з поштовими службами;
- A3: Власний відділ перевезень;
- A4: Розширення бази даних;
- A5: Нотифікації користувачів;
- A6: Точки видачі товару;
- A7: Відсутність інтеграцій;

– A8: Відсутність системи відстежування.

Список стратегій гравця В є реакціями на стратегії гравця А та включає:

– B1: Купівля невеликого товару;

– B2: Купівля великого товару;

– B3: Відстежування товару;

– B4: Отримання товару;

– B5: Використання власних методів доставки замість передбачених сервісом.

Так, поставивши у відповідність кожній стратегії А кожну стратегію В отримаємо матрицю виграшів. Таку матрицю необхідно побудувати для кожного з двох гравців.

Слід зауважити, що значення виграшу, або значення корисності, згадані на початку даного розділу є приблизними абстрактними значеннями визначеними порівняльним методом, де 0 – відсутність виграшу, а 1 – умовна одиниця виграшу, що відповідає або безумовному прибутку гравця А, або задоволеності гравця В. Кожна стратегія може мате значення виграшу більше за 1, порівняно один до одного. Такі порівняння є також абстрактними та можуть бути неточними.

Результуючі таблиці виграшів зображені на таблиці 2.14.

Таблиця 2.14

Вхідні дані матричної гри

		Гравець А										Гравець В							
\		A1	A2	A3	A4	A5	A6	A7	A8	\		A1	A2	A3	A4	A5	A6	A7	A8
B1		-0,3	-0,1	-1	-0,1	-0,1	-0,5	0,5	1	B1		0,5	1,5	0,5	0,1	0,1	1	-0,2	0
B2		-0,3	-1	-0,3	-0,1	-0,1	-0,5	1	0,5	B2		0,5	0	1,5	0,1	0,1	1	0	-0,4
B3		-0,2	-0,1	-0,5	-0,1	-0,1	-0,5	0,5	0,5	B3		1	1	1	0,3	0,3	0,1	-0,1	-0,1
B4		-0,2	-0,1	-0,5	-0,2	-0,1	-0,3	0,5	0,5	B4		1	1	1	0,1	0,1	1,5	-0,1	-0,1
B5		-1	-1	-1,5	-0,7	-1	-1,5	1	1	B5		0	-0,2	-0,5	0,1	0	0	0	0

Матриць виграшів всього дві, тому дану гру можна вважати біматричною.

2.6.4. Обчислення матричної гри

Так, маємо дві матриці вигравів, гравці-власники яких мають хоч і різні, але не повністю незалежні цілі. З таблиці 2.24 можна побачити, що таблиця вигравів гравця А майже повністю наповнена від'ємними значеннями, тобто програшами. Це, насамперед, пов'язано із сукупністю фінансових витрат та робочим навантаженням, що необхідне для реалізації більшості дій гравця А.

Вони (програші) відрізняються в залежності від дій гравця В, тому що використання впроваджених дій гравця А гравцем В «окупає» ці дії. Таким чином, дія гравця В «використання власних методів доставки замість передбачених сервісом» (В5) є найменш привабливою та найбільш програшною для гравця А, тому що у такому разі більшість його дій є безрезультатними, та не принесуть очікуваного прибутку.

Отже, остаточний вигреш компанії також залежний від реакції користувача, а вигреш користувача – від зручності дії компанії. Звідси виходить, що загальний вигреш кожної стратегії є сумою вигравів обох гравців. Матриця загальних вигравів зображена у таблиці 2.15.

Таблиця 2.15

Матриця загальних вигравів

\	A1	A2	A3	A4	A5	A6	A7	A8	Сум
B1	0,2	1,4	-0,5	0	0	0,5	0,8	0,2	2,6
B2	0,2	-1	1,2	0	0	0,5	0,6	0,2	1,7
B3	0,8	0,8	0,5	0,2	0,2	-0,4	0,4	0	2,5
B4	0,8	0,8	0,5	-0,1	0	0,7	0,4	0,2	3,3
B5	-1	-1,2	-2	-0,6	-1	-1,5	1	0,5	-5,8
Сум	1	0,8	-0,3	-0,5	-0,8	-0,2	3,2	1,1	

З матриці загальних вигравів також можна побачити, що дія В5 приносить менший вигреш у більшості випадків обом гравцям.

Нарешті, необхідно провести пошук сідлових точок. Так як загальні значення знаходились сумою вигравів, цілі гравців є залежними і можлива кооперативна гра, сідлові точки будуть знаходитись методом максимакс.

Таблиця 2.16

Пошук оптимальних стратегій

\	A1	A2	A3	A4	A5	A6	A7	A8	
B1	0,2	1,4	-0,5	0	0	0,5	0,8	0,2	1,4
B2	0,2	-1	1,2	0	0	0,5	0,6	0,2	1,2
B3	0,8	0,8	0,5	0,2	0,2	-0,4	0,4	0	0,8
B4	0,8	0,8	0,5	-0,1	0	0,7	0,4	0,2	0,8
B5	-1	-1,2	-2	-0,6	-1	-1,5	1	0,5	1
	0,8	1,4	1,2	0,2	0,2	0,7	1	0,5	

З таблиці 2.16, де відображено максимуми по кожному ходу обох гравців, можна побачити, що розглядувана гра має 5 сідлових точок.

Деякі пари стратегій взаємовиключні одна з іншою, тому стратегії з меншим результатом буде виключено.

Взаємовиключні стратегії наведено нижче:

- A2B1 та A3B2 взаємовиключні з A7B5 та мають більше значення;
- A1B1 взаємовиключна з A8B5 та має більше значення.

Отже, оптимальними стратегіями є:

A1B3, A1B4, A2B1, A3B2

Іншими словами, оптимальним рішенням стане імплементація наступних рішень:

- A1: Власна система відстежування товару;
- A2: Інтеграція з поштовими службами;
- A3: Власний відділ перевезень.

Тепер необхідно детально проаналізувати даний результат.

2.6.5. Аналіз результатів матричної гри

Так, основним шуканим результатом задачі є набір рішень, що необхідно (рекомендується) імплементувати. Але можна побачити, що результат також надає дані о діях користувача та очікуваності (корисності) даних рішень.

Наприклад, система відстежування хоча і не є критичною функцією (і можливо не самою прибутковою), але:

1. Має невеликі затрати (фінансові і трудові);

2. Легко імплементується;

3. Корисна користувачеві одразу на багатьох діях та при використанні будь-якого методу доставки (на відміну від інтеграції з поштовими службами та впровадження відділу перевезень).

Крім того, згідно з таблицею вигащів користувача з задачі, відмова від даної ф-ії має більший вплив на користувача, ніж відмова від імплементції інших ф-ій, що стали результатом задачі.

Висновок

Було проведено пошук оптимальної комбінації рішень щодо вирішення задачі бізнес-проекту методом матричних ігор.

Отримане рішення складається із альтернатив, що необхідно впровадити для виконання мети бізнес-проекту. Крім того, воно показує очікувану реакцію користувачів на обрані зміни, що дозволяє провести додаткові дії щодо її покращення.

Функціями, що, за результатами задачі, рекомендується реалізувати, є:

– власна система відстежування товару;

– інтеграція з поштовими службами;

– власний відділ перевезень.

Це також обґрунтовано ціновим обмеженням компанії та системою ціноутворення, розробка якої відображена у вигляді тез у додатку Г.

Подальший детальний аналіз результатів задачі виявив значну користь першої функції, що не є очевидним результатом з коефіцієнтів, що мали результатні стратегії в задачі.

Крім того, за результатами обчислень, обрана стратегія передбачає організацію власних перевезень, тому необхідно розробити надійну мережу

таких перевезень між філіалами підприємства, наприклад за допомогою **транспортної задачі.**

2.7 Транспортна задача для оптимізації перевезень

У даному розділі розглядатиметься знаходження оптимального плану перевезення між умовними точками впроваджуваного відділу з перевезень товару мебельного інтернет-маркетплейсу, а також можливими додатковими точками його видачі.

Результатом стануть числові показники пріоритетності шляхів між точками, що покажуть оптимальні маршрути для швидших або стабільніших перевезень.

2.7.1 Постановка задачі

Існує відділ меблевого інтернет-маркетплейсу з перевезень товару. Даний сервіс існує по всій країні, тому необхідно залучити роботу філіалів в усіх областях для організації перевезень по країні.

Це передбачає певну кількість точок, між якими можна встановити сталі маршрути перевезень. Такі маршрути можна оптимізувати для часткового полегшення (прискорення) перевезень. Крім того, можуть бути присутні спеціалізовані точки видачі товару, між якими також можна прокласти сталі маршрути, наприклад на випадок якщо товар було доставлено у неправильну точку та необхідна швидка переадресація.

Слід очікувати, що не кожний найшвидший шлях є надійним, але зважаючи на загальний масштаб міжобласних перевезень можна зважати меншу увагу на даний фактор, тому що на міжміських лініях рідко можуть статися транспортні затори. В тому, мати запасні маршрути також не є зайвим.

Нарешті, додатково можна буде проаналізувати загальну частоту замовлень товару великого розміру, що на основі даного показнику встановити приблизну частоту використання прокладених маршрутів.

Для визначення реальної ефективності цих шляхів необхідно провести аналіз отриманих при рішення задачі результатів.

2.7.2 Обґрунтування вибору методу для рішення даної задачі

Транспортна задача направлена на пошук оптимальних навантажень між точками відправлення та точками видачі, що є ціллю поставленої задачі.

За точки відправлення та точки прибуття можна взяти філіали компанії що мають відділи перевезень, а також можливі точки видачі у якості додаткових точок прибуття.

Необхідно також взяти до уваги, що у рамках доставки маршрути, що прокладаються між відділом та продавцем/покупцем, враховуватись не будуть, тому що їх не є можливим спрогнозувати та вони є у більшості випадків одноразовими.

З приводу того, що відділи лише починають свою діяльність та буде нелегко передбачити навантаження на них, вхідні дані будуть приблизними, а число самих відділів – невеликим.

Таким чином, можна побудувати матрицю тарифів або матрицю навантажень даних маршрутів. Слід зауважити, що такі дані як вага, або навантаження, є приблизними, особливо зважаючи на експериментальну роль даних відділів.

Отже, поставлену задачу можна укласти в рамки транспортної задачі для знаходження оптимального плану перевезень. Тепер, для рішення даної задачі, необхідно побудувати вхідну модель та визначити вхідні дані.

2.7.3 Побудова моделі та визначення вхідних даних

З тої причини, що кожний відділ може приймати товари, а може відправляти, то кількість точок відправлення та точок прибуття буде однаковою.

Як приклад, товар, що перебуває у Львівській області, замовлений з доставкою у Київську, спочатку дійде до Львівського відділу, звідки піде у Київський, що робить Львівський відділ – точкою відправлення, а Київський

– точкою прибуття. Але у протилежній ситуації, де товар піде із Київської в Львівську, відповідні відділи «обмінюються ролями».

Слід зауважити, що у випадку, коли точка відправлення та прибуття співпадає за відділом, навантаження на такий шлях дорівнює 0, тому що доставка у рамках одного й того ж самого відділу є несуттєвою.

Щодо кількості відділів перевезень – початковою кількістю було вирішено взяти 6 відділів перевезень та 0 точок видачі – таке рішення було прийнято щоб спочатку побачити реальний приблизний попит на дані перевезення, та поступово додавати відділення згодом.

Таким чином, маємо 6 точок відправлення та 6 точок прибуття. Побудуємо матрицю тарифів всіх точок задачі.

З приводу того, що товар приходить та уходить за замовленнями, сумарна кількість запасів повинна дорівнювати сумарній потребі, наприклад:

$$\sum a = 50 + 20 + 70 + 40 + 50 + 50 = 280$$

$$\sum b = 40 + 90 + 10 + 50 + 70 + 20 = 280$$

Дані рівняння є необхідною та достатньою умовою розв'язання задачі.

Так як запаси дорівнюють потребам, модель даної транспортної задачі є закритою. Наступним кроком необхідно побудувати опорний план.

Матриця тарифів із припущеними навантаженнями, а також запасами та значеннями потреби наведена у таблиці 2.17.

Таблиця 2.17

Матриця тарифів

	B1	B2	B3	B4	B5	B6	Запаси
A1	90	12	7	2	15	3	50
A2	20	90	3	5	1	30	20
A3	2	17	90	1	8	10	70
A4	7	4	15	90	20	25	40

A5	10	1	4	15	90	2	50
A6	17	7	1	5	12	90	50
Потреби	40	90	10	50	70	20	

Оскільки між відповідними точками відправлення та прибуття неможливе навантаження, то для відшукування оптимального плану значення їх маршрутів було замінено на тарифи з великим значенням (максимальним для даної матриці можна визначити $90 - 30$ помножене на 3, де 30 – найбільше серед інших навантажень).

2.7.4 Обчислення транспортної задачі

Для початку необхідно знайти перший опорний план, використовуючи, як приклад, метод найменшої вартості.

Так, опорний план із заповненими методом найменшої вартості значеннями наведено у таблиці 2.18 (нові значення вказані у скобках поряд із початковими значеннями).

Таблиця 2.18

Пошук первинного опорного плану

	B1	B2	B3	B4	B5	B6	Запаси
A1	90(20)	12	7	2	15(10)	3(20)	50
A2	20	90	3	5	1(20)	30	20
A3	2(20)	17	90	1(50)	8	10	70
A4	7	4(40)	15	90	20	25	40
A5	10	1(50)	4	15	90	2(0)	50
A6	17	7	1(10)	5	12(40)	90	50
Потреби	40	90	10	50	70	20	

Оскільки весь товар використано та потреба точок прибуття задоволена, а план відповідає системі обмежень даної транспортної задачі, такий опорний план можна вважати допустимим.

Кількість нових зайнятих клітин таблиці дорівнює 11, що також відповідає формулі $m + n - 1$ ($6 + 6 - 1 = 11$), тому опорний план є невідродженим.

Значення цільової функції для цього опорного плану дорівнює:

$$F(x) = 90 \cdot 20 + 15 \cdot 10 + 3 \cdot 20 + 1 \cdot 20 + 2 \cdot 20 + 1 \cdot 50 + 4 \cdot 40 + 1 \cdot 50 + 1 \cdot 10 + 12 \cdot 40 = 2820$$

Тепер необхідно оптимізувати даний опорний план.

Для початку перевіримо його поточну оптимальність. Знайдемо попередні потенціали u_i , v_j , за зайнятими клітинами таблиці, в яких $u_i + v_j = c_{ij}$, вважаючи, що $u_1 = 0$.

$$u_1 + v_1 = 90; 0 + v_1 = 90; v_1 = 90$$

$$u_3 + v_1 = 2; 90 + u_3 = 2; u_3 = -88$$

$$u_3 + v_4 = 1; -88 + v_4 = 1; v_4 = 89$$

$$u_1 + v_5 = 15; 0 + v_5 = 15; v_5 = 15$$

$$u_2 + v_5 = 1; 15 + u_2 = 1; u_2 = -14$$

$$u_6 + v_5 = 12; 15 + u_6 = 12; u_6 = -3$$

$$u_6 + v_3 = 1; -3 + v_3 = 1; v_3 = 4$$

$$u_1 + v_6 = 3; 0 + v_6 = 3; v_6 = 3$$

$$u_5 + v_6 = 2; 3 + u_5 = 2; u_5 = -1$$

$$u_5 + v_2 = 1; -1 + v_2 = 1; v_2 = 2$$

$$u_4 + v_2 = 4; 2 + u_4 = 4; u_4 = 2$$

Дані значення також наведені у таблиці 2.19.

Таблиця 2.19

Перевірка оптимальності опорного плану

	$v_1=90$	$v_2=2$	$v_3=4$	$v_4=89$	$v_5=15$	$v_6=3$
$u_1=0$	90(20)	12	7	2	15(10)	3(20)
$u_2=-14$	20	90	3	5	1(20)	30
$u_3=-88$	2(20)	17	90	1(50)	8	10

$u_4=2$	7	4(40)	15	90	20	25
$u_5=-1$	10	1(50)	4	15	90	2(0)
$u_6=-3$	17	7	1(10)	5	12(40)	90

Так, опорний план не є оптимальним, тому що існують оцінки вільних клітин, для яких $u_i + v_j > c_{ij}$.

$$(1;4): 0 + 89 > 2; \Delta_{14} = 0 + 89 - 2 = 87 > 0$$

$$(2;1): -14 + 90 > 20; \Delta_{21} = -14 + 90 - 20 = 56 > 0$$

$$(2;4): -14 + 89 > 5; \Delta_{24} = -14 + 89 - 5 = 70 > 0$$

$$(4;1): 2 + 90 > 7; \Delta_{41} = 2 + 90 - 7 = 85 > 0$$

$$(4;4): 2 + 89 > 90; \Delta_{44} = 2 + 89 - 90 = 1 > 0$$

$$(5;1): -1 + 90 > 10; \Delta_{51} = -1 + 90 - 10 = 79 > 0$$

$$(5;4): -1 + 89 > 15; \Delta_{54} = -1 + 89 - 15 = 73 > 0$$

$$(6;1): -3 + 90 > 17; \Delta_{61} = -3 + 90 - 17 = 70 > 0$$

$$(6;4): -3 + 89 > 5; \Delta_{64} = -3 + 89 - 5 = 81 > 0$$

Максимальну оцінку 87 має точка (1;4) зі значенням 2. Це означає, що можна провести зміну, збільшивши дане значення за рахунок (1;1) та (3;4). Дане перетворення візуально відображене у таблиці 2.20.

Таблиця 2.20

Перетворення опорного плану

	1	2	3	4	5	6	Запаси
1	90(20)(-)	12	7	2(+)	15(10)	3(20)	50
2	20	90	3	5	1(20)	30	20
3	2(20)(+)	17	90	1(50)(-)	8	10	70
4	7	4(40)	15	90	20	25	40
5	10	1(50)	4	15	90	2(0)	50
6	17	7	1(10)	5	12(40)	90	50
Потреби	40	90	10	50	70	20	

З мінусових навантажень x_{ij} необхідно вибрати найменше, тобто $y = (1;1) = 20$. Дане значення необхідно додати або відняти згідно знаків. Новий опорний план відображено у таблиці 2.21.

Таблиця 2.21

Перевірка оптимальності опорного плану

	$v_1=3$	$v_2=2$	$v_3=4$	$v_4=2$	$v_5=15$	$v_6=3$
$u_1=0$	90	12	7	2(20)	15(10)	3(20)
$u_2=-14$	20	90	3	5	1(20)	30
$u_3=-1$	2(40)	17	90	1(30)	8	10
$u_4=2$	7	4(40)	15	90	20	25
$u_5=-1$	10	1(50)	4	15	90	2(0)
$u_6=-3$	17	7	1(10)	5	12(40)	90

Даний опорний план не є оптимальним, тому що все ще існують оцінки вільних клітин, для яких $u_i + v_j > c_{ij}$.

$$(3;5): -1 + 15 > 8; \Delta_{35} = -1 + 15 - 8 = 6 > 0$$

Наступне перетворення наведено у таблиці 2.22.

Таблиця 2.22

Перетворення опорного плану

	1	2	3	4	5	6	Запаси
1	90	12	7	2(20)(+)	15(10)(-)	3(20)	50
2	20	90	3	5	1(20)	30	20
3	2(40)	17	90	1(30)(-)	8(+)	10	70
4	7	4(40)	15	90	20	25	40
5	10	1(50)	4	15	90	2(0)	50
6	17	7	1(10)	5	12(40)	90	50
Потреби	40	90	10	50	70	20	

На цей раз найменшим значенням є 10. Після перетворення отримано новий план, відображений у таблиці 2.23.

Необхідно також перевірити його оптимальність. Нижче наведено попередні потенціали u_i , v_j , за зайнятими клітинами таблиці, в яких $u_i + v_j = c_{ij}$, вважаючи, що $u_1 = 0$.

$$u_1 + v_4 = 2; 0 + v_4 = 2; v_4 = 2$$

$$u_3 + v_4 = 1; 2 + u_3 = 1; u_3 = -1$$

$$u_3 + v_1 = 2; -1 + v_1 = 2; v_1 = 3$$

$$u_3 + v_5 = 8; -1 + v_5 = 8; v_5 = 9$$

$$u_2 + v_5 = 1; 9 + u_2 = 1; u_2 = -8$$

$$u_6 + v_5 = 12; 9 + u_6 = 12; u_6 = 3$$

$$u_6 + v_3 = 1; 3 + v_3 = 1; v_3 = -2$$

$$u_1 + v_6 = 3; 0 + v_6 = 3; v_6 = 3$$

$$u_5 + v_6 = 2; 3 + u_5 = 2; u_5 = -1$$

$$u_5 + v_2 = 1; -1 + v_2 = 1; v_2 = 2$$

$$u_4 + v_2 = 4; 2 + u_4 = 4; u_4 = 2$$

Таблиця 2.23

Перевірка оптимальності опорного плану

	$v_1=3$	$v_2=2$	$v_3=-2$	$v_4=2$	$v_5=9$	$v_6=3$
$u_1=0$	90	12	7	2(30)	15	3(20)
$u_2=-8$	20	90	3	5	1(20)	30
$u_3=-1$	2(40)	17	90	1(20)	8(10)	10
$u_4=2$	7	4(40)	15	90	20	25
$u_5=-1$	10	1(50)	4	15	90	2(0)
$u_6=3$	17	7	1(10)	5	12(40)	90

Даний опорний план є оптимальним, тому що всі оцінки вільних клітин задовольняють умові $u_i + v_j \leq c_{ij}$. Мінімальні витрати становлять:

$$F(x) = 2*30 + 3*20 + 1*20 + 2*40 + 1*20 + 8*10 + 4*40 + 1*50 + 1*10 + 12*40 = 1020$$

2.7.5 Аналіз отриманих результатів

Отримані показники показують оптимальне навантаження шляхів (маршрутів) та можуть на пряму використовуватись при перевезеннях.

Отже, перше на що необхідно звернути увагу – це різницю між початковими даними (значеннями навантажень) та їх відповідними результатами. Так, згідно з різницею:

- з 1-го відділу до 4-го необхідно збільшити навантаження на 30 ум. од., а також до 6-го відділу відповідно на 20 ум. од;
- з 2-го відділу можна перенести все навантаження на 5-ий відділ;
- з 3-го відділу до 1-го навантаження підвищилось на ум. 40 од., до 4-го – на 20 ум. од., а до 5-го – на 10 ум. од.;
- з 4-го та 5-го відділів можна перенести все навантаження на 2-ий відділ;
- з 6-го відділу до 3-го можна збільшити навантаження на 10 ум. од., та до 5-го на 40 ум. од..

Друге, на що необхідно звернути увагу – дана задача з поточними вхідними даними може мати безліч оптимальних планів, оскільки оцінка для маршруту з 5-го до 6-го відділів дорівнює 0. Але, зважаючи на поступові зміни та можливий ріст даного сервісу та, як результат, збільшення загального навантаження, у майбутньому дана деталь може усунутись.

Нарешті, взявши до уваги низьку частоту перевезень (низькі умовні значення навантаження), можна припустити що одного запасного маршруту для кожних двох точок є достатнім.

Висновок

Було сформовано та обчислено транспортну задачу для визначення оптимальних маршрутів між відділеннями перевезень компанії. Так, результати показали список додаткового навантаження, що можна

розподілити між шляхами для оптимальності плану. Крім того, було виявлено можливу неточність із навантаженням з відділу 5 у відділ 6.

Далі було проведено аналіз отриманих результатів. Відповідно до нього, отримана неточність може бути усунена із збільшенням навантаження, що може статися з подальшим ростом популярності сервісу та використання перевезень.

У рамках додаткової безпеки, було також обмірковано необхідність запасних шляхів на випадок проблем з основними. На даний момент було вирішено не впроваджувати додаткові шляхи у зв'язку із відносно низьким навантаженням.

Отримані результати використовуватимуться як частина схем та інструкцій (і можливих розкладів) даної системи маршрутів для покращення бази знань серед працівників.

З приводу візуальної складності даної системи, а також системи перевезень і доставки в цілому, необхідно **схематизувати процеси** даних систем для полегшення їх розуміння.

2.8 Схематизація та побудова сценаріїв роботи служби доставки

У даному розділі буде побудовано детальні схеми систем, пов'язаних з доставкою товарів в рамках розглядуваного сервісу, а також схеми їх процесів та підпроцесів.

Буде побудовано схеми потоку даних, сценаріїв та процесів використовуючи стандарти EPC та UML, а також use-case сценарії.

2.8.1 Вибір типів схем та обґрунтування

Внутрішні процеси будь-якої організації можуть представляти собою складні та заплутані системи, що взаємодіють одна з одною в рамках ще більшої, поверхневої системи. Найчастіше клієнти такої організації не підозрюють о процесах що відбуваються при їх взаємодії з нею.

Але навіть якщо знання цих процесів не є необхідним для клієнта, це знання може бути критичним для співробітника цієї організації. Це знання може бути обмежено підсистемою, у рамках якої працює співробітник, а може охоплювати всю систему загалом.

На такий випадок полегшення розуміння цієї системи є критичним.

Одним із способів зробити систему більш зрозумілою є схематизація та декомпозиція – зображення системи у вигляді схем її підсистем та їх взаємодії однієї з іншою.

Для схематизації систем існують популярні нотації, такі як EPC та UML. Але зважаючи на те, що перша нотація є звичною блок-схемною нотацією та підходить для більш детального планування та проектування систем, нотація UML має більший вибір напрямків призначень схем. В тому слід зауважити, що дана нотація спрямована здебільшого на комунікацію із командою або замовником, а тому може потребувати більш простого вигляду або структури, яку легко зрозуміти. Саме такі схеми можна буде вважати результатом даної задачі.

2.8.2 Нотація EPC. Система доставки та її процеси

У рамках даної задачі системою високого (першого) рівню з боку користувача є система купівлі товару. Одним із кроків даної системи є система доставки - дана система також тісно взаємодіє з системами, що передбачають перевезення товару - це можуть бути як інтеграції з поштовими службами, так і транспортними компаніями, або, насамперед, з власним відділом з перевезень. Нижчі операції також наведені у вигляді тез у додатку Г.

Для побудови підійде нотація EPC [63] – вона ідеально підходить для зображення компактних схем систем та процесів.

Як приклад, розглянемо підсистему доставки товару, купленого у інтернет-маркетплейсі. За уточнення візьмемо умову, що даний товар може бути як малим за розміром (можлива доставка більшістю поштових служб), так і великим (необхідна спеціалізована служба з перевезень, у даному випадку власний підрозділ з перевезень).

Так, схема, зображена на рисунку 2.7, відображає абстрагований процес купівлі товару від його вибору до його отримання.

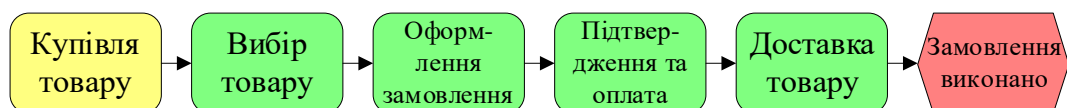


Рисунок 2.7 – Загальна модель процесу купівлі товару

Це простий лінійний процес, де кожний крок представляє собою підсистему із своїми підпроцесами. Розглянемо крок «Доставка товару» більш детально (рисунок 2.8).

Можна побачити, що дана модель також містить процеси, що можна деталізувати та зобразити у вигляді окремих підсистем. До таких процесів відносяться «Доставка власним відділенням», що описує процес доставки відділенням перевезень компанії, та «Повернення товару», що представляє собою інакший процес перевезення товару.

Треба зауважити, що процеси «Доставка Новою Поштою», «Доставка Укрпоштою» та «Доставка ІнТайм» є процесами взаємодії із окремими організаціями [54].

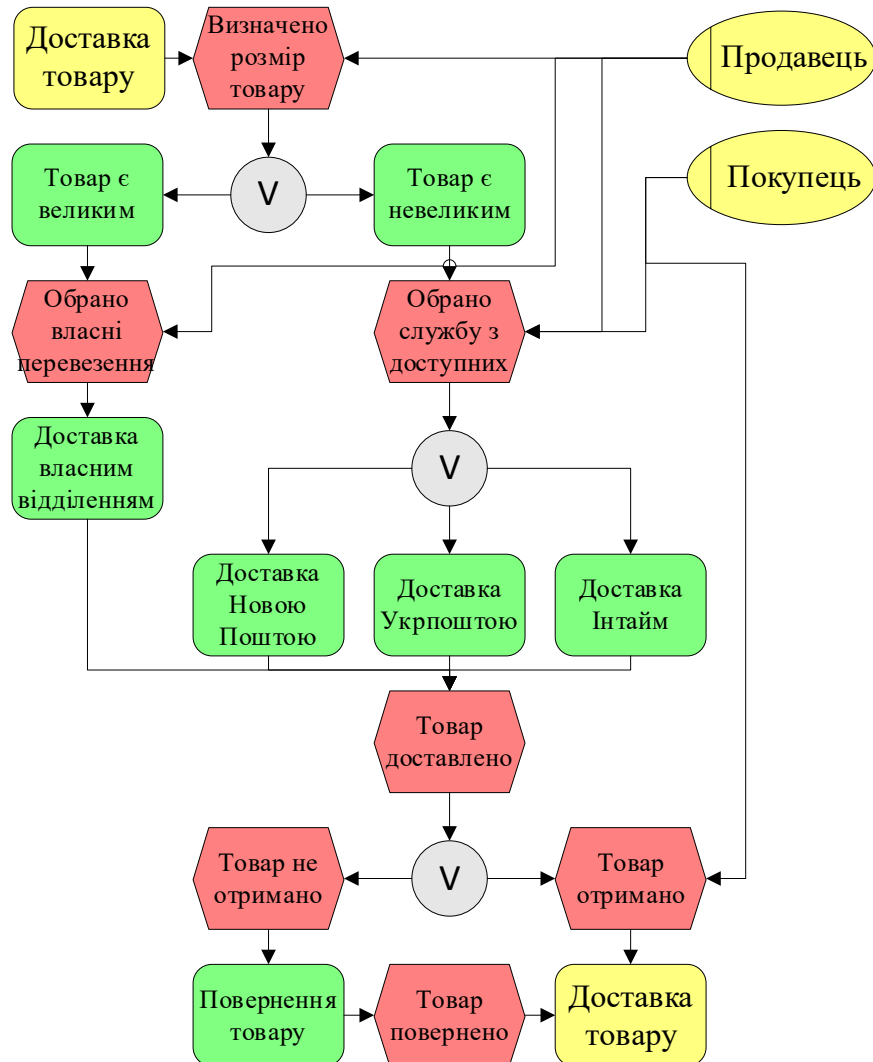


Рисунок 2.8 – Деталізація процесу «Доставка товару»

Так, розглянемо підпроцеси «Доставка власним відділенням» та «Повернення товару» більш детально. Вони представляють собою більш прості, але не атомарні процеси, серед яких також можна виділити свої підпроцеси. Схеми даних підпроцесів зображені на рисунках 2.9-2.10.

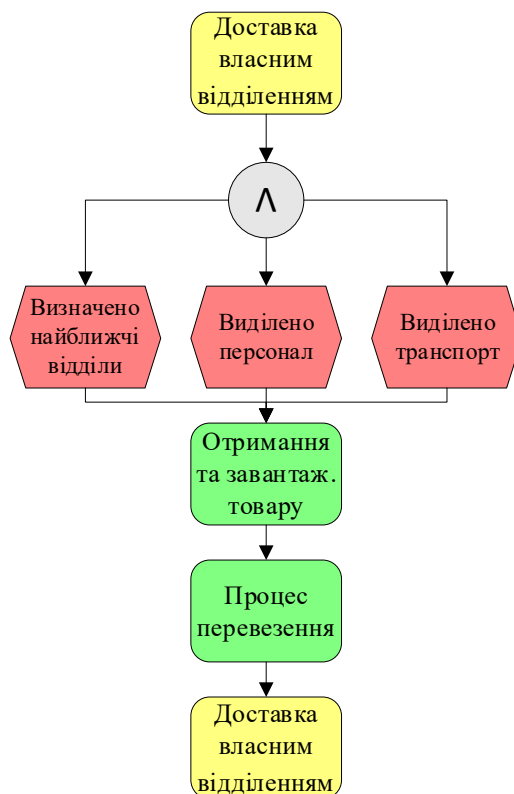


Рисунок 2.9 – Деталізація процесу «Доставка власним відділенням»

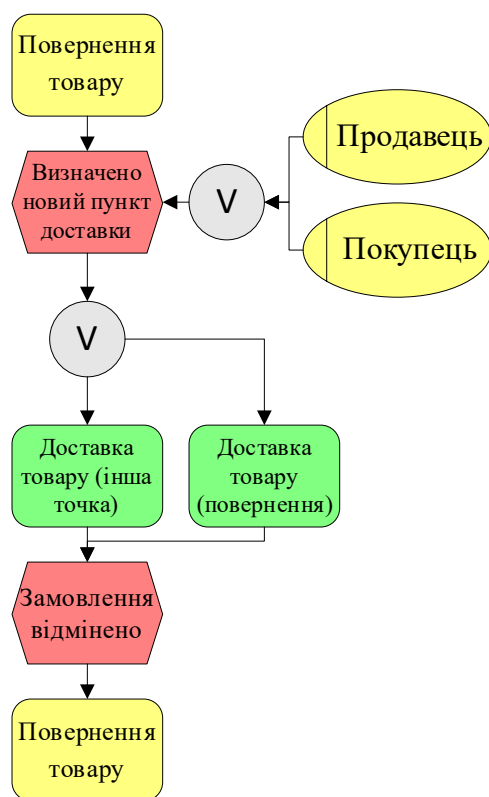


Рисунок 2.10 – Деталізація процесу «Повернення товару»

Можна помітити, що схема доставки має ще один підпроцес, що може потребувати декомпозиції – «Отримання та завантаження товару».

Крім того, процес повернення має рекурсивні процеси як його частину: процеси «Доставка товару (інша точка)» та «Доставка товару (повернення)» посилаються на процес «Доставка товару», що і містить даний процес.

Також можна виділити, що на нову точку прибуття товару за схему може вплинути або продавець (при поверненні), або покупець.

2.8.3 Нотація UML. Випадки використання процесів доставки

Для спрощення візуального сприйняття побудованих EPC схем, можна виокремити окремі їхні частини та представити їх у вигляді діаграм випадків використання за нотацією UML. Треба зауважити, що дані сценарії є спрощеними для візуалізації та можуть не відображати всі деталі та випадки окремої ситуації.

Для початку необхідно розглянути дві базові частини купівлі товару, що оточують його доставку:

- замовлення товару;
- отримання товару.

Дані випадки використання веб-додатку є єдиними випадками, пов'язаними із системою доставки. Замовлення товару є вхідним випадком, результатом якого є успішно доставлений товар, отримання товару ж є вихідним випадком, результатом якого вважається завершення замовлення.

Діаграма випадків використання замовлення товару зображена на рисунку 2.11, використання отримання – на рисунку 2.12.

Можна помітити, що наведені діаграми мають схожі групи акторів – користувачів веб-додатку, які у даному випадку є головними учасниками замовлення, а також відділ перевезень, що залучається при виборі відповідної опції при оформленні замовлення, та, нарешті, адміністрація сайту, що відповідає за оновлення статусу замовлення. За адміністрацію сайту також може вважатися система, що може провести певні дії автоматично.

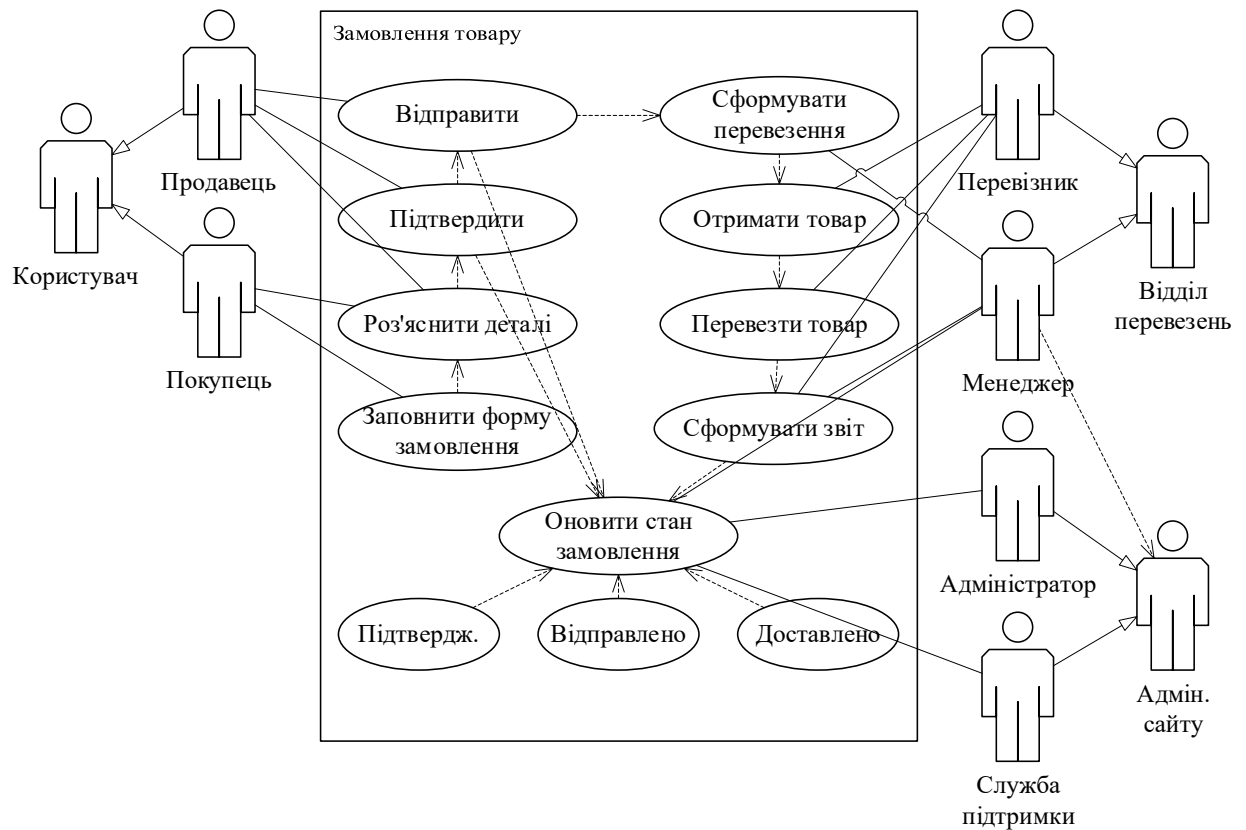


Рисунок 2.11 – Діаграма випадків використання замовлення товару

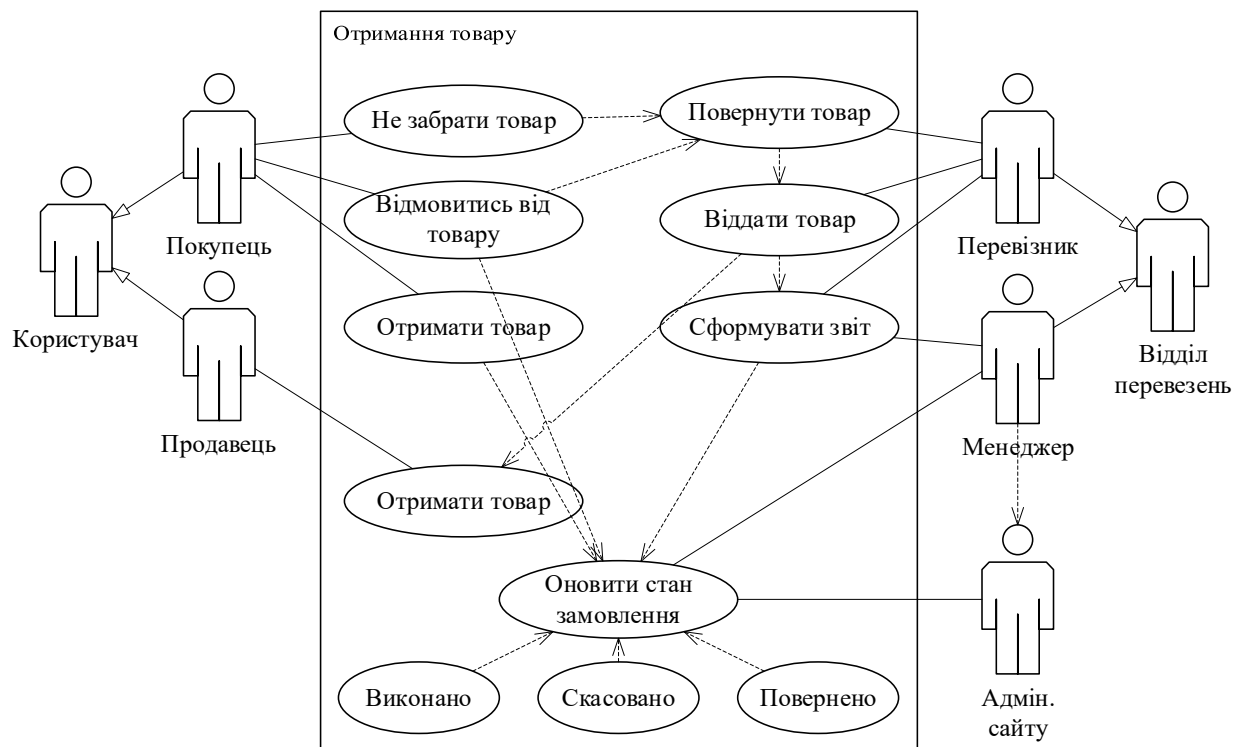


Рисунок 2.12 – Діаграма випадків використання отримання товару

2.8.4 Нотація UML. Діаграми діяльності процесів доставки

Щоб у простому вигляді відобразити послідовність подій у випадках використання, описаних у попередньому розділі, на основі даних послідовностей можна побудувати діаграми діяльності.

Так, на рисунку 2.13 відображена діяльність системи у випадку замовлення товару, а на рисунку 2.14 – у випадку отримання або повернення.



Рисунок 2.13 – Діаграма діяльності замовлення товару

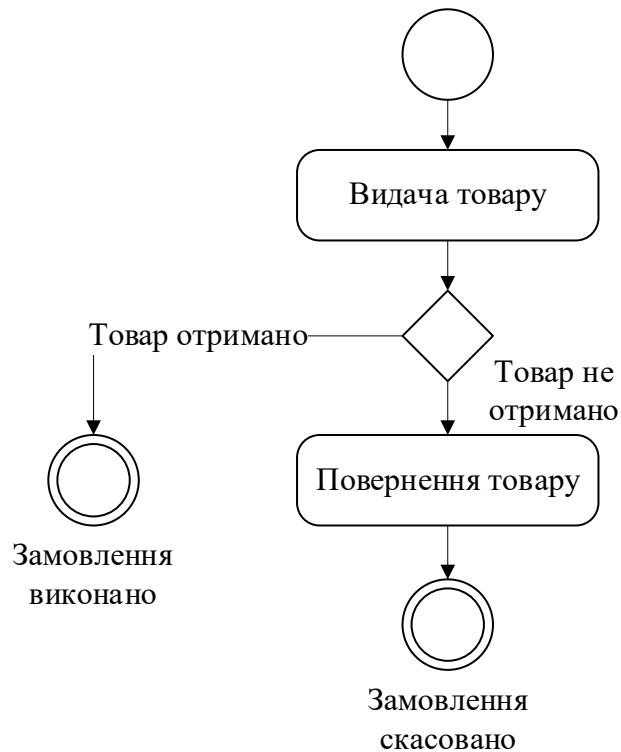


Рисунок 2.14 – Діаграма діяльності отримання товару

Можна побачити, що більшість дій та операцій наведених у діаграмах 2.13 та 2.14 відповідає діям акторів та їх залежностям у діаграмах 2.11 та 2.12 відповідно.

2.8.5 Use-Case Сценарії замовлення через веб-додаток

Детальніше за все процеси взаємодії користувача із системою доставки можна описати у вигляді сценаріїв використання, що включають всі процеси, які відбуваються у веб-додатку в рамках купівлі товару.

Рівень деталізації має бути обґрунтований виходячи з потреб реалізації проекту. Так, деталі щодо оплати, а також випадки крайньої рідкості, що не передбачаються для використання, можна опустити.

Насамперед, внутрішній менеджмент перевезень та процеси переміщення транспорту від точки до точки було опущено для дотримання основної суті процесу та сфокусуватись на взаємодії користувачів з системою.

Побудовані сценарії використання приведені нижче.

BP-01 – Купівля товару (процес верхнього рівня)

Актор: Користувач (покупець).

Основний потік

1. Актор вирішує купити певний товар.
2. Актор відкриває веб-додаток.
3. Актор переходить у каталог товарів необхідної категорії.
4. Актор знаходить бажаний товар.
5. Актор оформляє замовлення (BP-02).
6. Продавець підтверджує замовлення (BP-03).
7. Товар прямує до точки доставки (BP-04).
8. Товар доставлено.
9. Автор отримує товар.
10. Замовлення виконано успішно.

Розширення

- 5а. Замовлення не оформлено.
 - 5а1. Перехід к п. 10а.
- 6а. Замовлення відхилено (не підтверджено).
 - 6а1. Система відправляє відповідь актору.
 - 6а2. Перехід к п. 10а.
- 8а. Товар не доставлено (не відправлено).
 - 8а1. Система нотуфікує актора.
 - 8а2. Перехід к п. 10а.
- 9а. Актор назначає нову точку доставки.
 - 9а1. Перехід к п. 7.
- 9б. Актор скасовує замовлення.
 - 9б1. Актор повертає товар.
 - 9б2. Товар прямує до точки відправки (BP-04).
 - 9б3. Продавець отримує товар.
 - 9б4. Перехід к п. 10а.

9б3а. Продавець не отримує товар.

9б3а1. Долучається служба з непередбачених випадків.

9б3а2. Перехід к п. 9б4.

9в. Актор ігнорує товар.

9в1. Проходить умовний час видачі або збереження товару.

9в2. Перехід к п. 9б2.

10а. Замовлення скасовано або не виконано.

10а1. Замовлення оновлює статус на сайті.

10а2. Актору повертаються гроші.

ВР-02 – Оформлення замовлення

Актор: Користувач (покупець).

Основний потік

1. Товар доступний або в наявності.
2. Актор заповнює форму.
3. Форма перевіряється системою.
4. Форма заповнена правильно.
5. Замовлення успішно оформлене.

Розширення

1а. Товар недоступний.

1а1. Купівля неможлива. Актора повертає на сторінку товару.

4а. Форма заповнена неправильно.

4а1. Система виводить помилки заповнення.

4а2. Перехід к п. 2.

ВР-03 – Підтвердження замовлення

Актор: Користувач (продавець).

Основний потік

1. Система відправляє повідомлення про замовлення актору.
2. Актор реагує на повідомлення.

3. Актор перевіряє стан товару.
4. Товар в наявності та готовий до відправки.
5. Актор підтверджує замовлення.
6. Замовлення підтвержене.

Розширення

- 3а. Власник не відреагував на повідомлення.
 - 3а1. Система чекає умовний відрізок часу.
 - 3а2. Перехід к п. 6а.
- 4а. Товар неможливо відправити.
 - 4а1. Перехід к п. 6а.
- 5а. Актор не підтверджує замовлення.
 - 5а1. Перехід к п. 6а.
- 6а. Замовлення скасовано або не підтверджено.
 - 6а1. Купівля неможлива.

ВР-04 – Перевезення товару

Актор: Обрана служба перевезень.

Основний потік

1. Обрано відділ перевезень компанії.
2. Актор формує оптимальний маршрут.
3. Актор визначає наявність доступних ресурсів.
4. Наявні необхідні ресурси (обладнання, персонал).
5. Актор формує перевезення.
6. Актор прямує на точку відправки.
7. Актор отримує товар від джерела (продавця).
8. Актор перевозить товар на точку прибуття (доставки).
9. Товар доставлено.
10. Актор формує звіт.

Розширення

- 1а. Обрано поштову службу.

1a1. Актор виконує організаційні функції перевезення, обумовлені даною службою.

1a2. Перехід к п. 6.

1б. Обрано метод доставки, не інтегрований з веб-додатком.

1б1. Продавець та покупець взаємодіють незалежно від системи.

1б2. Перехід к п. 9.

4а. Наявних ресурсів не вистачає.

4а1. Актор звітує про проблему.

4а2. Служба підтримки зв'язується із учасниками перевезення для визначення наступних дій.

7а. Актор не отримує товар від джерела.

7а1. Актор звітує про проблему.

7а2. Система нотифікує отримувача щодо ситуації.

7а3. Купівля неможлива.

У випадку, коли купівля неможлива, вона вважається скасованою.

Висновок

Після проведення схематизації та декомпозиції процесу доставки товару, було описано основний процес доставки товару та його підпроцеси, використовуючи нотацію EPC. У ході опису було визначено:

- системи інших організацій, з якими відбувається взаємодія;
- окремі процеси, що потребують подальшої деталізації;
- окремі рекурсивні посилання на процеси.

Також було детально розглянуто, і, як результат, декомпозовано та візуально описано процеси замовлення та отримання товару у вигляді діаграм випадків використання та діаграм діяльності.

Нарешті, було виконано опис всіх бізнес-процесів, пов'язаних із взаємодією користувача із доставкою товару, у формі use case сценаріїв. Результатом опису є сценарії ВР-01, ВР-02, ВР-03 та ВР-04, наведені вище.

ВИСНОВКИ

У даній роботі було проведено проектування БД та реалізацію бек-енду, веб-серверу та стратегії розгортання веб-додатку, вибору серверу для нього, а також стратегії управління цим сервером. Було проаналізовано та обрано методи визначення рейтингу та прогнозування статистики товарів на сайті. На основі проведених робіт було розроблено та презентовано бізнес-проект з додання системи доставок, визначення альтернатив досягнення його мети, а також оптимізовано маршрути нового відділу перевезень. Нарешті, було проведено детальну схематизацію всіх процесів кінцевої системи доставок.

Під час моделювання бази даних було побудовано та оптимізовано ER-структуру і повну реляційну схему, список всіх полів та зв'язків, SQL-запит для створення всієї структури та вспоміжні SQL-запити.

Проектування бек-енду і веб-серверу, а також опис стратегії розгортання відобразили готовність веб-додатку до запуску у реальному середовищі. Аналіз обраних технологій допоміг підвищити швидкість розробки проекту.

Вибір алгоритмів формування рейтингу товарів на сайті і порівняння методів прогнозування їх статистики дозволили реалізувати поглиблену роботу з даними сайту, що також покращить роботу системи доставки.

Вибір серверу серед 56 опцій та вибір стратегії його управління дозволив звернути увагу на невідповідність обраного серверу грошовим обмеженням, що призвело до повторного вибору серверу.

Презентація бізнес-проекту дозволила виділити альтернативи досягнення його мети та вибрати необхідні, а також визначити обсяги подальших робіт.

Оптимізація маршрутів дозволив зменшити навантаження на нові відділи перевезень. Аналіз результатів звернув увагу на можливі зміни у майбутньому.

Аналіз результуючої системи доставки дозволив побудувати схеми цієї системи, що значно полегшило розуміння її структури.

З урахуванням успішності кожної частини проведеної роботи, можна сказати, що в ході даної роботи було задоволено мету дослідження та виконано всі поставлені задачі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що Таке Маркетплейс – <https://horoshop.ua/blog/chtotakoe-marketpleys/>
2. Методичні рекомендації до виконання кваліфікаційної роботи магістра студентами галузі знань 12 Інформаційні технології спеціальності 124 Системний аналіз / Т. А. Желдак, Т.В. Хом'як; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». Дніпро: НТУ «ДП», 2021. – 32 с.
3. Що Таке Системний Аналіз – Кафедра інформаційних систем та мереж, Інститут комп'ютерних наук та інформаційних технологій (ІКНІ), НУ «Львівська політехніка»: <https://ba-ds.lviv.ua/system-analysis/>
4. Методичний посібник з дисципліни “Системний аналіз”, Дивак М.П.: <http://dspace.wunu.edu.ua/jspui/bitstream/316497/627/1/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%B8%D0%B9%20%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7.pdf>
5. Моделі й методи прийняття рішень, Ус С.А., Коряшкіна Л.С., Державний ВНЗ «НГУ», 2014: [https://sau.nmu.org.ua/ua/osvita/metod/Models_and_decision-making_techniques\(Us_Koryashkina\)_NMU_SAU.pdf](https://sau.nmu.org.ua/ua/osvita/metod/Models_and_decision-making_techniques(Us_Koryashkina)_NMU_SAU.pdf)
6. Прийняття Оптимальних Рішень Методом Навчання З Нечіткою Логікою, Кравець П. О., Проданюк О. М., НУ “Львівська політехніка”, 2009: <http://science.lpnu.ua/sites/default/files/journal-paper/2019/apr/16222/vis653ism-129-147.pdf>
7. Метод аналізу ієрархій. Методичні вказівки – 137-139 с.
8. Ус С.А. Прийняття рішень в умовах ризику. Метод. вказівки – 7-11 с.
9. Прогнозування – Академічний Тлумачний Словник української мови: в 11 томах. – Том 8, 1977. – Стор. 153.: <http://sum.in.ua/s/proghnozuvaty>

10. Прогнозування юридичне – Юридична енциклопедія, Шемшученко Ю. С. Українська енциклопедія ім. М. П. Бажана, 2003. – Т. 5 : П – С. – 736 с. – ISBN 966-7492-05-2.: <https://cyclop.com.ua/content/view/1273/58/1/18/#25327>
11. Що таке база даних, Герасимик І., Кафедра АПЕПС ТЕФ КПІ ім. І. Сікорського: <http://apeps.kpi.ua/shco-take-basa-danykh>
12. SQL – Вікіпідручник: <https://uk.wikibooks.org/wiki/SQL>
13. Основи нормалізації бази даних: <https://ukr.4meahc.com/database-normalization-basics-60451>
14. База даних PostgreSQL – офіційний сайт: <https://www.postgresql.org/>
15. Мова програмування Python – офіційний сайт: <https://www.python.org/>
16. Фреймворк Django – офіційний сайт: <https://www.djangoproject.com/>
17. Бібліотека Poetry – офіційний сайт: <https://python-poetry.org/>
18. Стил-гайд PEP: <https://www.python.org/dev/peps/pep-0008/>
19. Бібліотека Flake8 – документація: <https://flake8.pycqa.org/en/latest/>
20. Веб-сервер Nginx – офіційний сайт: <https://www.nginx.com/>
21. WSGI-сервер Gunicorn – офіційний сайт: <https://gunicorn.org/>
22. Технологія Docker – офіційний сайт: <https://www.docker.com/>
23. Бібліотека Docker-compose: <https://docs.docker.com/compose/>
24. Система контролю версій Git – офіційний сайт: <https://git-scm.com/>
25. Мова конфігурації Bash: <https://www.gnu.org/software/bash/>
26. Мова конфігурації Yaml – офіційний сайт: <https://yaml.org/>
27. Мова конфігурації Toml – офіційний сайт: <https://toml.io/en/>
28. Мова Json – офіційний сайт: <https://www.json.org/json-en.html>
29. Сервіс DigitalOcean – офіційний сайт: <https://www.digitalocean.com/>
30. Сервіс GitHub – офіційний сайт: <https://github.com/>
31. Переваги та недоліки Django: <https://python-scripts.com/django-obzor>
32. Виконання транзакцій у PostgreSQL:
<https://www.postgresql.org/files/developer/transactions.pdf>
33. PostgreSQL та MVCC:

<https://devcenter.heroku.com/articles/postgresql-concurrency>

34. Структура проектів Django – документація:

<https://django-project-skeleton.readthedocs.io/en/latest/structure.html>

35. Додатки Django – документація:

<https://docs.djangoproject.com/en/3.2/ref/applications/>

36. Моделі Django – документація:

<https://docs.djangoproject.com/en/3.2/topics/db/models/>

37. Представлення Django – документація:

<https://docs.djangoproject.com/en/3.2/topics/http/views/>

38. HTTP-шаблони Django – документація:

<https://docs.djangoproject.com/en/3.2/topics/templates/>

39. Проміжні шари Django – документація:

<https://docs.djangoproject.com/en/3.2/topics/http/middleware/>

40. Словник Economy-Pedia.com: Бізнес-проект. URL: <https://uk.economy-pedia.com/11039215-business-project>

41. Бізнес-планування підприємницької діяльності: навч. посіб. / З.С. Варналій, Т.Г.Васильців, Р.Л.Лупак, Р.Р.Білик. Чернівці: Технодрук, 2019. 8 с.

42. Бізнес-планування: навч. посіб. / Т. Г. Васильців, Я. Д. Качмарик, В. І. Блонська, Р. Л. Лупак. – К. : Знання, 2013. – 8 с.

43. Теорія ігор в дослідженні конфліктних ситуацій / В.М. Рева, О.П. Купенко. – Д.: Національний гірничий університет, 2011. – 4 с.

44. Системний Аналіз Пошуку Оптимальних Рішень В Економічних Конфліктах / О. І. Щедріна, І. В. Череда. – Д.: Київський національний економічний університет імені Вадима Гетьмана, 2019. – 243 с.

45. Теорія ігор в дослідженні конфліктних ситуацій / В.М. Рева, О.П. Купенко. – Д.: Національний гірничий університет, 2011. – 6-7 с.

46. Біматричні ігри. Рівновага Неша. Оптимальність Парето. URL: <http://um.co.ua/8/8-9/8-99301.html>

47. Транспортна задача: навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення» / КПІ ім. Ігоря Сікорського; уклад.: О. К. Молодід. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 5 с.

48. Практикум за курсом «Методи оптимізації та дослідження операцій». Частина I. Дослідження операцій: навч. посіб. / Л.С. Коряшкіна, С.А. Ус / М-во освіти і науки України; Дніпро: НТУ «ДП», 2020. – 53-54 с.

49. Основи UML. Про UML – Документація KDE. URL: <https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml-basics.html>

50. UML для бізнес-моделювання: для чого потрібні діаграми процесів. URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html>

51. Класифікація моделей. URL:

https://pidru4niki.com/91294/ekologiya/klasifikatsiya_modeley

52. Парфененко Ю.В., Шендрик В.В., Красніков С.І. Концептуальна модель інформаційної системи аналізу теплозабезпечення. Електронний ресурс – 137-138 с. <https://core.ac.uk/download/pdf/14063185.pdf>

53. Математичні моделі: означення, характеристики. URL: https://web.posibnyky.vntu.edu.ua/firen/3zlepko_osnovy_biomedychnogo_radioel_ektronного_aparatobuduvannya/6.html

54. Популярні способи доставки товарів з Інтернет-магазинів. URL: <https://neoseo.com.ua/uk/sposoby-dostavki-internet-magazina>

55. Аналіз Ринку Експрес-Доставки В Україні, с. 7 – Олексій Тараненко, 2020 р. URL: <https://www.econa.org.ua/index.php/econa>

56. Оплата та доставка – Prom.ua URL: <https://help.prom.ua/hc/uk/sections/360002436058-Загальне-по-доставці>

57. Як це працює – Shafa.ua URL: <https://shafa.kayako.com/ru/category/1-jak-se-pracjuje>

58. Доставка «день у день» і покриття по всій країні: топ поштових операторів України URL: <https://thepage.ua/ua/business/top-5-ukrayinskih-poshtovij-operatoriv-rejting-na-den-poshti>

59. Огляд українських служб доставки – Хорошоп URL: <https://horoshop.ua/blog/obzor-ukrainskikh-sluzhb-dostavki/>
60. Нова Пошта API URL: <https://developers.novaposhta.ua/>
61. Укрпошта API URL: <https://dev.ukrposhta.ua/>
62. Інтеграція ІнТайм URL: <https://apix-drive.com/ua/intime>
63. Нотація EPC – КН-груп. Управління проектами URL: <https://kn-grup.com/publications/glossary/257-event-driven-process-chain>

ДОДАТКИ

Додаток А

Відомість матеріалів кваліфікаційної роботи

№ з/п	Позначення				Назва	Кількість	Примітки		
1									
2					Документація				
3									
4	САУ.КР.22.04.ПЗ				Пояснювальна записка	102	Формат А4		
5									
6	САУ.КР.22.04.ДМ				Демонстраційні матеріали		Презентація на CD-R		
7									
8	САУ.КР.22.04.КР				Копія роботи	1	Диск CD-R		
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
					САУ.КР.22.04.ДА.ПЗ.				
Змін.	Аркуш	№ докум.	Підпис	Дата					
Розроб.		Владимиров Я.Д.			Матеріали кваліфікаційної роботи	Літ.	Аркуш	Аркушів	
Керівн.		Хом'як Т.В.							
Керівн. Сп. Р.		Хом'як Т.В.				НТУ «ДП», 12; 124м-21			
Н.контр.		Хом'як Т.В.							
Зав. каф.		Желдак Т.А.							

Відгук на кваліфікаційну роботу магістра
студента групи СА-124м-21-1 Владимирова Якіма Дмитровича
спеціальності Системний аналіз

Тема кваліфікаційної роботи: Аналіз способів перевезень товарів меблевого маркетплейсу.

Обсяг кваліфікаційної роботи 102 с., 19 рисунків, 23 таблиці, 4 додатки, 63 джерела.

Мета кваліфікаційної роботи: аналіз транспортних систем та поштових служб з метою реалізації інтернет-маркетплейсу та системи доставок, вибору серверу на основі багатьох критеріїв та стратегії його управління, а також вибору методу прогнозування статистики та оптимізації маршрутів перевезень.

Актуальність теми обумовлена тим, що у сучасних умовах розвитку принципи побудови управління електронно комерційною діяльністю роздрібних мереж значно відрізняються від фізичних типів торгівлі специфікою управління і вимагають власних адекватних стратегій розвитку.

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності магістра спеціальності "Системний аналіз", оскільки включає всі необхідні етапи системного дослідження, а саме: аналіз об'єкта дослідження, розробка програмного забезпечення та побудову відповідних моделей і алгоритмів і впровадження їх у практичній діяльності.

Виконані в кваліфікаційній роботі завдання відповідають вимогам до професійної діяльності фахівця освітньо-кваліфікаційного рівня магістра.

Оригінальність наукових рішень полягає у розробці функціональної моделі і її декомпозиції для проведення системного аналізу, а саме розробці програмного забезпечення з метою автоматизації електронної комерції меблевої промисловості.

Практичне значення отриманих у роботі результатів полягає в оптимізації та автоматизації процесу продажу товарів меблевої промисловості, а також розширення його функціоналу, що призведе до збільшення масштабів компанії та популярності (і прибутковості) сервісу.

Висновки підтверджують можливість використання результатів роботи в умовах електронно комерційної діяльності, за умови ідентифікації потрібних параметрів.

Оформлення пояснювальної записки та демонстраційного матеріалу до неї виконано згідно з вимогами. Роботу виконано самостійно, відповідно до завдання та у повному обсязі.

В процесі роботи Владимирів Я.Д. показав здатність до самостійного вирішування задач системного аналізу, високий рівень підготовки. Зауважень до роботи немає.

Кваліфікаційна робота в цілому заслуговує на оцінку: відмінно, а її автор заслуговує присвоєння кваліфікації “магістр з системного аналізу”.

Керівник кваліфікаційної роботи магістра,

К.ф.-м.н., доцент

Хом'як Т.В.

Приведені у роботі тези:

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ЦІНОУТВОРЕННЯ

Владимиров Я. Д., Козир С. В.

Національний технічний університет
«Дніпровська політехніка», Дніпро, Україна

Вступ. Оптимальна ціна для товарів кожного магазину розраховується окремо, так як у кожного магазину свій розмір, асортимент, контингент покупців, споживчий попит. Класичний підхід до цінової еластичності використовує тільки два показники: зміну обсягу попиту та зміну ціни товару, але не враховує сезонність та інші фактори. [1].

Постановка проблеми. Побудувати модель формування ціни товару в певному магазині. Вихідними даними даної моделі є ціна магазину на товар. Випадковими вхідними даними є частковий попит на товар, ціна товару у постачальника магазину та курс валюти. Невипадковими вхідними даними є НДС магазину на товари та сезонність товару. Як саме ціноутворення залежить від вказаних вхідних даних: чим вище попит на товари, тим вище ціни цих товарів; збільшення ціни на товар негативно відображається на попиті; ціна товарів у постачальників магазину та курс валюти прямо пропорційні цінам товарів у магазині; НДС є відсотковим параметром, що буде підвищувати ціни. Для спрощення експериментів з даною моделлю НДС буде завжди вважатися 10%, а всі товари будуть вважатися зимовими. (ціна буде збільшуватися в два рази взимку).

Вибір та обґрунтування методу розв'язування. Дана модель має нелінійну поведінку, причинно-наслідкові зв'язки, невизначеність і велику кількість параметрів, що дозволяє імітувати її за допомогою методу імітаційного моделювання. Застосування імітаційної моделі є доцільним для перевірки плановірності припущень, а також для демонстрації кінцевих результатів дослідження для досить повної моделі реальної ситуації. [2].

До уваги взяти наступні недоліки імітаційної моделі:

1. Імітаційна модель не є точною, і ми не в змозі виміряти ступінь цієї неточності. Цей недолік може бути подолане лише частково шляхом аналізу чутливості моделі до зміни певних параметрів [3].

2. Складність оптимізації системи (багатоваріантність розрахунків при наявності імовірнісних перешкод);

3. Ймовірнісна оцінка похибки.

Одиницями модельного часу взято дні. В рамках моделі сезон йде протягом 90 днів кожного року (365 днів). При ініціалізації експерименту сезон починається одразу. Курс валюти визначається на основі випадкового значення за кожний день, що генерується моделлю в діапазоні 20 – 30 грн за долар. Для визначення коефіцієнту, що впливатиме на результуючу ціну, поточний курс ділиться на середній. Для спрощення моделі, ціна товару у постачальника не залежить від курсу валюти.

Формула результуючої ціни:

(Ціна у постачальника * (1.0 + НДС) * Попит на товар * (Курс / Середній курс (за останні 3 місяці))).

Реалізація обраного методу. Вигляд процесу (без схованих агентів), а також результати імітації зображено на рисунку 1.

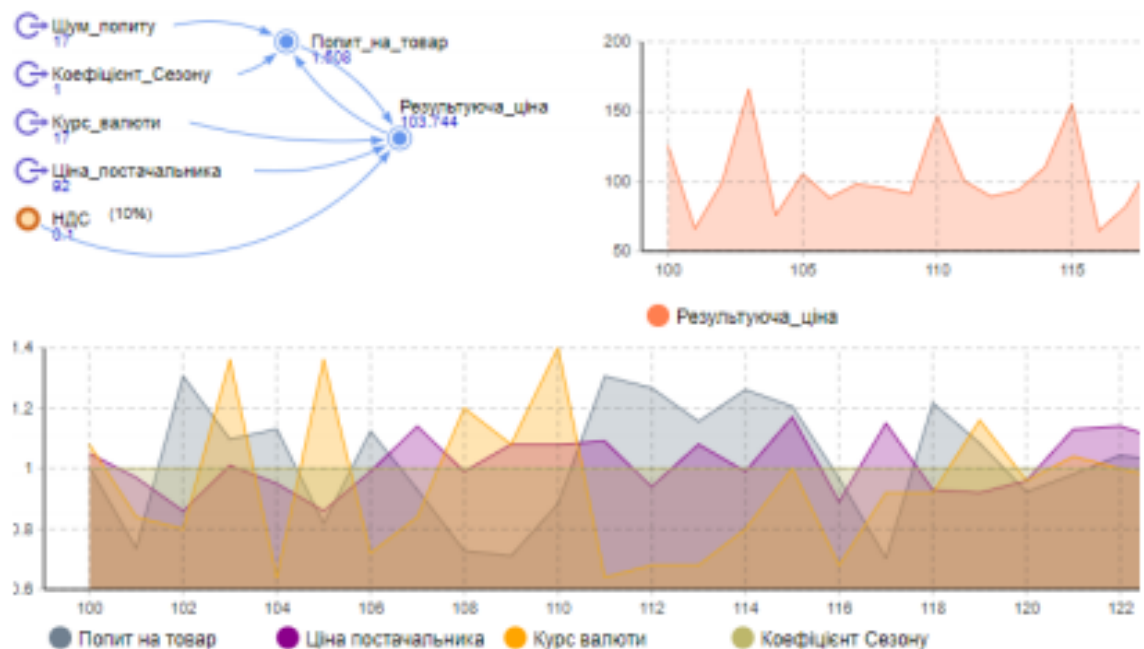


Рисунок 1 – Імітаційне моделювання процесу ціноутворення в середовищі AnyLogic

Висновки. Проведені експерименти показали, що:

- коефіцієнт сезону є найбільш значимим коефіцієнтом у даній моделі (якщо активний);
- результуюча ціна має найменшу залежність від курсу валюти (для більш точного результату треба проводити експеримент більших масштабів, де враховувати кризи та інфляцію валюти).

Список літературних джерел

1. Мікроекономіка. Еластичність попиту та пропозиції [Електронний ресурс]. – Режим доступу: <http://gx3000.ru/mikroekonomika/elasticnist-popitu-ta-propozicii.html>.
2. Куприяшкин, А.Г. Основы моделирования систем: учеб. пособие / А.Г. Куприяшкин; Норильский индустр. ин-т. –Норильск: НИИ, 2015. – 135 с.
3. Імітаційне моделювання [Електронний ресурс] – Режим доступу: <http://ukr.vipreshebnik.ru/entsiklopediya/55-i/3709-imitatsijne-modelyuvannya.html>

ВИБІР ТА УПРАВЛІННЯ СЕРВЕРНИМ ОБЛАДНАННЯМ ДЛЯ РОБОТИ З ВЕБ-ДОДАТКОМ

При розробці веб-додатку виникає задача вибору серверу та стратегії управління з подальшим аналізом результату. Мета таких обчислень полягає у оптимізації витрат на ресурси, необхідні для роботи додатку [1].

Змістова модель. Розробити математичну модель, яка дозволяє описати вибір оптимального серверу та стратегії управління цим сервером на основі багатьох критеріїв. Модель повинна:

1. враховувати часове обмеження, грошове обмеження (виділений капітал) та прогнозоване навантаження на сервер;
2. враховувати всі параметри та можливі сценарії поведінки серверу, а також всі види серверів, представлені стороннім сервісом.

Вихідні дані:

1. типи серверів та їх кількість;
2. стратегія управління сервером (-ами);
3. залишки капіталу.

Концептуальна модель:

Дано:

1. прайс-лист Digital Ocean (список серверів та їх характеристик);
2. виділений капітал (максимальна грошова сума, виділена на вибір та обслуговування серверів) = 10000 грн;
3. часове обмеження (час, на протязі якого планується витратити виділений капітал) = 1 рік;
4. приблизний спрогнозований потік клієнтів (приблизне навантаження на сервери) ≈ 1000 звертань до серверу на день.

Необхідно:

1. обрати оптимальний сервер, враховуючи виділений капітал та приблизний потік клієнтів. При виборі серверів ціна повинна наближатись до мінімуму, а загальна оцінка – до максимуму;
2. визначити оптимальний план обслуговування серверів (обрати найоптимальнішу стратегію), враховуючи залишки капіталу та часові обмеження;
3. у випадку нестатку капіталу знову провести вибір серверу.

¹ студент групи 124-18ск-1, НТУ «Дніпровська політехніка»

² доцент кафедри САіУ, НТУ «Дніпровська політехніка», к. ф.-м. н.

Задача 1: Необхідно звернути увагу на вхідний список серверів. Проблема такого списку полягає у тому, що завелика кількість опцій може сильно ускладнити процес застосування методу аналізу ієрархій.

Вибір серверу відбувається з урахуванням однорічного плану, в рамках якого на роботу з сервером виділяється 10000 грн, що приблизно дорівнює 500 доларам. Звідси максимальна ціна серверу = $500/12 = 41,65$ доларів. Така сума знижує кількість доступних опцій від 56 до 6.

Тепер треба відзначити важливі та неважливі характеристики. Так, характеристики “\$ в годину” та “\$ на місяць” відображають однакову інформацію, це означає, що одну з них можна не враховувати.

Іншим неважливим показником можна вважати варіант SSD.

Фінальна вибірка даних зображена у таблиці 1.

Таблиця 1 – Вхідні дані задачі

Назва	Пам'ять	vCPU	Потік	Об'єм	\$ на місяць
Опція 1	1GB	1vCPU	1TB	25GB	\$5
Опція 2	2GB	1vCPU	2TB	50GB	\$10
Опція 3	2GB	2vCPU	3TB	60GB	\$15
Опція 4	4GB	2vCPU	4TB	80GB	\$20
Опція 5	8GB	4vCPU	5TB	160GB	\$40
Опція 6	4GB	2vCPU	4TB	25GB	\$40

Результат використання методу ієрархій представлено в таблиці 2. Більшість значень в ході використання методу визначені експертним шляхом на основі порівняння загальної потужності серверів.

Таблиця 2 – Глобальні пріоритети

Альтернативи	Критерії					Глобальні пріоритети
	Пам'ять	vCPU	Потік	Об'єм	\$ на місяць	
	0,058226025	0,036531731	0,14022135	0,331185726	0,433835169	
Опція 1	0,039764731	0,064214358	0,033402287	0,043939618	0,425724049	0,208591157
Опція 2	0,099334267	0,064214358	0,091168437	0,157978713	0,198952612	0,159546397
Опція 3	0,099334267	0,166488737	0,144720873	0,163964656	0,162444127	0,156935646
Опція 4	0,19141564	0,166488737	0,224673842	0,255747064	0,093981541	0,174203837
Опція 5	0,378735454	0,351716523	0,28136072	0,33443033	0,073989089	0,217211473
Опція 6	0,19141564	0,186877288	0,224673842	0,043939618	0,044908582	0,083511489

Таким чином, оптимальним вибором є опція 5. Слід також відмітити, що другим по оптимальності вибором є опція 1.

Задача 2: Модель представляє собою набір станів серверу та експертних стратегій поведінки у даних ситуаціях.

Можливі стани серверу:

- θ_1 – навантаження є нормальним (очікуваним);
- θ_2 – навантаження занизьке;
- θ_3 – навантаження завелике;
- θ_4 – сервер не працює.

Можливі стратегії:

- f_1 – замінити на сервер з іншою потужністю;
- f_2 – нічого не робити;
- f_3 – розділити навантаження;
- f_4 – перезавантажити сервер та провести технічні роботи.

Числові значення рішень мають додатне значення у діапазоні від 0 до 10 та визначені експертним шляхом в залежності від того, наскільки логічним є прийняття певної дії у певній ситуації.

Наприклад, нічого не робити є більш логічним рішенням, якщо сервер працює нормально, ніж якщо він не працює.

Всього сума рішень для кожного стану дорівнює 10. Всі числові значення зображені у таблиці 3.

Таблиця 3 – Математична модель другої задачі

	f_1	f_2	f_3	f_4
θ_1	1	8	1	0
θ_2	2	5	0	3
θ_3	3	2	4	1
θ_4	3	0	1	6

Підсумки результатів по всім методам наведені у таблиці 4.

Таблиця 4 – Підсумки обчислень

Методи	Стратегія f
Без вирогідностей	
Метод Вальда (Максимін)	1
Метод Севеджа (Мінімакс)	2
Метод Гурвіца	2
Метод Бернуллі	1
Найкращі стратегії:	2
3 приблизними вирогідностями	

Метод Ходжеса-Лемана	2
Найкращі стратегії:	2
Методи	Стратегія f
3 чіткими вирогідностями	
Метод Байєса	2
Метод Мінімуму дисперсії	1
Метод Максимізації ймовірності розподілу оцінного функціоналу	2
Метод Модальний	2
Метод Мінімуму ентропії	2
Метод Комбінованого критерію λ_1	2
Метод Комбінованого критерію λ_2	1
Метод Комбінованого критерію λ_3	2
Найкращі стратегії:	2
Підсумок:	2

Отже, стратегія f2 є найефективнішою.

Аналіз: відомо, що оптимальним рішенням є придбання серверу із найбільшим об'ємом дискового простору.

У ході пошуку оптимальної стратегії виявилось, що хоч знайдена оптимальна стратегія (нічого не робити) і не використовує залишений капітал, друга по ефективності стратегія (замінити сервер) передбачає додаткові витрати, які мають дуже великий шанс не задовольнити обмеження, виставлене залишковим капіталом.

Таким чином, має сенс переглянути вибір серверу та звернути увагу на сервер, що виявився другим по оптимальності – сервер з найнижчою ціною. Вибір такого серверу дозволить не тільки купити одразу декілька серверів такого типу, але і залишити більше грошей на можливі ризики.

Висновки: під час підрахування ефективності критеріїв серверів отримано одразу два оптимальні результати – сервери з мінімальною та максимальною цінами. Важливо зауважити, що перший сервер виявився незначно більш ефективним за другий. Проаналізувавши можливі ризики можна сказати, що такий вибір хоч і є найбільш ефективним, він також є найбільш небезпечним, тому що не передбачає можливості фінансових витрат у випадку їх необхідності.

ПЕРЕЛІК ПОСИЛАНЬ

1. Гіржев В.С. Розробка оптимального додатку. Електронний ресурс – 14 с.
http://eprints.library.odku.edu.ua/3510/1/Hirzhev%20VS_Rozrobka_optymaln_oho_dodatku_dlya_skladannya_rozkladu_zanyat_MKR_2018.pdf

Владимиров Яким Дмитрович, магістр
НТУ «Дніпровська політехніка», м. Дніпро

ОРГАНІЗАЦІЯ ДОСТАВКИ ТОВАРУ ІНТЕРНЕТ-МАРКЕТПЛЕЙСУ

Кожний інтернет-магазин фізичних товарів передбачає способи отримати куплені речі. Частіше за все покупці або самі забирають товар з рук продавця або з точок видачі, або використовують поштової сервіс для доставки товару у найближче поштове відділення або поштомат. Онлайн-сервіси з продажу ліків взагалі дозволяють забирати заздалегідь оплачені препарати з обраних лікарень.

З найпопулярніших типів отримання товару на сьогоднішній день виділяють [1]:

- доставку кур'єром;
- транспортні компанії;
- поштові сервіси;
- самовивіз.

Ринковий аналіз поштових служб станом на 2020 рік показує «позитивні прогнози розвитку української економіки 3-4% на рік разом з продовженням активного зростання споживання, а також активне зростання електронної комерції на 16-18% на рік», що є основними факторами для зростання всього ринку експрес-доставки [2].

Хоча у випадку інтернет-маркетплейсу імплементація даних методів не є прямою необхідністю, має сенс впровадження певного рівню інтеграції з ними для полегшення взаємодії як покупця з продавцем, так і обох типів користувачів із сайтом та, у окремих випадках, із сервісом доставки.

Це також полегшує комунікацію стосовно статусу замовлень, тому що додатково дозволяє покупцю відстежувати процес доставки товару. Така функція може контролюватися як продавцем, так і самим сервісом доставки. Чудовим прикладом першого варіанту є українська торгівельна платформа prom.ua [3],

прикладом другої, більш складної імплементації, є інтернет-маркетплейс одягу shafa.ua [4].

Так, імплементація систем взаємодії з поштовими сервісами доставки потребує часу та ресурсів, особливо зважаючи на те що кожен такий сервіс працює по-різному і є неможливим побудова однієї системи однакової взаємодії з будь-яким сервісом.

З цього приводу має сенс обмежити список сервісів, з якими буде будуватись взаємодія. Вибір можна провести на основі популярності сервісу та його рівню цифровізації. Так, згідно з The Page, найпопулярнішим поштовим сервісом за останній час є Нова Пошта [5], [2]. Повний список найпопулярніших поштових сервісів та їх порівняння приведено на рисунках 1-2 і в таблиці 1 [6].



Рисунок 1 – Популярність служб доставки серед інтернет-магазинів

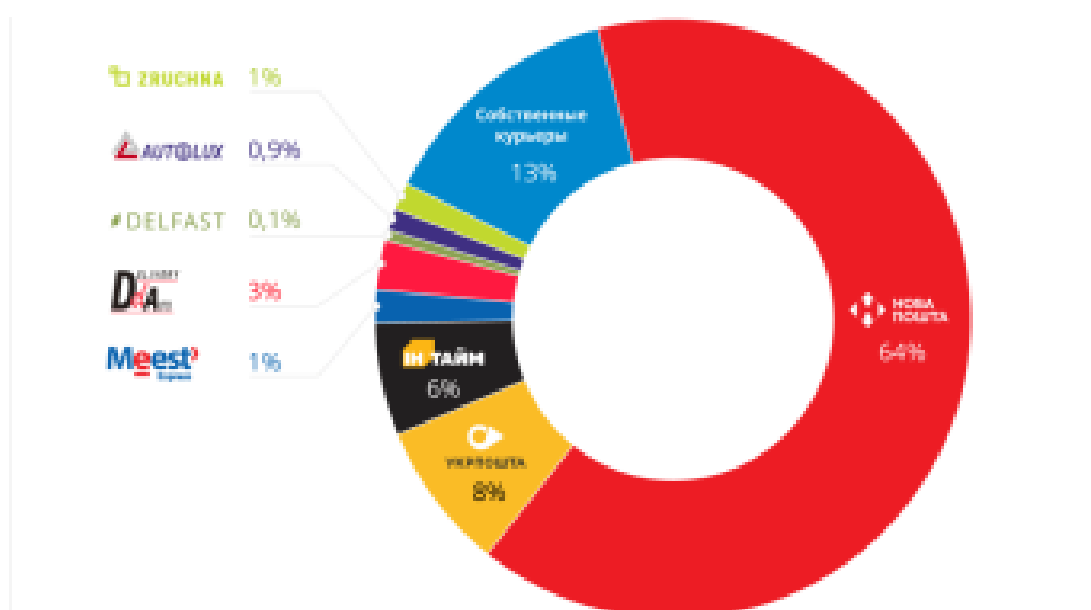


Рисунок 2 – Відсоток замовлень, доставлених окремою службою

Таблиця 1 – Особливості служб доставки

	Нова Пошта	УкрПошта	Meest	Meest Express	Делфаст	Делфаст	Аптека	Зручна
Кількість відділень	2448	Пошта 12808	648	2578	1800	8	200	8
Сроки доставки по Україні	1-2 дні	1-8 днів	1-2 дні	1-3 дні	1-3 дні	нескільки годин	1-3 дні	1 день
Наличчя мобільного провайдера	+	+	+	+	+	+	+	-
API	+	+	+	+	+	+	+	+
Програма лояльності	+	+	-	-	+	-	+	-
Міжнародна доставка	+	+	+	+	+	-	+	-
Фуфільмент	+	-	+	-	-	-	-	+
Отримання невеликих товарів на вибір	-	-	+	-	-	-	-	+
Доставка по часовим інтервалам	+	-	+	-	+	+	+	+
Перевозка крупногабаритних грузів	+	+	+	+	+	-	+	-

Можна побачити, що найпопулярнішими методами доставки є як і вказана раніше Нова Пошта, так і поштова служба ІнТайм, Укрпошта, а також власні кур'єри.

З приводу того, що інтеграції з останнім варіантом реалізувати майже неможливо, залишаються три кандидати – Нова Пошта, Укрпошта та ІнТайм.

На щастя, для полегшення інтеграції дані сервіси мають власні API, призначені для взаємодії з веб-додатками.

Так, Нова Пошта має власний портал для розробників з детальною документацією щодо використання їх API [7]. Укрпошта має схожу інтеграцією з повною документацією – Укрпошта API [8].

Нажаль, на даний момент ІнТайм не має повноцінної документації свого API, але присутні майже готові plug-in інтеграції, як наприклад очікувана інтеграція від APIxDrive [9].

З цього приводу, оптимальним рішенням буде зачекати на готову інтеграції з ІнТайм, та сфокусуватись на інших варіантах.

Висновок: після проведення ринкового аналізу поштових служб, виявлення їх основних характеристик з подальшим їх порівнянням, а також фільтрування списку опцій за даними характеристиками та можливістю інтеграції, було вирішено реалізувати інтеграцію лише з Новою Поштою та Укрпоштою. Взаємодію з із поштовою службою ІнТайм можна також провести пізніше, по факту наявності документації або готових інтеграцій. Реалізації інтеграції з іншими поштовими службами також можна реалізувати в майбутньому у випадку підвищення попиту на ці служби.

Література

1. Популярні способи доставки товарів з Інтернет-магазинів: що запропонувати покупцям (Інтернет-джерело): <https://neoseo.com.ua/uk/sposoby-dostavki-internet-magazina>
2. Аналіз Ринку Експрес-Доставки В Україні, с. 7 – Олексій Тараненко, 2020 р. (Інтернет-джерело): <https://www.econa.org.ua/index.php/econa>
3. Оплата та доставка, Загальне по доставці – Prom.ua (Інтернет-джерело): <https://help.prom.ua/hc/uk/sections/360002436058-%D0%97%D0%B0%D0%B3%D0%B0%D0%BB%D1%8C%D0%BD%D0%B5-%D0%BF%D0%BE-%D0%B4%D0%BE%D1%81%D1%82%D0%B0%D0%B2%D1%86%D1%96>
4. Як це працює – Shafa.ua (Інтернет-джерело): <https://shafa.kayako.com/ru/category/1-jak-ce-pracjuje>
5. Доставка «день у день» і покриття по всій країні: топ поштових операторів України – Діана Полуніна, The Page (Інтернет-джерело): <https://thepage.ua/ua/business/top-5-ukrayinskih-poshtovij-operatoriv-rejting-na-den-poshti>
6. Огляд українських служб доставки – Хорошоп (Інтернет-джерело): <https://horoshop.ua/blog/obzor-ukrainskikh-sluzhb-dostavki/>
7. Портал для розробників – Нова Пошта API (Інтернет-джерело): <https://developers.novaposhta.ua/>
8. API від Укрпошти – Укрпошта API (Інтернет-джерело): <https://dev.ukrposhta.ua/>
9. Інтеграція ІнТайм за API - APIxDrive (Інтернет-джерело): <https://apix-drive.com/ua/intime>

СХЕМАТИЗАЦІЯ СИСТЕМИ ДОСТАВОК ТОВАРУ ІНТЕРНЕТ-МАРКЕТПЛЕЙСУ

Внутрішні процеси будь-якої організації можуть представляти собою складні та заплутані системи, що взаємодіють одна з одною в рамках ще більшої, поверхневої системи. Найчастіше клієнти такої організації не підозрюють о процесах що відбуваються при їх взаємодії з нею.

Але навіть якщо знання цих процесів не є необхідним для клієнта, це знання може бути критичним для співробітника цієї організації. Це знання може бути обмежено підсистемою, у рамках якої працює співробітник, а може охоплювати всю систему загалом.

На такий випадок полегшення розуміння цієї системи є критичним.

Одним із способів зробити систему більш зрозумілою є схематизація та декомпозиція – зображення системи у вигляді схем її підсистем та їх взаємодії однієї з іншою. Для побудови підійде нотація EPC [1].

Як приклад, розглянемо підсистему доставки товару, купленого у інтернет-маркетплейсі. За уточнення візьмемо умову, що даний товар може бути як малим за розміром (можлива доставка більшістю поштових служб), так і великим (необхідна спеціалізована служба з перевезень, у даному випадку власний підрозділ з перевезень).

Так, схема, зображена на рисунку 1, відображає абстрагований процес купівлі товару від його вибору до його отримання.



Рисунок 1 – Загальна модель процесу купівлі товару

Це простий лінійний процес, де кожний крок представляє собою підсистему із своїми підпроцесами. Розглянемо крок «Доставка товару» більш детально (рисунком 2).

Можна побачити, що дана модель також містить процеси, що можна деталізувати та зобразити у вигляді окремих підсистем. До таких процесів відносяться «Доставка власним відділенням», що опише процес доставки відділенням перевезень компанії, та «Повернення товару», що представляє собою інакший процес перевезення товару.

Треба зауважити, що процес «Доставка <Назва служби>» є процесами взаємодії із окремими організаціями [2].

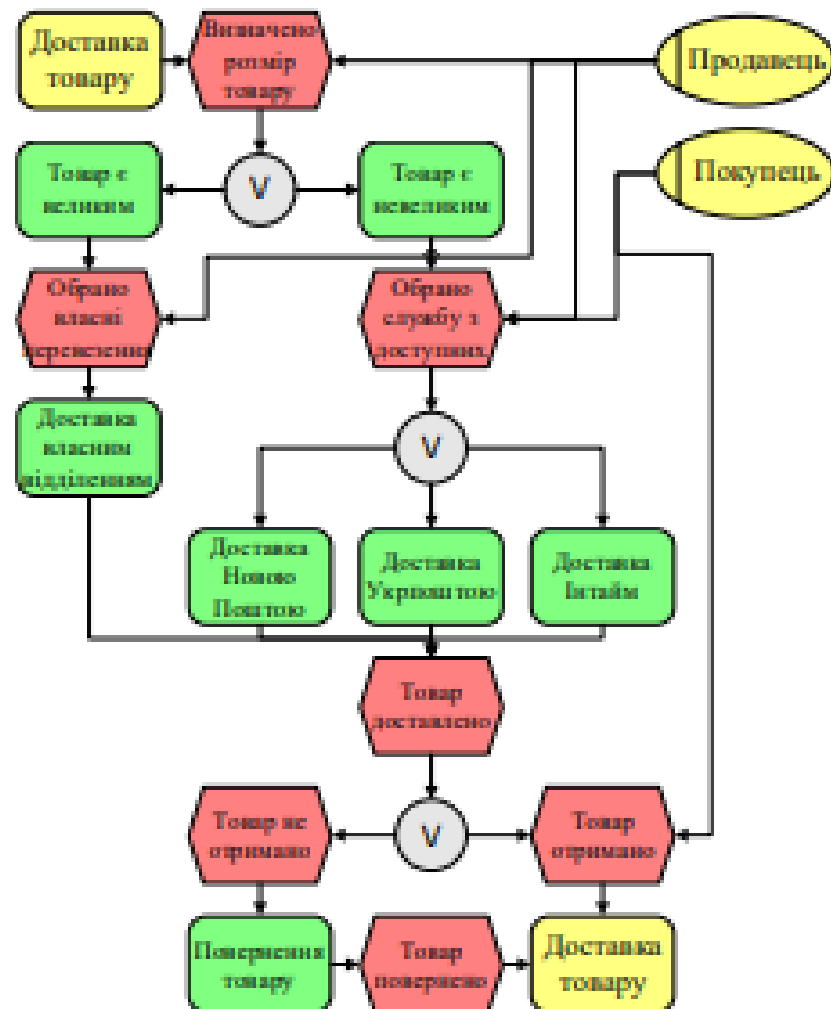


Рисунок 2 – Деталізація процесу «Доставка товару»

Висновок: після проведення схематизації та декомпозиції процесу доставки товару, було визначено не тільки системи інших організацій, з якими відбувається взаємодія, а й окремі процеси, що потребують подальшої деталізації.

Література

1. Нотація EPC – KN-груп. Управління проектами (Інтернет-джерело): <https://kn-grup.com/publications/glossary/257-event-driven-process-chain>
2. Популярні способи доставки з Інтернет-магазинів (Інтернет-джерело): <https://neoseo.com.ua/uk/sposoby-dostavki-internet-magazina>