

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
«КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
«ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ЕКОНОМІЦІ»**

Кафедра інформаційних систем в економіці

галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ПРОЕКТ

на тему

«Розробка телеграм-боту для порівняння цін на цифрову техніку»

студента Кандали Владислава Юрійовича _____

Науковий керівник:

к.е.н., професор

_____ Іванченко Г. Ф.

**Кваліфікаційний бакалаврський проект
допущений до захисту в Екзаменаційній
комісії з атестації здобувачів вищої освіти
завідувача кафедри _____ Тішков Б.О.**

Київ 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
«КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА»
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»

Кафедра інформаційних систем в економіці

галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри

Тішков Б.О.

« » 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

Студенту **Кандалі Владиславу Юрійовичу**

денна форми навчання

на підготовку кваліфікованого бакалаврського проекту

на тему: ***«Розробка телеграм боту для порівняння цін на цифрову техніку»***

Тему затверджено наказом ректора Університету від «11.03.2024. №529 ст
Кваліфікаційний бакалаврський проект виконується на матеріалах

План кваліфікаційного бакалаврського проекту

Розділ 1 виконати до 10.02.2024	Характеристика та аналіз предметної галузі (назва розділу)
Розділ 2 виконати до 20.03.2024	Розробка вимог і моделювання інформаційної системи (назва розділу)
Розділ 3 виконати до 30.03.2024	Проектування та реалізація компонентів системи (назва розділу)

Об'єкт дослідження:	розробка телеграм-боту, спрямованого на забезпечення користувачів актуальною інформацією про ціни на цифрову техніку
	в реалізації бот використовується для автоматизованого порівняння цін на різноманітні товари в області цифрової техніки
Предмет дослідження:	частина об'єкту, що розглядаються в рамках даного проекту, зокрема інформаційні процеси, комп'ютерні технології з пошуку найкращої ціни на вибраний продукт
Мета випускного бакалаврського проекту:	провести дослідження і виконати проектні рішення з розроблення інформаційної системи телеграм-боту для порівняння цін на цифрову техніку, створення ефективного боту, який забезпечить користувачів зручним і швидким способом отримання інформації про ціни на цифрову техніку

Конкретні завдання, які студент повинен виконати для досягнення поставленої мети:

У розділі 1 необхідно дати характеристику галузі – телеграм-боту для порівняння цін на цифрову техніку, визначити структуру інформаційної системи та функції її складових, особливу увагу слід приділити факторам, що впливають на організацію інформаційної системи, доцільно навести схему організаційної структури. Дослідити існуючі ІС для вибраної предметної області. Аналіз літературних джерел проводиться з метою виявлення прогресивних методик найшвидшого порівняння цін на техніку. Розробити концепцію ІС.

У розділі 2 провести аналіз і специфікацію вимог до інформаційної системи телеграм-боту. Дотримуючись вимог стандартів виконати постановку та алгоритм розв'язання задач з порівняння цін на цифрову техніку в боті.

У розділі 3 виконати проєктування таких компонентів системи: інформаційного – загальну схему інформаційного забезпечення, технічного – діаграма розгортання, програмного – діаграма компонентів, та організаційного – діаграма станів забезпечення у відповідності з методичними рекомендаціями кафедри, вказати можливості і доцільності впровадження.

**Завдання підготував
науковий керівник**

(підпис)

Іванченко Г. Ф.

(ініціали, прізвище)

« 28 » січня 2024 р.

Кандала В. Ю.

Завдання одержав студент

(підпис)

(ініціали, прізвище)

« 28 » січня 2024 р.

В і д г у к
про кваліфікаційний бакалаврський проєкт
здобувача навчально-наукового інституту «Інститут інформаційних технологій в
економіці»
спеціальності «Комп'ютерні науки»

Кандали Владислава Юрійовича

на тему
«Telegram-бот для порівняння цін на цифрову техніку»

1. Актуальність теми: Боти займають все більш вагоме значення в розвитку демократичного суспільства. При цьому телеграм-боти є одними з ланок, що характеризують простоту та легкість комунікації між покупцем та продавцем. Тому створення швидкого телеграм-боту для пошуку найкращих цін на цифрову техніку є на сьогодні актуальною проблемою.
2. Позитивні риси кваліфікаційного бакалаврського проєкту: на основі глибокого аналізу літературних джерел і практичного досвіду внесені пропозиції і розроблені проєктні рішення з мінімізації часу пошуку найкращої ціни на вибрану продукцію.
3. Наявність самостійних розробок: автор самостійно розробив ряд програмних модулів, які можуть бути використаними на практиці.
4. Цінність теоретичних висновків та практичних рекомендацій: дослідив зарубіжний та вітчизняний досвід розробки телеграм-ботів і вніс пропозиції про запровадження передових технологій у вітчизняну практику.
5. Наявність недоліків: можна відмітити наявність окремих стилістичних погрішностей.
6. Загальна оцінка кваліфікаційного бакалаврського проєкту та його допущення до захисту перед ЕК: Допустити до захисту кваліфікований бакалаврський проєкт Кандали В. Ю., оцінка відповідно.

Науковий керівник - професор, к.е.н., доцент
(посада, учене звання, науковий ступінь)

Іванченко Г. Ф.
(підпис) (прізвище, ініціали)

«___» _____ 20__ р.

АНОТАЦІЯ

кваліфікаційного бакалаврського проекту студента 4 курсу

Навчально-наукового інституту

«Інститут інформаційних технологій в економіці»

Кандали Владислава Юрійовича, виконаного на тему:

«Телеграм-бот для порівняння цін на цифрову техніку»

Київ: кафедра інформаційних систем в економіці, 2024 р.

Кваліфікаційний бакалаврський проект присвячений розробці інформаційної системи телеграм-боту для порівняння цін на цифрову техніку. Метою проекту є створення ефективного телеграм-боту, який забезпечить користувачів зручним і швидким способом отримання інформації про ціни на цифрову техніку. У процесі розробки проекту проведено аналіз існуючих інформаційних систем, їхніх сильних та слабких сторін, та визначено основні вектори для вдосконалення.

На основі зібраних даних та сучасних технологічних рішень сформовано концепцію інформаційної системи, яка охоплює всі аспекти діяльності телеграм-боту порівняння цін: від вибору компанії виробника до вибору сортування сайтів з продукцією. Система включає модулі для роботи з запитами покупців, управління різновидом сайтів для покупок товарів, редагування даних в базі даних. Значна увага приділяється інтуїтивності інтерфейсу телеграм-боту.

У першому розділі проведено аналіз галузі телеграм-боту для порівняння цін на цифрову техніку з метою визначення структури об'єкта управління та його функцій. Особлива увага приділена факторам, які впливають на організацію інформаційної системи. В роботі наведено схему організаційної структури телеграм-боту та виявлено прогресивні методики найшвидшого порівняння цін на техніку з використанням літературних джерел.

У другому розділі проведено аналіз та специфікацію вимог до інформаційної системи телеграм-боту згідно з вимогами стандартів. Розроблено постановку та алгоритм розв'язання задач з порівняння цін на цифрову техніку в боті.

У третьому розділі виконано проєктування таких компонентів системи як інформаційного, технічного та програмного забезпечення у відповідності з методичними рекомендаціями кафедри. Проаналізовано можливості та доцільність впровадження запропонованих рішень.

У додатках представлені фрагменти коду для бекенду системи, сторінки прототипу системи, розроблені у Figma, тестова документація, підготовлена за допомогою Enterprise Architect та результат перетворення ER-моделі в модель даних СУБД MySQL.

Висновки містять рекомендації щодо доцільності проєктування та впровадження інформаційної системи для телеграм-боту порівняння цін на цифрову техніку.

Також, визначено ключові структурні одиниці інформації, розроблено відповідну базу даних. Кваліфікаційний бакалаврський проєкт включає рекомендації щодо подальшої розробки та впровадження системи в реальних умовах ботів порівняння цін.

РЕФЕРАТ

Кваліфікаційний бакалаврський проект 74 сторінки, 4 таблиці, 58 рисунків, 26 перелік джерел посилань з найменувань, 4 додатки.

Проектування інформаційної системи для телеграм-боту порівняння цін на цифрову техніку.

Перелік ключових слів: *інформаційна система, цифрова техніка, телеграм-бот, порівняння цін, інновації, пошук найкращої ціни, телеграм, Figma, Об'єктно-орієнтоване програмування, діаграма Use case, діаграма послідовності, діаграма класів, стани, компоненти системи UML, SysML, Enterprise Architect, тест-кейси, функціональні вимоги, нефункціональні вимоги, бізнес-цілі...*

Об'єктом розроблення є розробка телеграм-боту, спрямованого на забезпечення користувачів актуальною інформацією про ціни на цифрову техніку. В реалізації бот використовується для автоматизованого порівняння цін на різноманітні товари в області цифрової техніки.

Мета кваліфікаційного бакалаврського проекту створення ефективного телеграм-боту, який забезпечить користувачів зручним і швидким способом отримання інформації про ціни на цифрову техніку.

Методами розроблення кваліфікаційного бакалаврського проекту є створення прототипу інформаційної системи в середовищі розробки PyCharm 2023.1.3 і використанні його відносно зручного функціоналу в виді додаткових функцій а саме: БД в MySQL, засоби програмування включають в себе мову програмування Python, бібліотеки для роботи з API, а також інструменти для розробки телеграм-ботів.

Апаратура використана при розробленні кваліфікаційного бакалаврського проекту є персональним комп'ютером:

- Процесор Intel(R) Core(TM) i5-7400 CPU @ 3,00GHz;
- Оперативна пам'ять – 8 Гб;
- Накопичувач – SSD, 1Тб;

- Екран – IPS, 24”.

Одержані результати можуть бути використані для практичного та теоретичного застосування, наприклад, у практичній діяльності великого торговельного бізнесу, розроблена система надає глобальним телеграм-ботам ефективний інструмент для швидкості та вибору цифрової техніки, запитами покупців, управління різновидом сайтів для покупок товарів, редагування даних в базі даних. Нововведення та підходи, розроблені в проекті, можуть слугувати базою для подальшої розробки подібних систем, а також надихати на створення нових інноваційних рішень.

Рік виконання кваліфікаційного бакалаврського проекту – 2024. Рік захисту кваліфікаційного бакалаврського проекту – 2024.

ЗМІСТ

АНОТАЦІЯ.....	2
РЕФЕРАТ	2
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	2
ВСТУП.....	3
РОЗДІЛ 1	5
ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	5
1.1 Характеристика предметної галузі та об'єкта дослідження	5
1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі	6
1.2.1 Дослідження існуючих інформаційних систем для обраної предметної галузі.....	6
1.2.2 Порівняння інформаційних систем	13
1.3 Розробка концепції інформаційної системи	15
РОЗДІЛ 2	18
РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	18
2.1 Аналіз і специфікація вимог до інформаційної системи.....	18
2.1.1 Функціональні вимоги до системи	19
2.1.2 Нефункціональні вимоги до системи	21
2.1.3 Бізнес-вимоги до системи	24
2.2 Обґрунтування методології проєктування та функціональна модель задачі.....	27
2.3 Моделювання предметної галузі	28
2.4 Постановка задачі.....	30
2.4.1 Вихідна інформація	31
2.4.2 Вхідна інформація	32
2.5 Математичне забезпечення та алгоритм функціонування системи	34
2.5.1 Математичний опис	34
2.5.2 Алгоритм розв'язання задачі.....	35
2.6 Моделювання інформаційної системи	37
2.6.1 Моделювання структури системи.....	40

2.6.2	Моделювання користувацького інтерфейсу.....	44
2.6.3	Прототипування користувацьких інтерфейсів.....	48
РОЗДІЛ 3.....		52
ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ		52
3.1	Інформаційне забезпечення.....	52
3.2	Організаційне забезпечення	56
3.3	Програмне забезпечення.....	57
3.4	Технічне забезпечення	58
3.5	Результати реалізації інформаційної системи	59
РОЗДІЛ 4.....		61
ТЕСТУВАННЯ І ВЕРИФІКАЦІЯ ПРОЕКТНИХ РІШЕНЬ		61
4.1	Розробка тест-кейсів	61
4.2	Трасування вимог до системи	63
ВИСНОВКИ		67
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....		70
ДОДАТКИ.....		74
ДОДАТОК А.....		74
	Тестова документація, підготовлена засобами Enterprise Architect.....	74
ДОДАТОК Б.....		80
	Фрагменти коду, що було створено для реалізації бекенду системи.....	80
ДОДАТОК В.....		91
	Сторінки прототипу системи зроблені у Figma	91
ДОДАТОК Г		96
	Результат перетворення ER-моделі в модель даних СУБД MySQL.....	96
ДОДАТОК Д.....		98
	Фрагменти ескізів форм подання вихідної інформації:	98

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД	База даних
ІС	Інформаційна Система
СКБД	Система керування базами даних
API	Application Programming Interface
UML	Unified Modeling Language
SysML	Systems Modeling Language
PyCharm	Інтегроване середовище розробки
SysML	Systems Modeling Language
BDD	Block definition diagrams
ООП	Об'єктно-орієнтований підхід
OPD	Object-Process Diagram
MVC	Model View Controller
SQL	Structured Query Language
IBD	Internal Block diagram

ВСТУП

Актуальність теми кваліфікаційного бакалаврського проекту «Проектування інформаційної системи телеграм-боту для порівняння цін на цифрову техніку» зумовлена стрімким розвитком технологій оброблення інформації та зростаючими вимогами ринку пошуку найкращої ціни на товари. У сучасному світі цифрової техніки пропозиція на ринку щоразу зростає, що ускладнює завдання споживача вибрати найбільш вигідну ціну для покупки. З метою полегшення цього процесу та забезпечення усвідомленого рішення, була обрана тема розробки телеграм-боту, який допомагатиме порівнювати ціни на цифрову техніку. Це вимагає впровадження комплексних інформаційних систем, які б забезпечували цілісний підхід до управління всіма аспектами вибору товару — від вибору виробника до вибору сортування вибраних сайтів.

Актуалізація теми також обумовлена розповсюдженням сайтів з не правомірними цінами на продукцію, споживачам потрібно знати, які пропозиції існують на ринку та де можна отримати найкращу ціну. Однак, пошук та порівняння цін вимагають значних зусиль з боку користувача, а також засобів інформації та аналітичних навичок. Тому розробка телеграм-боту, який автоматизує цей процес та надає швидкий доступ до актуальних цін, є надзвичайно актуальною та корисною.

Мета кваліфікаційного бакалаврського проекту – спроектувати інформаційну систему, а саме «телеграм-бот для порівняння цін на цифрову техніку».

Пошуки шляхів досягнення цієї мети обумовили необхідність визначення наступних завдань:

- дослідити предметну область для якої створюється інформаційна система;
- визначити вимоги до інформаційної системи (бізнес-вимоги, функціональні та не функціональні);

- обґрунтувати вибір програмних рішень для розроблення інформаційної системи;
- розробити інформаційну модель системи та описати вхідні і вихідні дані;
- розробити функціональну модель системи;
- розробити структурний алгоритм роботи інформаційної системи;
- спроектувати архітектуру інформаційної системи;
- спроектувати базу даних для інформаційної системи;
- спроектувати користувацький інтерфейс для інформаційної системи;
- скласти тест-кейси для перевірки вимог до інформаційної системи;
- представити прототип інформаційної системи;
- представити спроектовану систему в PyCharm 2023.1.3.

Практичне значення результатів, одержаних від проєктування інформаційної системи для порівняння цін на цифрову техніку в телеграм-боті полягає в значному підвищенні ефективності пошуку та вибору найкращого товару за найвигіднішою ціною. Система забезпечує автоматизацію ключових операцій, вдосконалення процесу вибору за багатьма параметрами. Це сприяє не лише швидкості вибору, але й допомагає сайтам отримати більший трафік покупців.

Використані інструментальні засоби: середовище проєктування I PyCharm 2023.1.3, система проєктування баз даних MySQL Workbench 8.0 CE, Python, Enterprise Architect, OPCAT, операційна система Windows 10, текстовий редактор Microsoft Word 2016 для підготовки та оформлення пояснювальної записки до кваліфікаційного бакалаврського проєкту.

Структура роботи зумовлена метою і завданнями та складається зі вступу, трьох розділів, висновку, переліку джерел посилання та додатків.

РОЗДІЛ 1

ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Характеристика предметної галузі та об'єкта дослідження

Месенджери відіграють велику роль у повсякденному житті сучасної людини. В кожного з нас є багато різних груп, каналів, звідки ми дізнаємося новини, що відбувається у закладі освіти, у місті та багато іншого. [1] Серед месенджерів, якими активно користується молодь можна виділити Telegram, для якого є окремий вид програмного забезпечення – телеграм-боти, які займають провідне місце в ІТ-сфері на сучасному ринку. Відбувається це тому, що телеграм-боти дозволяють вирішити повсякденні питання дуже швидко, спрощують деякі додатки та деяким користувачам більш зручно розмовляти із ботом, наприклад при оформленні замовлення, ніж із людиною, заповнюючи різні бланки інформації та даних, які бот заповнить автоматично. Зараз є багато програм, які дозволяють порівнювати ціни на якісь товари, проте розробка www-бота дозволить зробити вибір товару більш зручнішим та простішим за рахунок швидкості реагування ботом, структурованим вибором та зручним сортуванням.

Процес пошуку товару з найвигіднішою ціною здійснюється у два етапи.

Перший етап включає вибір покупцем необхідних параметрів, таких як марка товару, модель, тип та інше, в залежності від бажаного результату.

Другий етап передбачає вибір сортування сайтів за цінами, вибір найбільш вигідного в порівнянні з іншими, коли покупець вибрав бажаний сайт, він може перейти на нього та придати обраний товар.

Цей крок є важливою частиною процесу покупки товару за вигідною ціною в нашій інформаційній системі, що демонструє легкість вибору товару онлайн, забезпечуючи зручність та ефективність для покупців.

1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі

Вибір інформаційних систем для порівняння цін на цифрову техніку за допомогою телеграм-боту може бути визначеним конкретними потребами та масштабом пошукової системи яка влаштована до боту. В області пошуку найкращої ціни в телеграм-боті розглянемо три популярні ІС: Amazon Search & Pricex[2], GHop_bot[3], Volunteer Support[4].

1.2.1 Дослідження існуючих інформаційних систем для обраної предметної галузі

Для порівняльної характеристики не було знайдено вузько направлених додатків спрямованих саме на порівняння цін на цифрову техніку. Тож для аналізу були обрані популярні додатки, що надають можливість відтворювати найголовнішу функцію задуманого додатку – швидке та легке порівняння цін на побутову техніку та речі (такі як Amazon Search & Price, GHop_bot, Volunteer Support).

Загальні характеристики обраних додатків:

Загальна характеристика Amazon Search & Price

Amazon Search & Price телеграм-бот який спеціалізується на пошуку будь-якого товару – є лідером для шопінгу в цілому світі. Тому що його функціонал є різностороннім, має простий та доволі привабливий вигляд, що спрощує взаємодію користувача з ним.

В ньому не потрібно реєструватись, для замовлення товару, дозволено просто додати посилання на продукт і тільки потім сплатити його (рисунок 1.1).

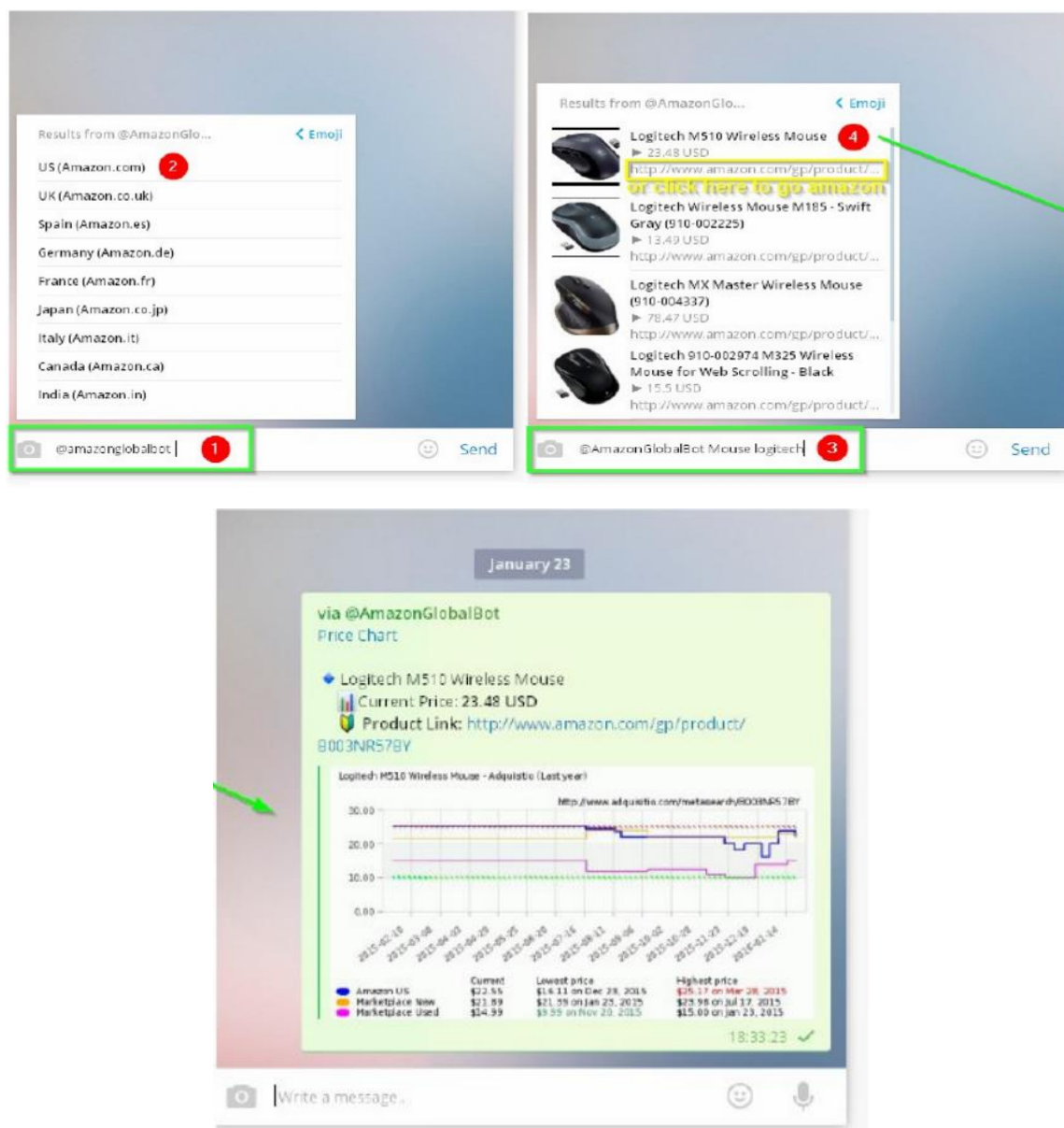


Рисунок 1.1 – Додавання товару в корзину телеграм бота Amazon Search & Price

Ще він може порівнювати товари між собою, потрібно тільки вказати посилання на ці два або більше товари. Також можна вибирати регіон, що полегшить пошук. Та швидкість доставки товару прямо під ваші двері.

Його переваги відносно інших розглянутих нижче додатків:

- Потужна система безпеки. Amazon Search & Price шифрує всі вхідні та вихідні повідомлення вибору товарів, які ви вибрали на своєму обліковому записи телеграму, не використовує обліковий запис за для просування реклами,

містить фільтр від спаму. Також надає можливість подивитись термін доставки товару.

- Скасування вибору товару: надає можливість скасувати оплату товару вже оплаченого на той момент, та переадресувати ці кошти на вибір іншого товару.

- Пошук товарів. Дозволяє швидко знаходити потрібний товар за ключовими словами, відправниками/одержувачами, темою, розміром, датою.

Але основним мінусом серед всього сказаного є його недоступність на території нашої країни, через це ми й розробимо нашого бота, який буде виконувати порівняння цін більш простіше та зручніше.

Загальна характеристика GHor_bot

GHor_bot телеграм-бот – це унікальний проєкт, який поєднує в собі одразу декілька складних рішень повсякденних питань для жінок, таких як: покупка одягу не виходячи з дому, а також оплата за допомогою картки, та доставка додому кур'єром. В нього є широкий вибір розділів, на які ми можемо натиснути та вибрати те, що нам потрібно (рисунок 1.2).

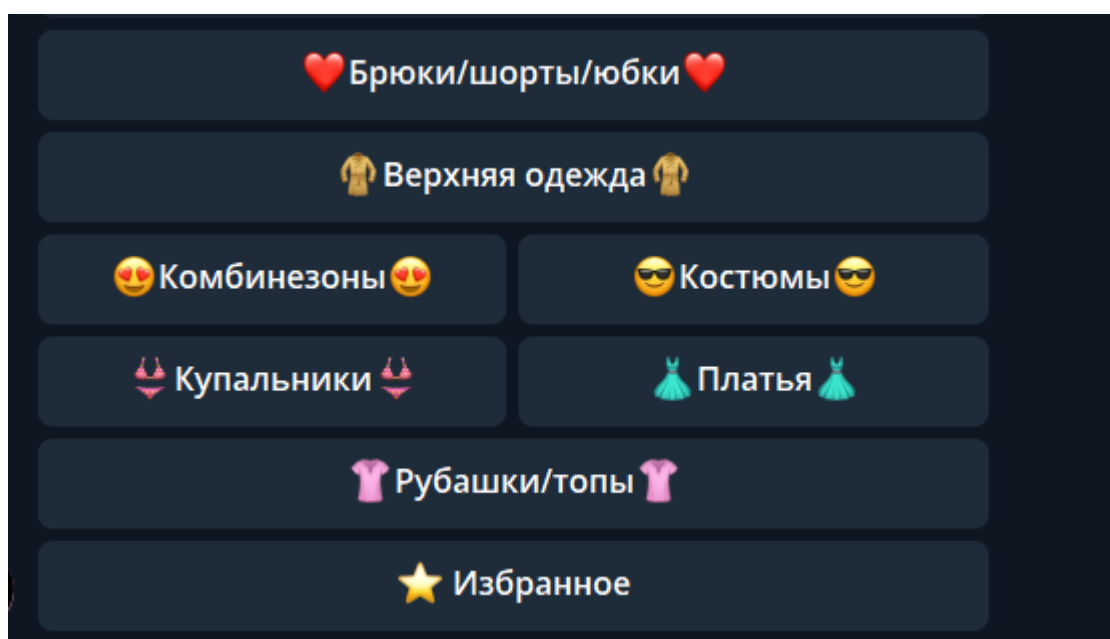


Рисунок 1.2 – Розділи GHor_bot-боту

Ці розділи дуже корисні, тому що можна проаналізувати помилки, щоб не допустити їх в нашому проекті. Простота відображення розділів – приваблює та концентрує увагу саме на них.

Переходячи в один з розділів, у нас впливає нове вікно, де у нас компоненти пов'язані напряму за певним типом одягу, наприклад «Верхній одяг» (рисунок 1.3), тобто іншими словами можна сказати що ми переходимо в підпункт, де саме вже й працюємо з іншими представленими кнопками, які в нас відображені на рисунку. В нашому боті буде взята одна функція звідси це «В начало каталога», ця функція є універсальним методом керування ботом.

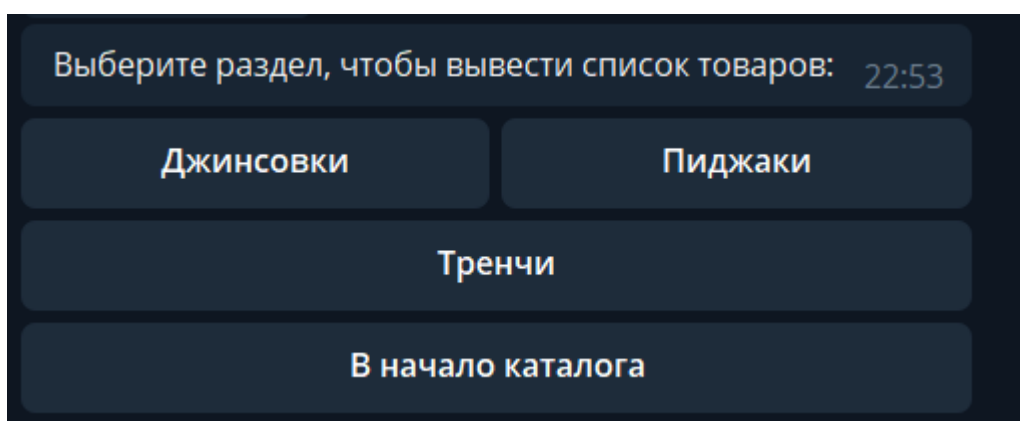


Рисунок 1.3 – Розділ товарів «Верхній одяг» у телеграм-боті GNor_bot

Переходячи далі в один з розділів, ми можемо побачити приклад з картинками, що концентрує нашу увагу саме на них. До того ж ми бачимо ще декілька кнопок, за якими можна здійснити дії тільки у відношенні до даної товару (рисунок 1.4).



Рисунок 1.4 – Пример товара телеграм-бота GHor_bot

Натискаючи на кнопку «купить», перед нами відкривається нове вікно, де ми можемо замовити саме цей товар. Саме функціонал цієї клавіші нам і цікавий для нашого телеграм-боту (рисунок 1.5).



Рисунок 1.5 – Замовлення в телеграм-боті GHor_bot

З цього телеграм-боту можна зробити декілька висновків. Чим простіше сам бот, тим краще для користувачів в розумінні та функціонуванні.

Загальна характеристика Volunteer Support

Є ще один бот, і його функціонал допоможе в розробці нашого, за допомогою, часткового копіювання його вигляду. Таким телеграм-ботом – є Volunteer Support (рисунок 1.6).

Він в собі поєднує багато стандартних методів та функцій, що робить його простим та зрозумілим у використанні, до того ж, ми можемо використати це в своїй роботі, що значно зменшить нам час розробки, та допоможе реалізувати ще більше функціоналу.

Volunteer Support – це унікальний проєкт, який поєднує в собі простоту функціоналу та багатогранність відповідей й питань, який розроблений командою Телеграм, для відповідей користувачам, на питання, які їх цікавлять, такими ж користувачами, що пройшли сертифікацію від телеграму.

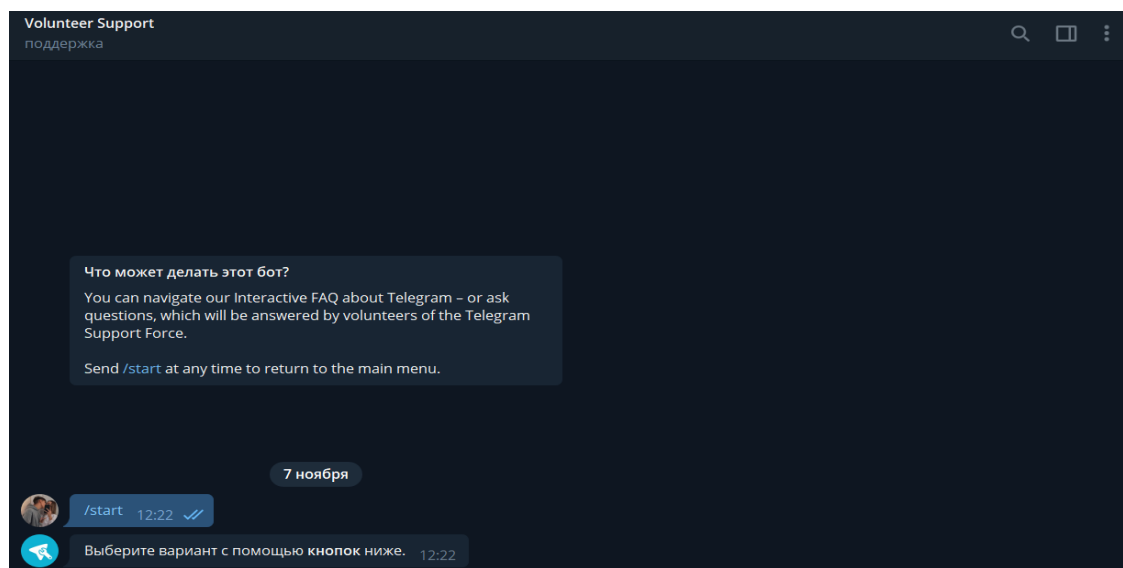


Рисунок 1.6 – Volunteer Support (вхід в телеграм-бот)

В нього всього декілька розділів але вони мають широкий спектр дій (рисунок 1.7).

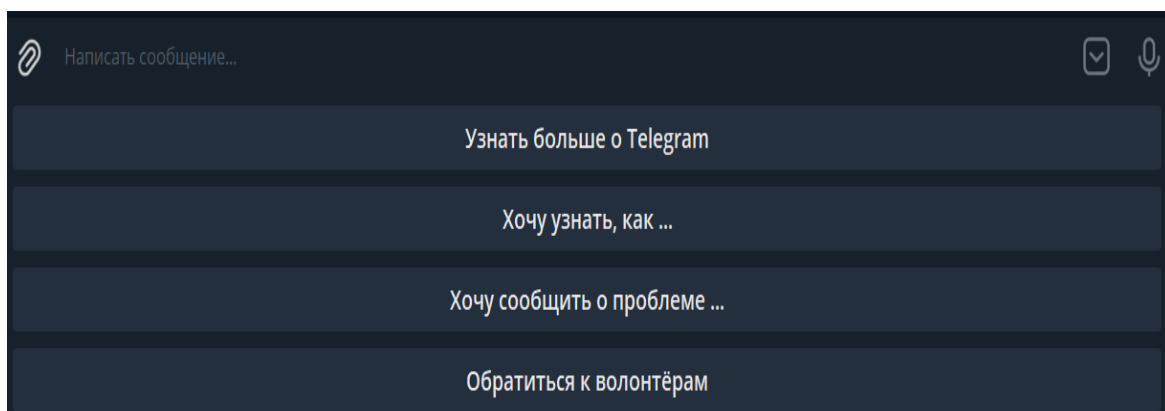


Рисунок 1.7 – Розділи боту Volunteer Support

Вибираючи один з розділів нас перекине далі, де ми, з запропонованих відповідей, можемо вибрати одну, і потім вже описати проблему, або дізнатись щось корисне для нас, чи запропонувати щось своє, нове, або покращення цього ПЗ (рисунок 1.8).

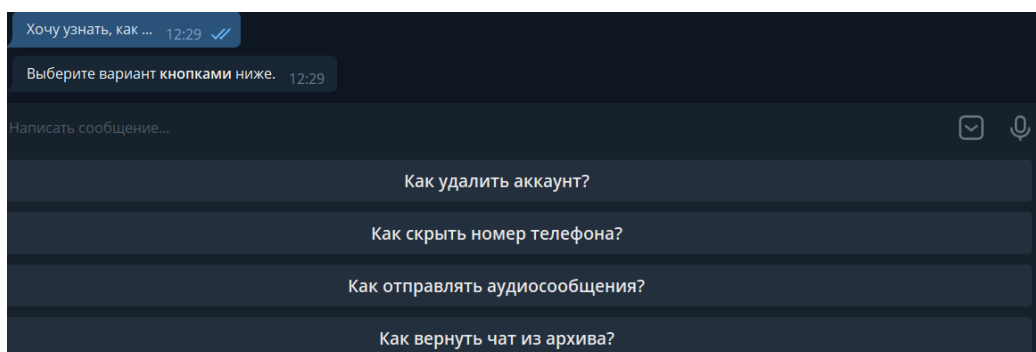


Рисунок 1.8 – Влаштовані розділи автоматичних відповідей на питання користувачів в Volunteer Support

Кожна з цих систем має свої переваги та обмеження, компонування цих систем у нашому боті, буде як основна перевага на бік покупців для легкості вибору найкращої ціни.

1.2.2 Порівняння інформаційних систем

Результати порівняння програмних систем Amazon Search & Price, GHop_bot, Volunteer Support, наведені в табл. 1.1:

Порівняння програмних систем

Таблиця 1.1

№	Можливості	Volunteer Support	<u>Amazon Search & Price</u>	<u>GHop_bot</u>
Зручність використання систем				
1	Відображення та завантаження активних посилань	Швидко та легке	Багато етапів підтвердження перед переходом за посиланням	Довге підвантаження посилань продукції
2	Інтуїтивна зручність інтерфейсу	Інтуїтивний	Зручний та легко	Громісткий та складний
3	Навігація системою	Легка та логічна	Проста та доступна	Інтуїтивна та швидка
4	Мобільний доступ	Підтримує мобільний доступ та оптимізований для пристроїв	Забезпечує мобільний доступ, інтерфейс оптимізований для мобільних пристроїв	Зручний інтерфейс
5	Візуалізація інформації	Графічне відображення ключових показників	Зручна візуалізація результатів пошуку	Графічне відображення інформації
Подання інформації				
6	Достатня кількість місця для подання необхідної інформації	так	так	так
7	Можливість завантаження великих файлів	так	так	ні
8	Завантаження документів та їх перегляд	так	так	так
9	Позначення важливої інформації	так	так	так
10	Пошук необхідного повідомлення	так	так	так
11	Структуроване подання інформації	ні	так	ні
12	Керування доступом на редагування/перегляд інформації	так	так	так
Спілкування				
13	Онлайн-чат в боті	ні	так	ні

14	Сповіщення про акції та пропозиції	так	так	ні
15	Система зворотного зв'язку	так	так	ні

Висновки за результатами аналізу програмних додатків:

Отже, розглянувши таблицю можна побачити що для пошуку товарів в мережі Інтернет найбільш зручним є сервіс Amazon Search & Price. Він дозволяє подати інформацію легко та просто для користувача, розділити її на блоки, додати всі необхідні фотографії, структуровано подати інформацію яка буде максимально зручна у використанні користувачеві, але він не працює на території нашої країни, через це ми й будемо створювати нашого бота.

Друге місце посідає телеграм бот Volunteer Support. Він доволі легкий у використанні, але має обмежений функціонал та саму структуру користування й взаємодії, так як має вузький функціонал.

Всі інші додатки розділили третє місце за кількістю недоліків.

Позитивні характеристики існуючих додатків:

- Можливість сортування інформації за назвами, або за ціною.
- Зворотній зв'язок з користувачами доволі важлива річ яка дозволяє нам краще розуміти що саме зараз потрібно йому.
- Інтуїтивна зручність інтерфейсу. Додаток має дати можливість швидко зрозуміти де що знаходиться та як цим користуватись, це все досягається шляхом простого інтерфейсу та структуровано направленою коду.

GNor_bot – вузько направлений бот, який не працює весь час та має обмеження за багатьма функціями такими як: пошук по слову, сортування за ціною. Що не робить його компактним та легким в функціонуванні, але не досить практичним.

В кожному з них, інтерфейс доволі місткий, та має навантаження на користувача, не має інтуїтивного та зрозумілого використання, всі вікна відображення знаходяться в різних частинах ботів, які складно знайти користувачеві не проклікав всі кнопки. Розв'язком цієї проблеми є поєднання легкості користування, простоти та інтуїтивності, разом з нативно зрозумілим

інтерфейсом з урахуванням специфіки відповідної платформи та доступом до всіх її ресурсів, який може допомогти користувачеві, що й буде відтворено в нашому майбутньому боті.

1.3 Розробка концепції інформаційної системи

Головним архітектурним підходом, обраним для розробки системи, є архітектура Agile, яка базується на гнучкості та ітеративності, що дозволяє внести зміни до вимог на будь-якому етапі розробки, що особливо важливо для проекту, де інформація про товари може часто змінюватися. Крім того, Agile дозволяє почати з мінімальною функціональністю та поступово розширювати її. Цей підхід забезпечує легкість та швидкість побудови системи.[5]

На рівні представлення фокус зосереджений на розробці інтерфейсу користувача, що включає обробку введення параметрів пошуку покупця та забезпечення інтуїтивності в діях.

Рівень доступу до даних забезпечує взаємодію з базою даних або іншими джерелами даних. Він містить класи для доступу та маніпуляції даними, такі як репозиторії або DAO (Data Access Objects), гарантуючи ефективне управління даними та їх цілісність. [6]

У межах архітектури Agile також використовуються класи, що представляють компоненти моделі, вигляду та контролера. Це дозволяє розробити баланс між зручністю користувача, ефективністю оброблення даних та гнучкістю управління бізнес-процесами. Така структура системи не тільки відповідає сучасним технічним стандартам, але й забезпечує масштабованість та адаптивність до змінних потреб пошуку найкращої ціни в телеграм-боті.

Також, у проекті для розробки інформаційної системи телеграм-боту для порівняння цін на цифрову техніку було обрано мову програмування Python, яка є однією з найбільш розповсюджених та надійних мов у світі розробки програмного забезпечення. Python відома своєю масштабованістю, безпекою та

легкістю написання кода, що робить її ідеальним вибором для створення великих, складних додатків, таких як корпоративні інформаційні системи.

Для розробки використовується середовище розробки PyCharm 2023.1.3[7], яке забезпечує широкий спектр інструментів для розробників та підтримує роботу з Python та іншими мовами програмування. PyCharm 2023.1.3 відзначається своєю продуктивністю, зручністю інтерфейсу та розширеними можливостями для відладки коду, що сприяє підвищенню ефективності процесу розробки.

У цілому, в моєму проекті використовується об'єктно-орієнтований підхід (ООП). Цей підхід дозволяє структурувати програмне забезпечення шляхом створення об'єктів, що представляють реальні сутності та концепції. Об'єктно-орієнтоване програмування забезпечує високу рівень абстракції та інкапсуляції, що дозволяє ефективно управляти складними системами, спрощуючи розробку, тестування та підтримку.[8]

Завдяки ООП, розробка стає більш модульною, що сприяє легшому повторному використанню коду та покращенню його читабельності. Також цей підхід сприяє більш ефективному управлінню пам'яттю та ресурсами, а також надає можливість легшого масштабування та адаптації програмного забезпечення до змінних вимог і потреб користувачів.

У моєму проекті інформаційної системи для телеграм-боту порівняння цін на цифрову техніку для розробки бази даних було обрано MySQL Workbench. Цей вибір зумовлений низкою переваг цього інструменту. Перш за все, MySQL Workbench відомий своїми візуальними інструментами для проєктування, що дозволяють легко створювати, візуалізувати та редагувати схеми баз даних. Це допомагає як в створенні структури даних, так і в розробці зв'язків та обмежень між таблицями системи, що є критично важливим для ефективного управління даними.[9]

Другий аспект, який робить MySQL Workbench відповідним для проекту, полягає в його здатності інтегруватися з іншими технологіями та

інструментами розробки, що використовуються в проєкті, включаючи Python та PyCharm. Це забезпечує плавну взаємодію між розробкою бази даних та іншими компонентами системи, що сприяє більш ефективному та координованому процесу розробки. Така інтеграція є ключовою для створення єдиного, злагодженого та простого середовища розробки, яке підтримує всі аспекти інформаційної системи.

MySQL Workbench надає великі можливості для внесення змін та обслуговування бази даних. Це включає інструменти для виконання запитів SQL, тестування їх продуктивності та оптимізації, а також можливості для резервного копіювання, моніторингу бази даних. Такий комплексний підхід до управління даними забезпечує високу продуктивність, безпеку та надійність інформаційної системи, що є важливим для стабільного та ефективного функціонування телеграм-боту.

У розробці фронтенду ключову роль відіграють Телеграм та PyCharm. Телеграм використовується як основа на якій будується інтерфейс за допомогою PyCharm. Телеграм як пластилін, на основі якого можливо будувати будь-яку систему що може ним підтримуватись, головне правило при роботі з ним, це чітка структура в написанні відповідного коду.

Для демонстрації користувацького інтерфейсу було створено прототип (клієнтської частини) інформаційної системи у векторному онлайн-сервісі Figma.

РОЗДІЛ 2

РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Аналіз і специфікація вимог до інформаційної системи

Задача аналізу потреб полягає у зборі, розумінні та аналізі детальних вимог і очікувань від системи з боку зацікавлених сторін. Це включає в себе адміністратора бота та покупців. Кожна з цих груп має свої унікальні потреби та очікування, які повинні бути враховані під час розробки системи.

Основні задачі аналізу інформаційної системи:

1. Адміністратор телеграм-боту:
 - потреби у швидкому редагуванні БД;
 - максимізація прибутковості та оптимізації використаних ресурсів.
2. Покупці в телеграм-боті:
 - очікування щодо простоти та зручності вибору найкращої ціни на вибрану продукцію;
 - можливість вибору серед багатьох сайтів конкурентів;
 - вибір серед компаній та моделей.

Далі, потрібно скористатися збором інформації, а саме проаналізувати тенденції ринку та дослідити сучасні, популярні практики у сфері телеграм-ботів з порівнянням цін.

На основі цього визначаються наступні види вимог:

- функціональні, які визначають, що система повинна робити;
- нефункціональні, які визначають вимоги до якості та характеристик інформаційної системи;
- бізнес-вимоги, які враховують потреби бізнесу та стратегічні цілі.

Мета цього аналізу є забезпечення глибокого розуміння вимог і очікувань усіх сторін, щоб розроблена інформаційна система була максимально

ефективною та відповідала потребам сучасного телеграм-боту з порівняння цін. Це допоможе розробити систему, яка не лише вирішує поточні завдання, але й має потенціал для адаптації та розвитку масштабного ринку телеграм-ботів.

2.1.1 Функціональні вимоги до системи

Функціональні вимоги визначають основні можливості та функції, які наш телеграм-бот ІС для пошуку найкращої ціни на продукцію за допомогою телеграм-боту повинна надавати для задоволення потреб покупців та досягнення бізнес-цілей проекту.

Характеристика головних функціональних вимог системи:

Пошукова функція, пошук потрібного товару за вибраними критеріями:

Система повинна забезпечувати можливість швидкого та зрозумілого пошуку потрібного товару за такими критеріями як: марка товару, вид, модель та тип сортування.

Перегляд результатів, а перегляд кожного етапу своїх дій:

Покупці повинні отримувати фідбек від боту, а саме відповідь з візуальним супроводженням після кожного етапу: вибору марки товару, вид, модель та тип сортування.

Підключення БД, взаємодія з даними які містяться в ній:

1. Система повинна здійснювати обмін даними по запиту користувача, та видавати результат за лічені секунди, який згенерований базою даних.

Детальніше можна переглянути на діаграмі функціональних вимог, яка представлена на рисунку 2.1.

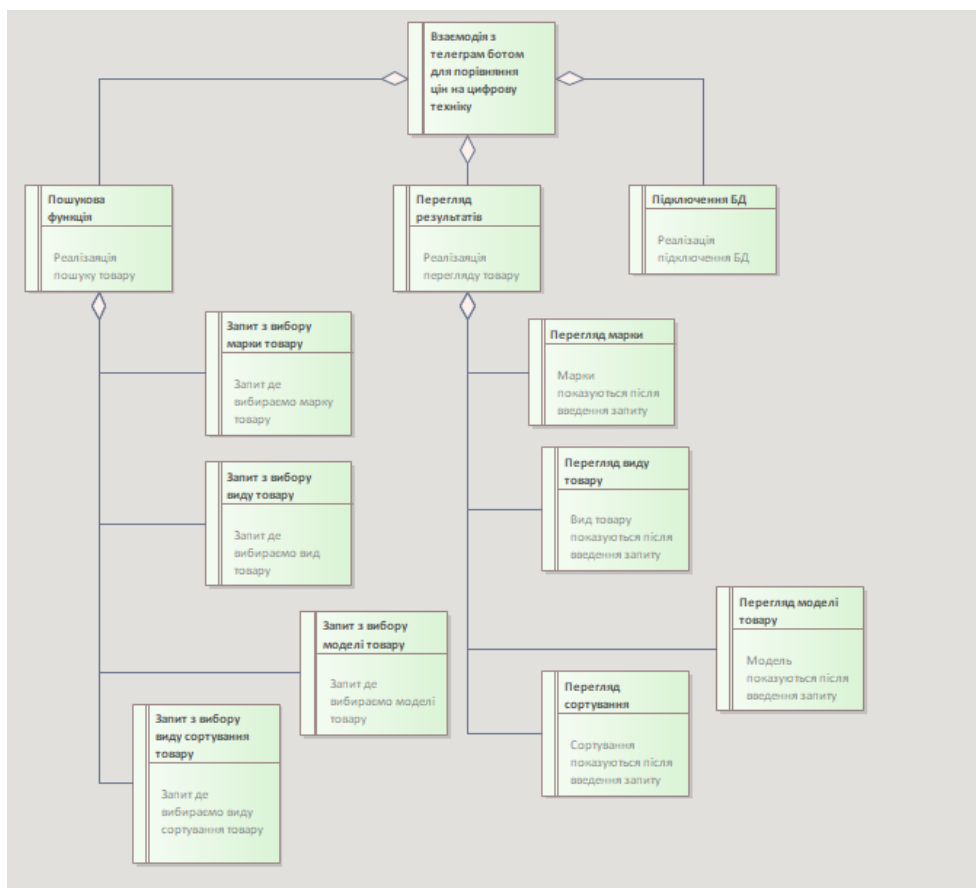


Рисунок 2.1 — Діаграма функціональних вимог

Також на рисунку 2.2 представлено специфікацію функціональних вимог створену за допомогою менеджера специфікацій. В специфікаціях вимог серед іншого зазначено: тип вимоги, статус вимоги, її складність та пріоритетність.

Specification Manager

Specification Manager:: Package: "Requirements" [Requirement]

Model Two Level Requirement Hierarchy Requirements

Item	Stereotype	Status	Difficulty	Priority
☒ Взаємодія з телеграм ботом для порівняння цін на цифрову техніку		Proposed	High	High
☒ Перегляд результатів Реалізація перегляду товару		Approved	High	Medium
☒ Перегляд виду товару Вид товару показуються після введення запиту		Approved	Low	Medium
☒ Перегляд марки Марки показуються після введення запиту		Approved	Low	Medium
☒ Перегляд моделі товару Модель показуються після введення запиту		Approved	Low	Medium
☒ Перегляд сортування Сортування показуються після введення запиту		Approved	Low	Medium
☒ Підключення БД Реалізація підключення БД		Proposed	High	High
☒ Пошукова функція Реалізація пошуку товару		Proposed	High	High
☒ Запит з вибору виду сортування товару Запит де вибираємо виду сортування товару		Proposed	Medium	Medium
☒ Запит з вибору виду товару Запит де вибираємо вид товару		Proposed	Medium	Medium
☒ Запит з вибору марки товару Запит де вибираємо марку товару		Proposed	Low	Low
☒ Запит з вибору моделі товару Запит де вибираємо моделі товару		Proposed	Medium	Medium

Рисунок 2.2 — Специфікація функціональних вимог

2.1.2 Нефункціональні вимоги до системи

Нефункціональні вимоги системи телеграм-боту для порівняння цін на цифрову техніку визначають критерії, які не стосуються безпосередньо основних функцій інформаційної системи, але мають вирішальне значення для її ефективності та придатності до використання.

Вони охоплюють багато аспектів, такі як: структурне подання інформації, зручність та швидкість отримання результатів.

Ключовою нефункціональною вимогою є забезпечення швидкості та ефективності роботи системи навіть при великому обсязі даних та великому

напливу покупців які шукають найвигідніші умови для покупки собі товару. Вимагається мінімізувати час відповіді системи на запити користувачів не більше 5 секунд, оптимізація роботи бази даних для швидкої взаємодії з нею.

Не менш важливою вимогою є структурне подання інформації. Система повинна мати структуру подання та коротку інформацію до кожного з пунктів, таких як: компанія товару(марка), параметри та відображення моделі, інформацію про техніку, інформацію щодо бота, різновид девайсів, різновид моделей, тип сортування магазинів за цінами.

Ще однією нефункціональною вимогою є зручність використання. Включає в себе зручну навігацію(нативні примітки), які допомагають для кращої взаємодії з ботом та хостинг, який дає можливість використання бота в будь-який проміжок часу.

Опис кожного блоку:

1. Компанія товару – повинна бути серед компаній, які займають провідні місця у рейтингу найпопулярніших в світі.

2. Різновид девайсів – повинен бути складений з тих, які стали найпопулярнішими за останній рік.

3. Різновид моделей девайсів – повинен бути складений з тих, які стали найпопулярнішими за останній рік.

4. Параметри та відображення моделі – повинен бути візуальний контакт, та розуміння у користувача що саме він вибирає та з якими параметрами.

5. Тип сортування магазинів за цінами на цифрову техніку – сортування повинно бути 2 варіацій за зростанням, або спаданням.

6. Інформація про техніку – про кожну техніку повинна міститись інформація в БД для зручного порівняння моделей та вибору кращої.

7. Інформація щодо бота – невеликий вступ про бота, для чого він був створений та що він вміє.

Загальні нефункціональні вимоги можна оглянути на рисунок 2.3 таспецифікації вимог зроблені за допомогою менеджера специфікацій (Specification Manager) Enterprise Architect, що зображені на рисунок 2.4.

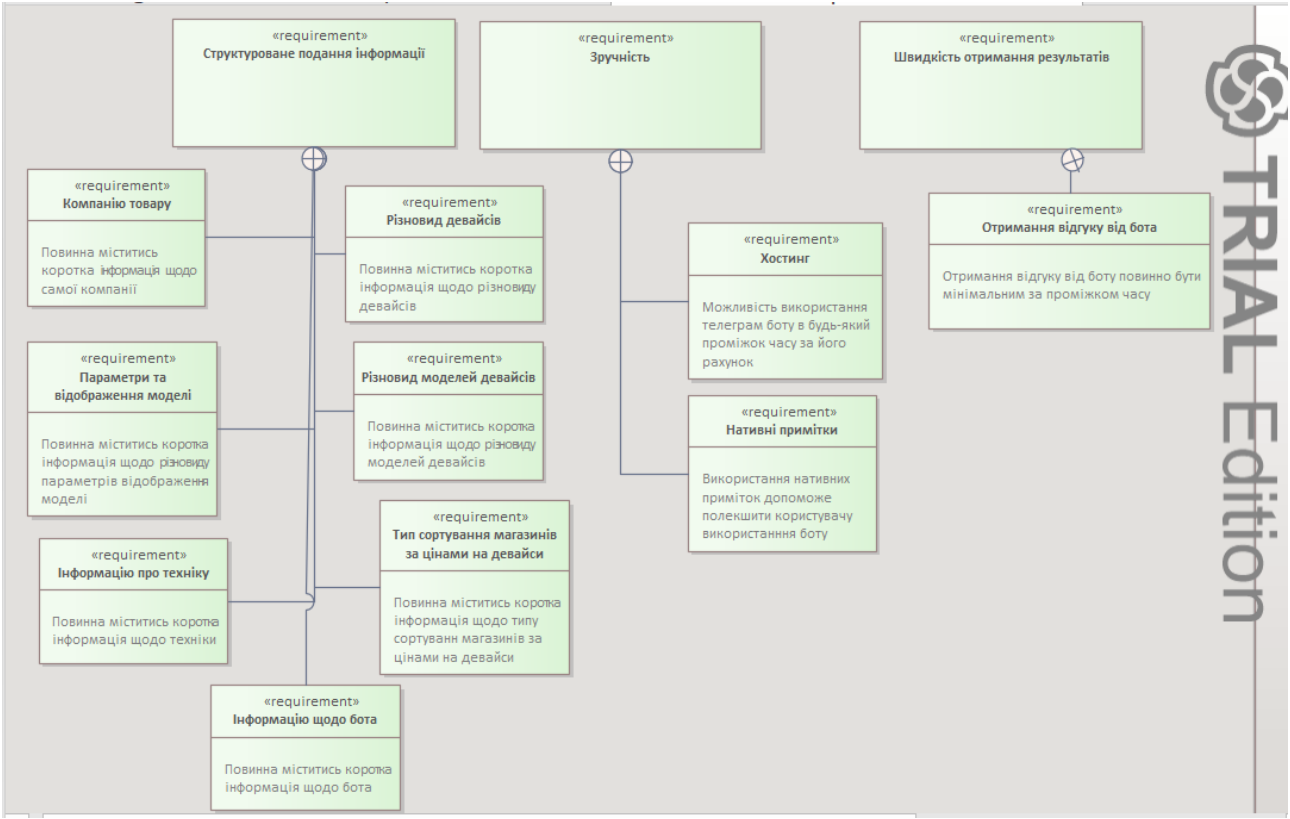


Рисунок 2.3 — Діаграма нефункціональних вимог

Specification Manager				
Specification Manager:: Package: "Requirements" [Requirement]				
← → ▶ / ▶ Model ▶ Two Level Requirement Hierarchy ▶ Requirements				
Item	Stereotype	Status	Difficulty	Priority
☒ Структуроване подання інформації	FunctionalReq...	Proposed	High	High
☒ Зручність	FunctionalReq...	Approved	Medium	High
☒ Швидкість отримання результатів	FunctionalReq...	Approved	Medium	Medium
☒ Компанію товару Повинна міститись коротка інформація щодо самої компанії	FunctionalReq...	Approved	High	Medium
☒ Інформацію про техніку Повинна міститись коротка інформація щодо техніки	FunctionalReq...	Proposed	Medium	Low
☒ Інформацію щодо бота Повинна міститись коротка інформація щодо бота	FunctionalReq...	Approved	Medium	Medium
☒ Параметри та відображення моделі Повинна міститись коротка інформація щодо різновиду параметрів відображення моделі	FunctionalReq...	Proposed	Medium	Medium
☒ Різновид девайсів Повинна міститись коротка інформація щодо різновиду девайсів	FunctionalReq...	Approved	Low	Low
☒ Різновид моделей девайсів Повинна міститись коротка інформація щодо різновиду моделей девайсів	FunctionalReq...	Approved	Medium	Medium
☒ Тип сортування магазинів за цінами на девайси Повинна міститись коротка інформація щодо типу сортуванн магазинів за цінами на девайси	FunctionalReq...	Proposed	High	Medium
☒ Отримання відгуку від бота Отримання відгуку від боту повинно бути мінімальним за проміжком часу	FunctionalReq...	Proposed	Medium	Low
☒ Хостинг Можливість використання телеграм боту в будь-який проміжок часу за його рахунок	FunctionalReq...	Approved	High	High
☒ Нативні примітки	FunctionalReq...	Proposed	Low	<u>Low</u>

Рисунок 2.4 — Специфікація нефункціональних вимог

2.1.3 Бізнес-вимоги до системи

Сформовані бізнес-цілі визначають основну мету створення системи телеграм-боту для порівняння цін на цифрову техніку та забезпечують відповідність її функціональності потребам покупців і вимогам ринку серед провідних телеграм-ботів.

Однією з цілей є орієнтованість на подальший розвиток та масштабованість боту, за допомогою інших цілей, таких як: монетизація – яка допоможе почати отримувати кошти та розбудовувати подальшу систему телеграм-боту, рекламні можливості для виробників продукції – це можливість розміщувати рекламу в середині боту, на самі цікаві та вигідні товари, партнерство з виробниками – відтворення їх контенту в боті, довгострокові рекламні компанії для просування в маси їх продукції.

Бізнес-вимоги визначають функціональність та характеристики, які має мати інформаційна система для задоволення потреб інвесторів і досягнення цілей бізнесу.

В ІС для телеграм-бот для порівняння цін на цифрову техніку інвестори включають розробника ПЗ та адміністратора бота. Кожен інвестор має свої унікальні потреби та очікування від системи, які визначаються вимогами до системи. Наприклад, розробник коду може бажати мати звіти про прибуток та витрати на рекламу, тоді як адміністратор телеграм-боту має потребувати можливість управління, редагування та корегування БД.

Діаграма бізнес-вимог представлена на рисунку 2.5, а на рисунку 2.6 представлені специфікації бізнес-вимог.

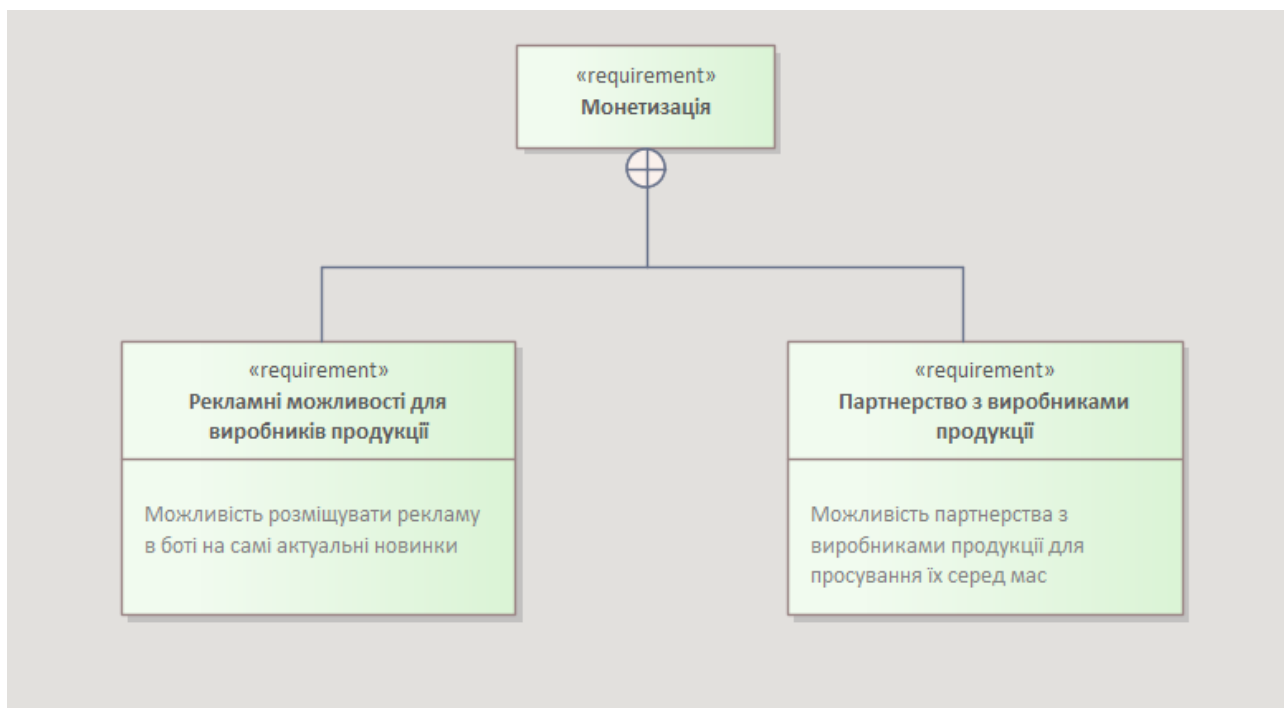


Рисунок 2.5 — Бізнес-вимоги

Start Page Specification Manager x				
Model One Level Requirement Hierarchy Requirements Find Package				
Item	Stereotype	Status	Difficulty	Priority
☒ Монетизація	FunctionalReq...	Proposed	High	<u>High</u>
☒ Партнерство з виробниками продукції Можливість партнерства з виробниками продукції для просування їх серед мас	FunctionalReq...	Proposed	Medium	High
☒ Рекламні можливості для виробників продукції Можливість розміщувати рекламу в боті на самі актуальні новинки	FunctionalReq...	Approved	Low	Low
Add New ▼				

Рисунок 2.6 — Специфікація бізнес-вимог

2.2 Обґрунтування методології проєктування та функціональна модель задачі

Для кваліфікаційного бакалаврського проєкту на тему проєктування інформаційної системи телеграм-боту для порівняння цін на цифрову техніку, я обрав методологію розробки – Agile. Використання Agile дозволяє оптимізувати процес розробки за рахунок поступової розробки проєкту та внесенню змін на будь-якому етапі розробки. На відміну від більш традиційних методів, як Kanban board, Agile фокусується на практичному використанні програмного забезпечення та постійному безперервному його вдосконаленню. Цей підхід забезпечує гнучкість та адаптивність у процесі розробки, що є особливо важливим у контексті динамічно змінюваних вимог порівняння цін на цифрові продукти в телеграм-боті. [10] Основні переваги Agile включають покращення швидкості розробки, поліпшення управління ризиками що забезпечує ефективну взаємодію з на ранніх стадіях проєкту.

Окрім цього, у проєкті застосовується об'єктно-орієнтований підхід (ООП), який дозволяє гармонічно організувати структуру програмного коду. Завдяки використанню принципів послідовної розробки та розбиттю кожного етапу на блоки, ми можемо досягти поставленої мети в розробці, за доволі короткий проміжок часу.[11] Це особливо важливо для такого телеграм боту як наш, де необхідно ефективно управляти різноманітними бізнес-процесами, від заключенням договорів з партнерами, закінчуючи влаштованою рекламою в середині бота. Використання ООП дозволяє не тільки покращити якість та зрозумілість проєкту, але й забезпечує легкість у підтримці та розширенні інформаційної системи в майбутньому.

Відповідно, поєднання методології Agile та ООП у розробці інформаційної системи забезпечує комплексний, єдиний підхід до вирішення задачі.

2.3 Моделювання предметної галузі

Основна мета цієї інформаційної системи телеграм-боту для порівняння цін на цифрову техніку полягає у спрощенні пошуку найвигіднішої ціни серед продукції. Ця система має спрощувати та полегшувати процеси, що пов'язані з пошуком найкращої пропозиції на ринку. Головна мета полягає в підвищенні швидкості, легкості пошуку потрібної продукції за найкоротший проміжок часу.

На системній діаграмі верхнього рівня «Інформаційна система телеграм-боту для порівняння цін на цифрову техніку» представлено головний процес системи, який є центральним елементом теми проєктування. Основним об'єктом на цьому рівні є сама інформаційна система, що забезпечує функціонал для ефективного пошуку найкращої ціни за допомогою телеграм-боту. Також, визначаються споживачі результатів – користувачі системи, які мають доступ до функціоналу, учасники процесу – це сам телеграм-бот та об'єкти - на які спрямована дія системи.

Модель предметної області подано ієрархією об'єктно-процесних діаграм мовою моделювання ОРМ (рисунок 2.7-2.9).

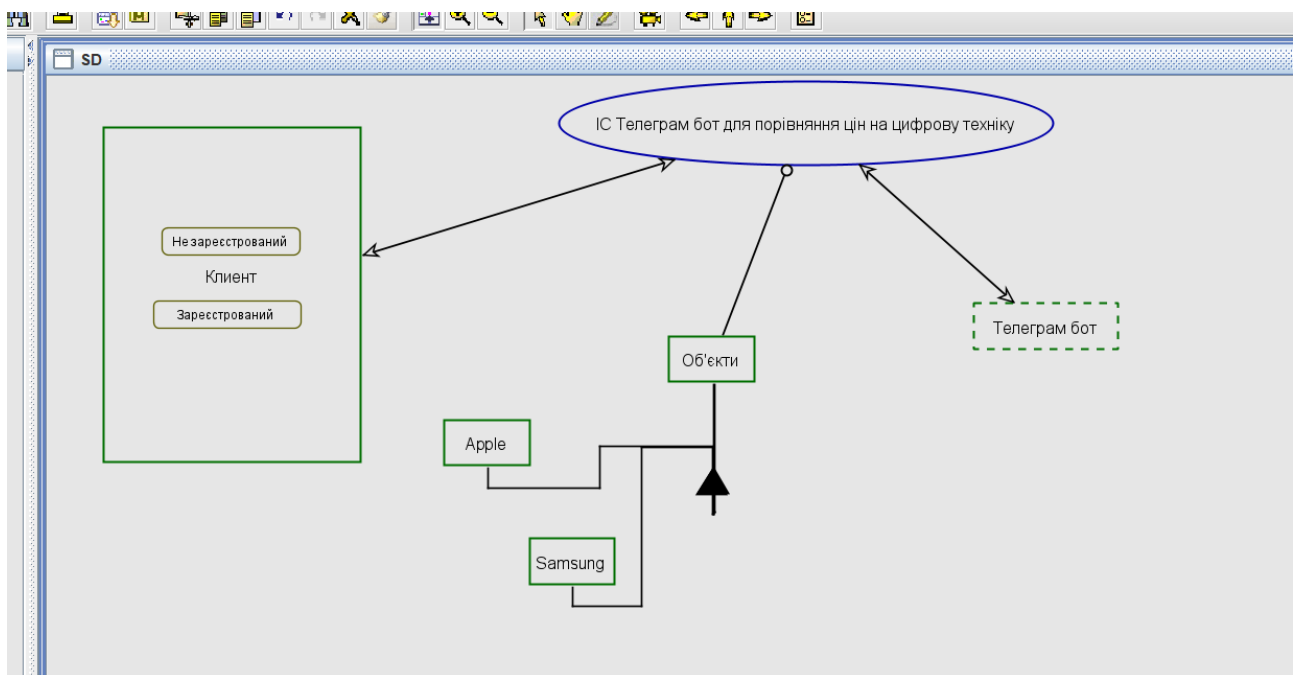


Рисунок 2.7 – Головна діаграма

На діаграмах наступного рівня деталізується функціонал системи через розгортання підпроцесів та асоційованих об'єктів. Підпроцес, "Пошук найкращої ціни на зацікавлену продукцію", визначає послідовність дій, які клієнт повинен виконати для того щоб отримати бажаний результат. В разі якщо порідного товару не знайшлося користувач може створити новий запит.

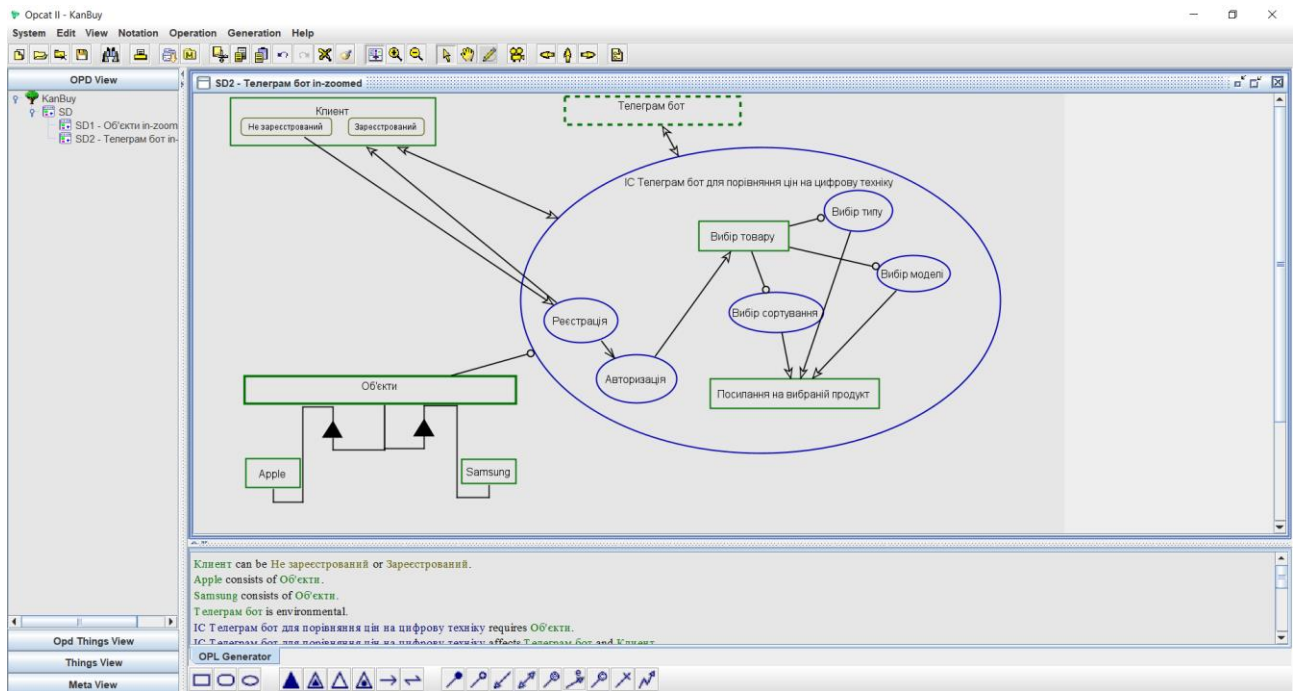


Рисунок 2.8 – Діаграма «Пошук найкращої ціни на зацікавлену продукцію in-zoomed»

На рисунку 2.9 за допомогою механізму збільшення більш детально показано входження конкретних об'єктів до телеграм-боту.

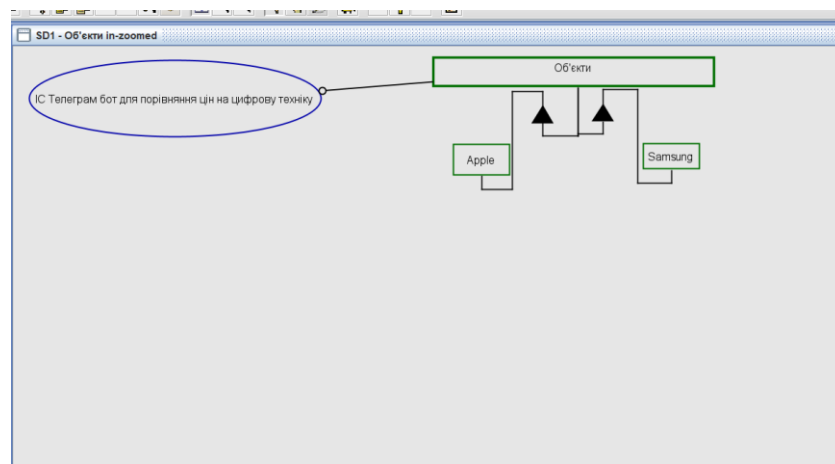


Рисунок 2.9 – Входження об'єктів до телеграм-боту

2.4 Постановка задачі

Основна мета проекту полягає в розробці комплексної інформаційної системи, призначеної для швидкого пошуку найкращої ціни на вибрану продукцію, за допомогою телеграм-боту. Система покликана пришвидшити пошук, мінімізувати затрати по часу пошуку потрібного товару.

Задача полягає у створенні багатфункціональної, зрозумілої, інформаційної системи, яка зможе задовольняти всі потреби сучасного покупця, який хоче придбати продукцію. Також важливо, щоб система була здатна адаптуватися до мінливих умов ринку та тенденцій серед діючих телегра-ботів, забезпечуючи при цьому швидкі відповіді на актуальні новинки ринку.

Проаналізувавши та використавши функціональні та нефункціональні вимоги дозволять нам створити ефективну та користувацько-орієнтовану інформаційну систему для порівняння цін на цифрову техніку, за допомогою телеграм-боту, який відповідатиме сучасним вимогам та тенденціям.

Вихідна інформація яка зберігається в базі даних. Вихідна інформація передбачає:

1. Структурування даних – база даних повинна мати чітко визначені таблиці для кожної категорії вихідної інформації, таких як сайти продукції, інформація про товари, ціни товарів. Використання реляційних зв'язків між таблицями для відображення взаємозв'язків між різними аспектами даних.
2. Ефективна нормалізація для забезпечення цілісності, зменшення дублювання даних та полегшення управління даними.
3. Індексція для оптимізації запитів – ключових стовпців для підвищення швидкості пошуку та витягування даних.

Розробка і впровадження інформаційної системи для пошуку найкращої ціни, за допомогою телеграм-боту є багатогранним процесом, що включає

детальне планування, глибокий аналіз потреб, розробку високої якості, та якісне тестування самої системи.

На рисунку 2.10 представлена інформаційна модель задачі, яка відображає взаємодію між різними компонентами інформаційної системи. Вона включає вузол для покупців, які можуть задати свої критерії для пошуку вигідної пропозиції при взаємодії з базою даних.

Система також включає функцію, таку як звітність(інформацію щодо товару), яка дозволить користувачам більш детально ознайомитись з інформацією та характеристиками товару.

Модель також відображає взаємодію з базою даних, яка включає таблиці для сайтів продукції, інформація про товари, ціни товарів. Система забезпечує різні форми вихідної інформації, що відповідають на запити клієнтів і дозволяють швидко знаходити потрібні сайти з продукцією.

Для описаної інформаційної моделі задачі використовуються різні способи подання інформації, зокрема екранні форми, електронні документи та файли, в даному саме випадку представлені саме екранні форми.

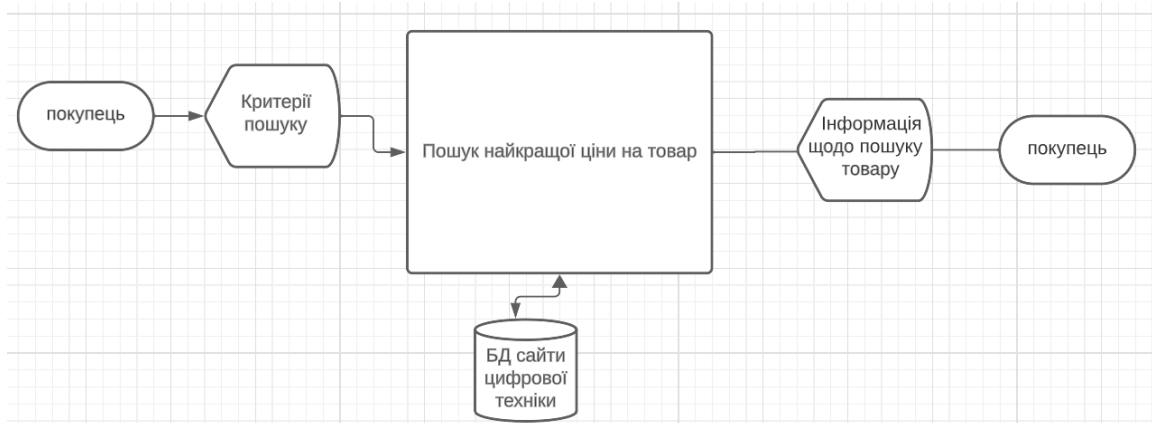


Рисунок 2.10 – Інформаційна модель задачі «Пошук найкращої ціни на товар»
«Критерії пошуку» представлені під таблицею «Вхідна інформація».

2.4.1 Вихідна інформація

Вихідна інформація використовується для отримання даних покупцем в залежності від його запиту. Вихідна інформація телеграм-боту складається з

основного повідомлення, зазначеного у таблиці 2.1 вихідна екранна форма наведена в Додатку Д рисунок Д-1.

Перелік і опис вихідних повідомлень

Таблиця 2.1

№ з/П	Назва вихідного повідомлення	Ідентифікатор	Форма подання і вимоги до неї	Періодичність видання	Термін видання і допустимий час затримки	Користувач інформації
1.	Інформація щодо пошуку товару	OUT001	Екранна форма	По запити	Миттєво	покупець

Структурні одиниці вихідного повідомлення OUT001:

- 1) компанія техніки яку вибрали;
- 2) модель вибраного товару;
- 3) фото товару;
- 4) посилання на сайти з продажу товару.

Таблиця містить повідомлення вихідної інформації, яка генерується інформаційною системою, екранна форма, в якій ця інформація подається та осіб, які її отримують.

Результати розв'язання задачі в інформаційній системі телеграм-боту для порівняння цін на цифрову техніку використовуються для оптимізації ключових бізнес-процесів телеграм-боту. Вони включають поліпшення швидкості пошуку потрібної продукції, різновид продукції, монетизація телеграм-боту та вихід в маси для заключення партнерства з брендами.

2.4.2 Вхідна інформація

В таблиці 2.2, зазначено вхідні екранні форми телеграм-боту «KanBuy», де покупець шукає найвигідніший товар за введеними критеріями пошуку взаємодіючий з базою даних, з якої береться інформація щодо моделей, марки, сайтів(посилання) та ціни продукта, які відображається при отриманні результату запиту.

Перелік і опис вхідних повідомлень

Таблиця 2.2

№ з/п	Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
1	Критерії пошуку	REQ001	Екранна форма	По запити	Покупець
2	Показники цифрової техніки	REQ002	БД	По запити	Сервер Телеграма

Структурні одиниці вхідного повідомлення REQ001:

1. компанія цифрової техніки;
2. категорій товарів цифрової техніки;
3. модель вибраного товару;
4. тип сортування;

БД сайтів цифрової техніки наведена в підрозділі 3.1.

Таблиця включає різні типи інформації, які система може отримувати через різні внутрішні процеси, уточнюючи метод подання та джерело вхідних даних.

2.5 Математичне забезпечення та алгоритм функціонування системи

2.5.1 Математичний опис

Розглянемо систематизований підхід до аналізу та оптимізації роботи телеграм-боту для порівняння цін на цифрову техніку.

Особливості та вимоги до реалізації телеграм-боту «Розробка Telegram-боту для порівняння цін на цифрову техніку» не передбачає значних математичних розрахунків, окрім формули розділення посилань сайтів з цифровою технікою, по кількості відображення в кожній з моделей, при використаному сортуванні.

Центральним елементом нашої моделі є розробка формул, що відображають взаємозв'язки між різними аспектами телеграм-боту.

Система відображення відсортованих сайтів, розраховується за допомогою формули наступним чином:

$$X = \left\lceil \frac{A_1}{3} \right\rceil$$

де: X – кількість сторінок відображення сайтів в боті, A_1 – кількість сайтів в БД;

3 – на початкову версію накладено обмеження кількості сайтів які закріплені за кожною з моделей (стале значення), якщо вказати більше, то будуть неполадки в роботі бота, бо через мірна кількість відображення сайтів перенавантажить систему й відображення буде не коректне, це може призвести до зупинки роботи бота;

Ця формула передбачає X – кількість сторінок які відображаються після сортування, шляхом ділення сайтів, що містяться в базі даних на кількість веб сторінок, що будуть розміщені під кожною моделлю, які відсортовані, це все відображено в Додатку Д рисунок Д-2. Квадратні дужки відображають округлення до більшого значення, оскільки, якщо кількість не ділитиметься на

З рівно, потрібно буде створити наступну вкладку з відображенням залишку сайтів, навіть якщо це лише один сайт. В такій ситуації, якщо значення буде рівне, наприклад: 3,5 , буде згенеровано 2 сторінки для відображення посилань на сайти.

2.5.2 Алгоритм розв’язання задачі

Алгоритм розв’язання задачі подано на рисунок 2.11.

Логічні переходи в системі функціонують як розгалуження, де на основі даних, введених покупцем, або його дій, визначається напрямок подальшої обробки запитів. Ці вузли прийняття рішень критично важливі для забезпечення коректної логіки роботи системи та правильного вибору функціоналу, який буде використовуватися відповідно до намірів покупця.

Процеси, такі як «вибір марки», «вибір гаджету», «вибір моделі», «вибір сортування», або «вибір сайту» представляють собою послідовність дій, які система виконує для обробки запитів покупця. Ці процеси включають в себе не тільки основні функції, але й додаткові операції, такі як продовження роботи з ботом, або припинення взаємодії, повернення до головного меню боту.

Бази даних є центральними елементами системи, що зберігає всю важливу інформацію, включаючи дані щодо сайтів, ціни та інформацію про товари.

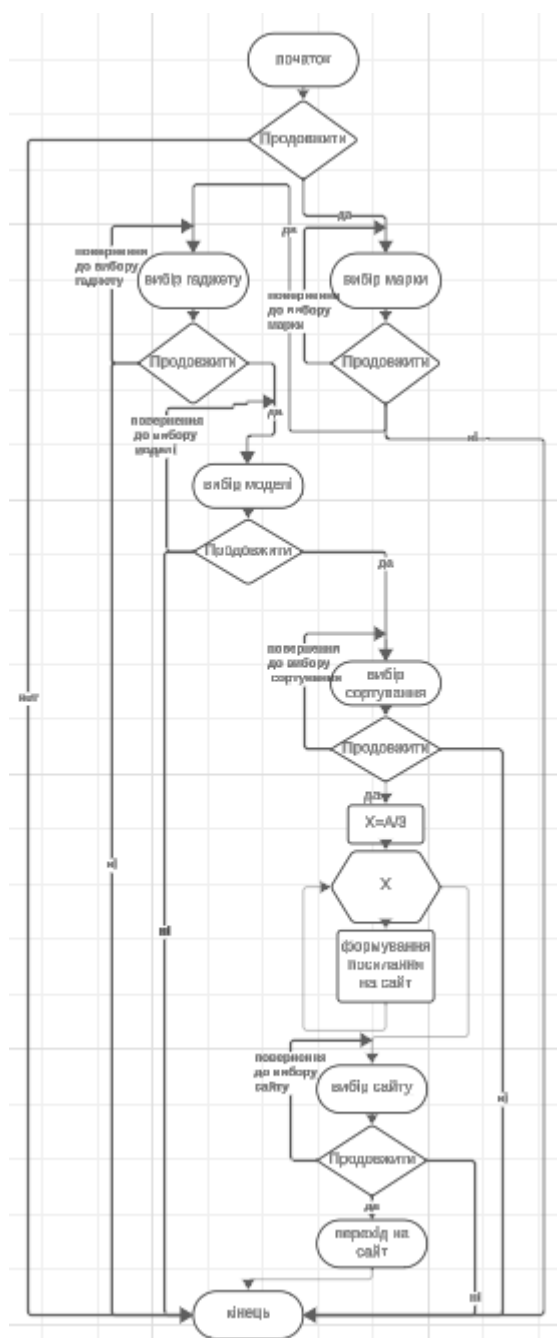


Рисунок 2.11 – Алгоритм розв’язання задачі з кількості відображення посилань сайтів на одній сторінці

2.6 Моделювання інформаційної системи

Моделювання поведінки інформаційної системи телеграм-боту включає взаємодію покупців з ботом, саму систему, обробку даних, поданням інформації в готовом відсортованом форматі.

Характеристика моделі поведінки системи для ІС «Телеграм-бот для порівняння цін на цифрову техніку»:

1. Створення запитів:

- можливість створювати запити зі вказаними фільтрами пошуку;
- покупці можуть знайти потрібний їм товар за вибраними власноруч фільтрами.

2. Перегляд результатів:

- Можливість переглядати отримані відповіді від поту, для кращого візуального розуміння, підходить товар за тим чи іншим параметром.

3. Зміна налаштувань системи:

- Адміністратор має можливість редагувати дані про новинки, змінювати послідовність товарів в боті.

4. Корекція в БД:

- адміністратор може змінювати дані в базі даних, видаляти, додавати та редагувати;
- мати повний доступ до корекцій бази даних.

Для роботи із ІС «Телеграм-бот для порівняння цін на цифрову техніку» можна виділити дві головні дійові особи:

- покупець (клієнт);
- адміністратор.

Взаємодія дійових осіб та функціональних можливостей системи відображена на діаграмі прецедентів (рисунок 2.12).

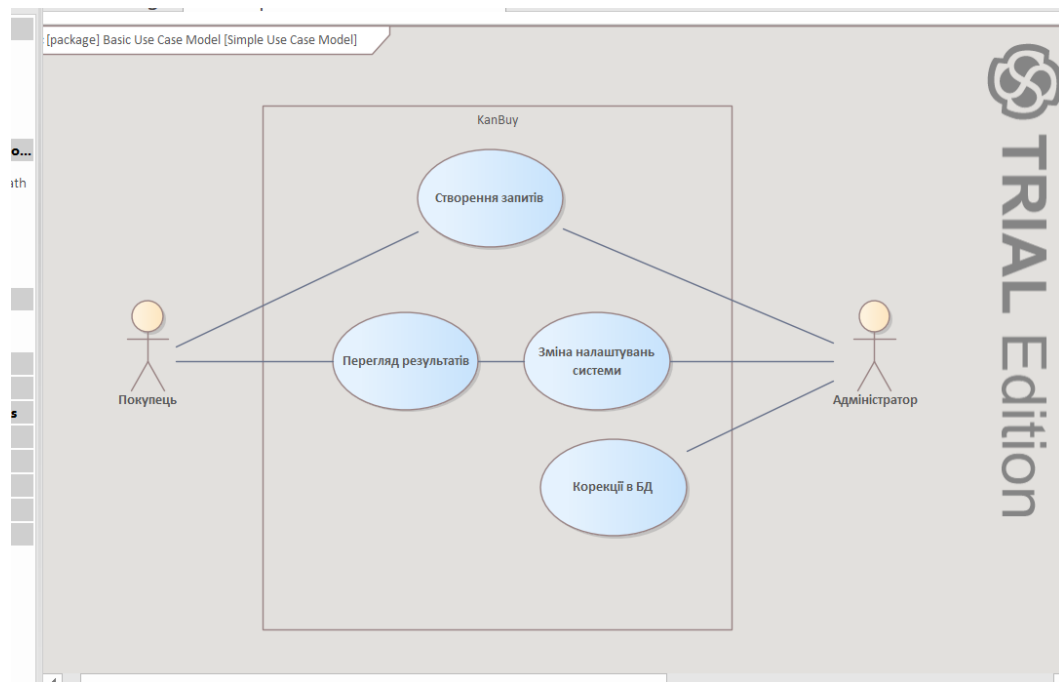


Рисунок 2.12 – Діаграма прецедентів

На діаграмі прецедентів (Use Case) для інформаційної системи телеграм-боту відображені ключові функціональні можливості системи та їх взаємодія з головними покупцями. Діаграма охоплює різні сценарії, які можуть відбуватися між користувачами (покупцем, адміністратором), включаючи створення запитів, перегляд результатів, зміна налаштувань системи та корекція БД.

Діаграма відображає різновид дій інформаційної системи. За допомогою телеграм-боту можна зрозуміти, як будуть сприйняті автоматизації бізнес-процесів та підвищенню ефективності телеграм-боту для порівняння цін на цифрову техніку, водночас забезпечуючи високий рівень задоволеності покупців через оптимізований інтерфейс та зручні функції.

На діаграмі послідовності для телеграм-боту, продемонстровано як об'єкти співпрацюють між собою в системі для пошуку найвигіднішої пропозиції(рисунок 2.13).

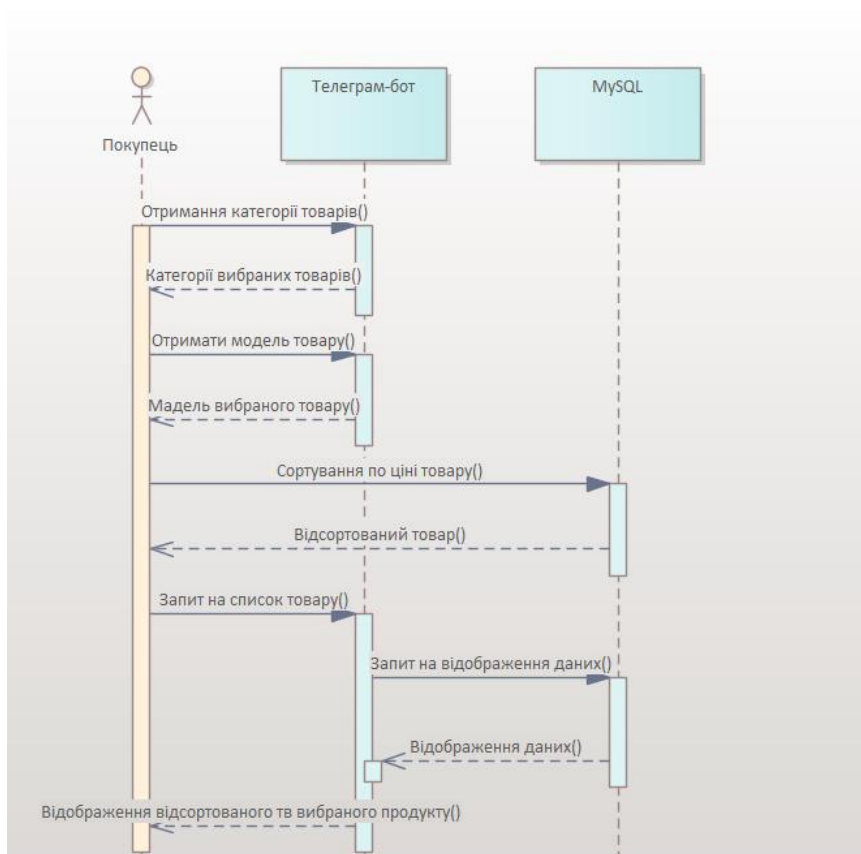


Рисунок 2.13 – Послідовність дій для пошуку найкращої ціни на вибрану продукцію

Вона включає такі елементи:

По-перше, актори – це зовнішні сутності, які взаємодіють з системою. На діаграмі є два актори: покупець та телеграм-бот.

Кроки діаграми:

1. Покупець надсилає запит до телеграм-боту на отримання категорій товару(марки компаній).
2. Телеграм-бот повертає список доступних категорій товару(марки компаній).
3. Покупець надсилає запит до телеграм-боту на отримання моделі товару.
4. Телеграм-бот повертає список доступних моделей в вибраній доступній категорії.

5. Покупець надсилає запит до телеграм-боту на сортування вибраного товару за ціною.
6. Телеграм-бот надсилає запит до бази даних на сортування вибраної продукції покупцем.
7. База даних повертає відсортовані дані щодо продукції до телеграм-боту.
8. Телеграм-бот відображає відсортовані дані щодо вибраної продукції покупцем.
9. Покупець отримує відсортовані дані щодо вибраної продукції з інформацією щодо неї.

Відображена комплексна взаємодія між компонентами телеграм-боту. При надсиланні запиту до бота, категорія товарів відображаються ті що містяться в середині системи разом з моделями товарів, а при запиті вже на сортування ціни, йде підключення та взаємодія з базою даних, яка сортує ціни на вибрану продукцію та повертає назад до телеграм-боту, де користувач вже бачить інформацію щодо товару, ціну, фотографію товару та відсортовані посилання на сайти, за ціною.

2.6.1 Моделювання структури системи

Головним архітектурним підходом, обраним для розробки системи, є трьохрівнева архітектура, яка базується на моделі MVC (Model-View-Controller)[12], що дозволяє розбити систему на три основні рівні: рівень представлення – це інтерфейс користувача, рівень бізнес-логіки та рівень доступу до даних. Цей підхід забезпечує чітке відокремлення логіки від інтерфейсу користувача та даних, сприяючи більшій організованості, гнучкості в діях та легкості управління з розширення системи.

На рівні представлення фокус зосереджений на розробці простого та зрозумілого інтерфейсу користувача, що включає обробку введення від користувача запитів та забезпечення зворотної відповіді.

Рівень бізнес-логіки відповідає за основну логіку та правила телеграм-боту, інтегруючи класи, які представляють різні сутності та компоненти бізнес-логіки системи.

Рівень доступу до даних забезпечує взаємодію з базою даних. Він містить класи для доступу та маніпуляції даними, гарантуючи ефективне та швидке управління даними.

У межах архітектури MVC також використовуються класи, що представляють компоненти моделі, вигляду та контролера. Це дозволяє розробити баланс між зручністю користувача, ефективністю обробки введених запитів та гнучкістю управління бізнес-процесами.[13] Така структура системи не тільки відповідає сучасним технічним стандартам, але й забезпечує адаптивність нашого телеграм боту до швидких сучасних змін.

Розглядаються окремі функціональні модулі, такі як модуль пошуку найвигіднішої пропозиції тощо. Для кожного модуля визначаються функції та взаємодія з іншими компонентами системи.

Інтерфейс користувача відіграє важливу роль у взаємодії між користувачами та інформаційною системою. Під час розробки інтерфейсу враховується різноманіття функціоналу телеграму, для створення бота. Основні моменти, які визначаються під час розробки інтерфейсу, включають дизайн, функціональність та зручність використання. Дизайн інтерфейсу має забезпечувати візуальну привабливість та комфортність використання для покупців. Зручність використання інтерфейсу полягає в його легкості та інтуїтивній зрозумілості, що дозволяє покупцям ефективно взаємодіяти з системою без зайвих зусиль.

В моєму проєкті використовується трьохрівнева архітектура (MVC-архітектура). В трьохрівневій архітектурі, система зазвичай розбивається на три рівні: рівень представлення - інтерфейс користувача, рівень бізнес-логіки та рівень доступу до даних. [14]

Також, наведено діаграму класів ІС «телеграм-боту для порівняння цін на цифрову техніку» спроектовану в EnterpriseArchitect (рисунок 2.15). В системі змодельовано повну структуру об'єктів, які в ній присутні, їхні взаємовідносини, властивості та операції тощо.

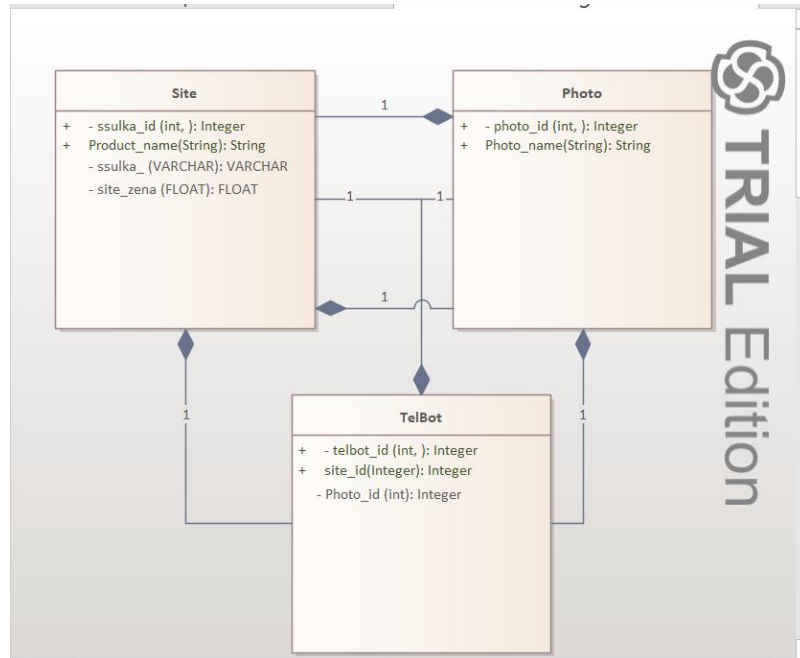


Рисунок 2.15 – Діаграма класів системи

Діаграма визначення блоків для відображає внутрішню структуру системи у вигляді ієрархії блоків різних типів відображає компоненти, які складають систему та їхні взаємозв'язки.[15] Для нашої інформаційної системи телеграм-боту така діаграма відображатиме різні рівні абстракції та компоненти, що складають систему. На верхньому рівні ієрархії блоків можуть бути такі компоненти, як "Презентація", "Бізнес-логіка", "Доступ до даних", "Система управління варіаціями дій". Кожен з цих блоків має свої власні залежності та підкомпоненти, що відображають структуру та взаємозв'язки частинами системи (рисунок 2.16).

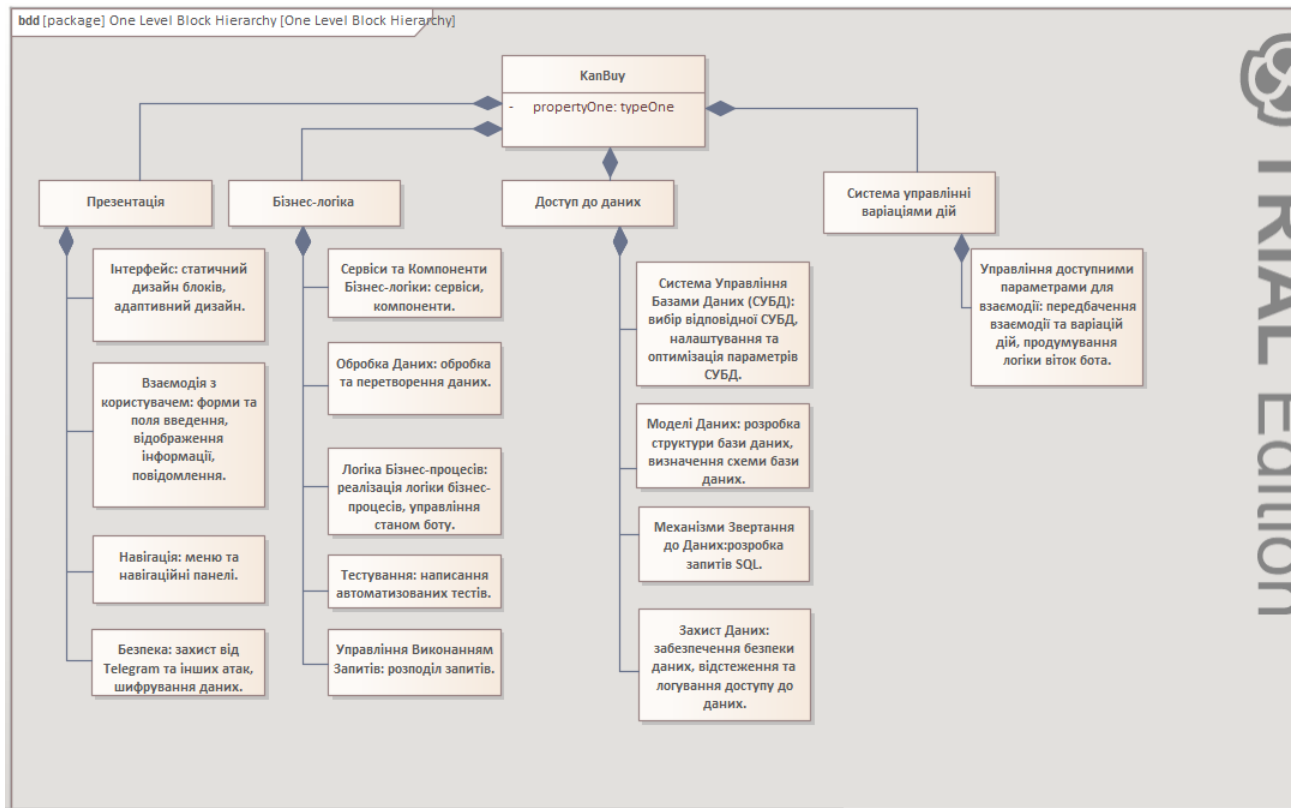


Рисунок 2.16 – Діаграма визначення блоків відображення внутрішньої структури системи

На рисунку 2.17 зображено приклад роботи API[16] компонент IC. На цій діаграмі показано фрагмент залежності компонентів системи, для пошуку найкращої ціни на обраний товар. Покупець обирає марку продукції, підтверджує свій вибір переходить до виду товару, де вибравши вид товару підтверджує його та обирає модель та тип сортування, після обрання цих параметрів отримує інформацію щодо товару в повному обсязі. Кожен з цих блоків може мати внутрішні компоненти, які відображають конкретні функціональні частини. Зв'язки між цими блоками можуть бути визначені через порти, які відповідають за обмін даними та виклики функцій між різними компонентами.

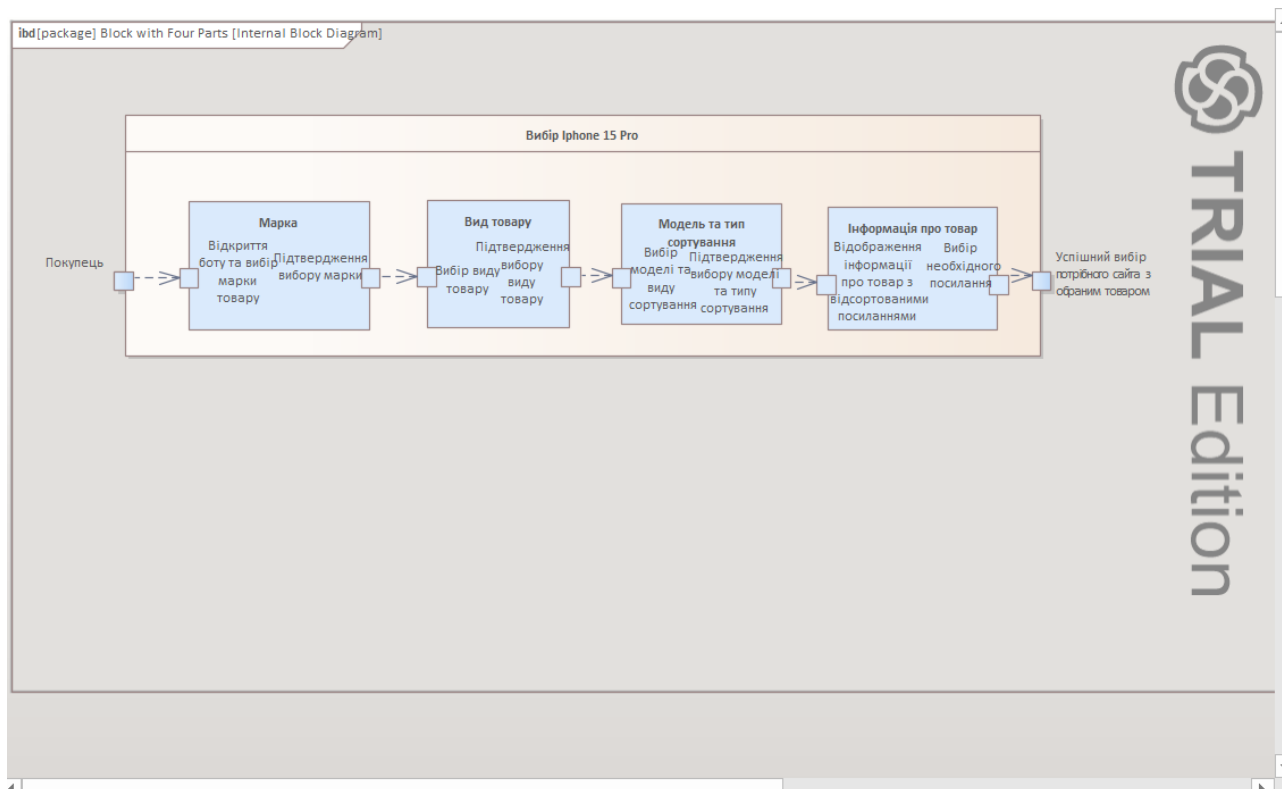


Рисунок 2.17 – Процес оформлення замовлення(IBD)

2.6.2 Моделювання користувацького інтерфейсу

На рисунках 2.18-2.23 зображено приклади користувацького інтерфейсу ІС, побудованого в Enterprise Architect. Наведено можливість швидкого вибору зацікавленої продукції.

1. Стартове вікно при початку взаємодії з ботом.

На рисунку 2.18 зображено стартова форма телеграм-боту.

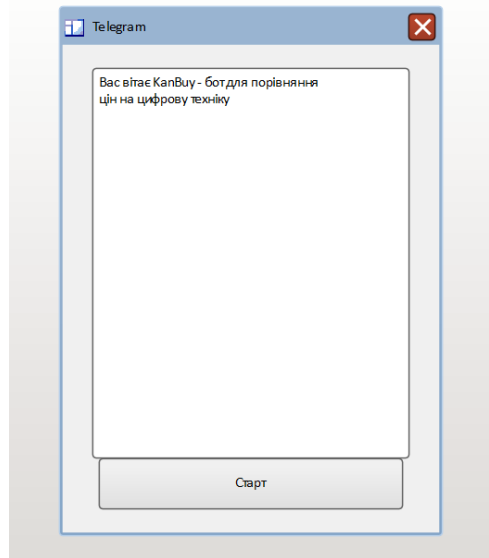


Рисунок 2.18 – Інтерфейс стартової форми телеграм-боту

2. Форма вибору марки продукції:

На рисунку 2.19 представлено прототип форми вибору марки товару, за допомогою цієї форми користувачі будуть здійснювати вибір потрібної їм продукції.

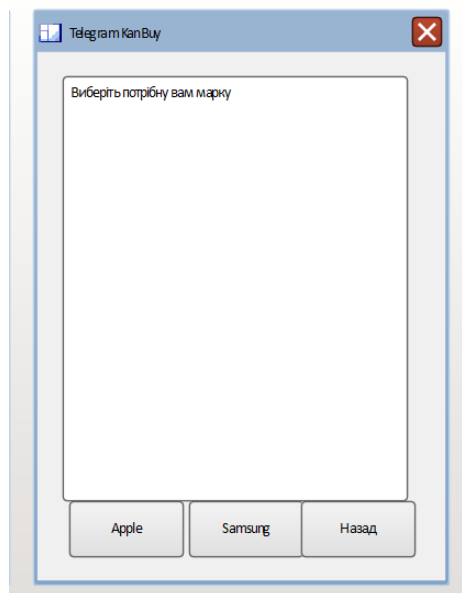


Рисунок 2.19 – Інтерфейс вибору марки товару

3. Вибір потрібного типу продукції:

На рисунку 2.20 продемонстровано прототип форми вибору типу продукції, за допомогою цієї форми можуть визначитись з тим що саме їм потрібно покупцеві.

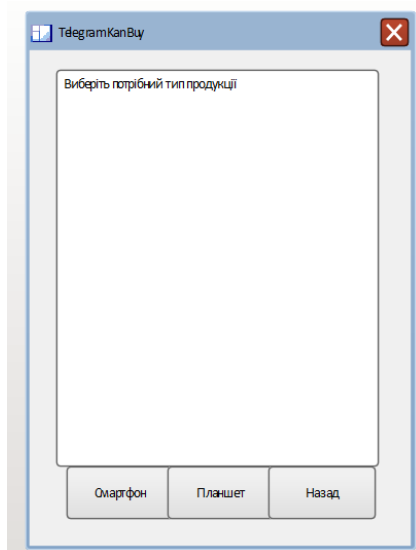


Рисунок 2.20 – Інтерфейс сторінки з функціоналом вибору типу продукції

4. Прототип для вибору моделі продукції:

На рисунку 2.21 продемонстровано прототип вибору моделі товару для покупки, на цій формі будуть відображатися різновид моделей продукції запропонованих для вибору покупцю.

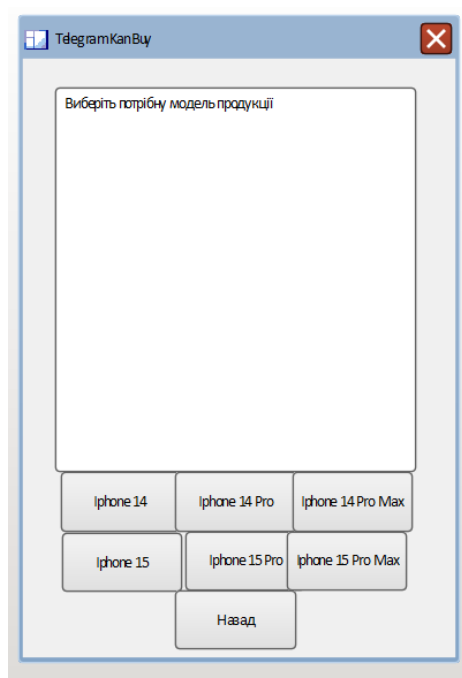


Рисунок 2.21 – Інтерфейс сторінки з функціоналом вибору моделі потрібної продукції

5. Прототип для вибору типу сортування вже вибраної продукції:

На рисунку 2.22 продемонстровано прототип вибору типу сортування для вибраної продукції, яку обрав покупець та хоче знайти найвигіднішу ціну на цю продукцію.

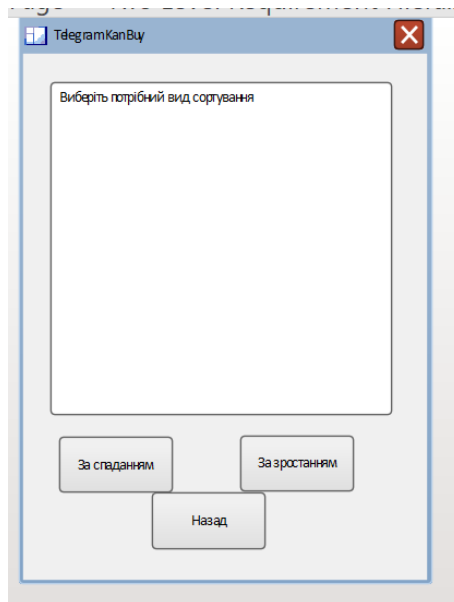


Рисунок 2.22 – Інтерфейс сторінки з функціоналом вибору типу сортування вибраної продукції

6. Прототип кінцевого результату щодо вибору товару покупцем:

На рисунку 2.23 продемонстровано прототип кінцевої відповіді бота на запит користувача з вибором потрібної йому продукції.

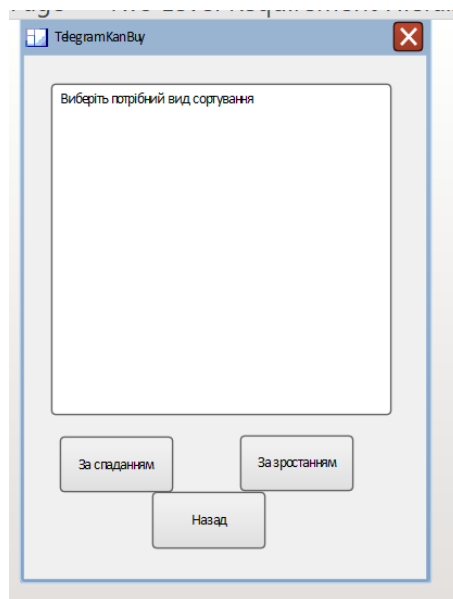


Рисунок 2.23 – Інтерфейс сторінки вибраної продукції відсортованої за ціною на сайтах

2.6.3 Прототипування користувацьких інтерфейсів

Для демонстрації користувацького інтерфейсу було створено прототип інформаційної системи у векторному онлайн-сервісі Figma[17]. На рисунок 2.24 представлено клікабельний прототип для показу повноти дій в телеграм-боті.

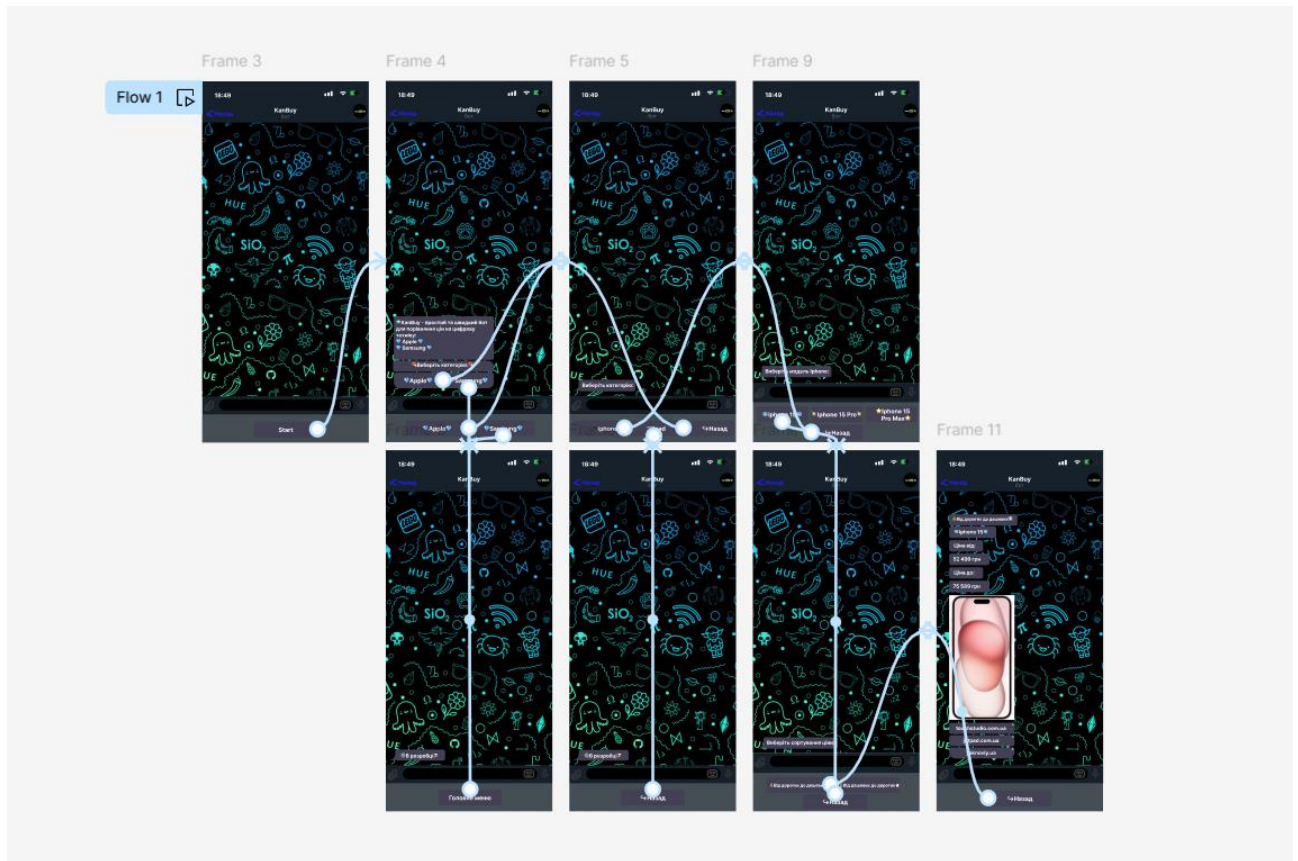


Рисунок 2.24 – Клікабельний прототип інформаційної системи «Телеграм-бот для порівняння цін на цифрову техніку» розроблений у Figma

Сторінки прототипу системи наведено у «Додатку В».

Інформаційна система для телеграм-боту має зрозумілий функціонал, представлений різноманітними сторінками та опціями, які дозволяють покупцям зручно та ефективно взаємодіяти з системою. Головне меню телеграм-боту призначене для вибору одразу марки продукції, щоб покупець міг одразу все вибрати що саме його цікавить. Після вибору марки перед покупцем відкривається вибір: типу пристрою, моделі та типу сортування сайті за ціною.

Після вибору необхідної марки продукції, покупець має змогу вибрати тип товару(планшет, телефон), після проходження кожного етапу він має змогу припинити роботу з ботом, або повернутись до головного меню.

Після обрання параметрів і натискання кнопки вибраного типу, покупець переходить на сторінку вибору моделі продукції, де фокусує свій взір на

цікавлячу модель. Тут він може побачити всю необхідну інформацію про моделі продукції вибравши одну з них.

Після вибору необхідної моделі користувач має змогу вибрати тип сортування сайтів з цією продукцією, при натисканні одного з двох варіантів, покупець отримує інформацію щодо продукції, мінімальну та максимальну ціну, фото гаджету та посилання на відсортовані сайти за ціною.

Також, був проведений аналіз узгодження з вимогами та функціонала системи (табл. 2.3).

Узгодження з вимогами та аналіз функціонала

Таблиця 2.3

Priority	User Story	Результат
1	Користувач системи (покупець), повинен мати можливість взаємодіяти з ботом.	Телеграм-бот має клікабельні кнопки з інформацією щодо дії, має можливість реагувати на запити користувача
2	Користувач системи (покупець), повинен мати можливість повертатись на крок назад	В дизайні та бекенді проекту був продуманий функціонал з поверненням на попередні сторінку боту
3	Користувач системи (покупець) повинен бачити інформацію про вибрану продукцію	В дизайні при виборі продукції продумано відображення інформації щодо продукції
4	Адміністратор, повинен мати можливість додавати, видаляти та редагувати інформацію в базі даних	Інформація в базі даних додається, редагується та видаляється
5	Користувач системи (покупець) повинен мати можливість переглядати фото продукції	Дизайн екрану для перегляду фото продукції. Клікабельні елементи для перегляду деталей.
6	Користувач системи (покупець) повинен мати можливість клікати на потрібний йому сайт та переходити за посиланням	В дизайні та бекенді створено можливість для переходу на потрібний сайт за посиланням
7	Користувач системи (покупець), повинен мати можливість в будь який час припинити роботу з ботом	Дизайн мість в собі моментальне припинення взаємодії з телеграм-ботом

Покупець може надсилати запити до бази даних на відображення інформації за допомогою телеграм-боту. Серед них є:

- запит на сортування ціни;
- запит на інформацію щодо продукцію;
- запит на відображення фото продукції;
- запит на вибір моделі продукції;
- запит на вибір типу гаджету;

- запит на повернення назад та відображення даних щодо попередньої сторінки;
- запит на відтворення посилання.

РОЗДІЛ 3

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1 Інформаційне забезпечення

Інформаційне забезпечення - це сукупність форм документів, нормативної бази та реалізованих рішень щодо обсягів, розміщення та форм існування інформації, яка використовується в інформаційній системі.[18]

Загальна схема інформаційного забезпечення наведена на рисунку 3.1.



Рисунок 3.1 – Загальна схема інформаційного забезпечення

При функціонуванні телеграм-боту використовуються такі елементи:

1. Вхідні повідомлення;
2. вихідні повідомлення;
3. БД.

Вхідні інформаційні повідомлення, що містять в собі: компанії цифрової техніки, тип цифрової техніки, модель, сортування.

Вихідні інформаційні повідомлення, що містять в собі: цифрової техніки, модель, фото товару, сайти з посиланнями на обраний продукт.

База даних, яка зберігається на зовнішньо-машинному ІБ, й містить в собі інформацію щодо посилань на сайти з продукцією, ціну товару та назву товару.

Перелік конкретних елементів ІЗ для функціонування бота:

1.1. Система запитів та відповідей:

Категорії продуктів: створення та управління категоріями товарів.

Запити користувачів: формування запиту та подальша обробка їх ботом.

Відповіді бота: зберігання стандартних відповідей на запити користувачів.

1.2. Інформація про товари та ціни:

База даних товарів: детальна інформація про кожний товар (модель, технічні характеристики, ціна).

Для створення бази даних системи було обрано MySQL.

Одним з головних видів програмного забезпечення, яке було використано нами – є MySQL є однією з найпопулярніших систем управління базами даних (СУБД) завдяки своїй високій продуктивності, надійності та легкості використання. Порівняно з іншими СУБД, такими як Oracle або Microsoft SQL Server, MySQL пропонує ряд переваг, особливо для проектів, які вимагають швидкої розробки та гнучкості, що є характерним для багатьох телеграм-ботів.

Перш за все, MySQL оптимізована для високої швидкості роботи та надійного зберігання даних, що робить її ідеальною для додатку з великою кількістю сайтів та даних про моделі та ціни, як телеграм-бот. Її відкритий вихідний код дозволяє розробникам налаштовувати і оптимізувати додаток під конкретні потреби проекту.[19]

Коли MySQL використовується у зв'язці з Python та середовищем розробки IntelliJ PyCharm, вона надає розробникам могутній інструментарій для створення, тестування та управління базами даних. Інтеграція з Python є безшовною завдяки API, який дозволяє Python – додаткам взаємодіяти з базою даних для виконання запитів, оновлень та отримання даних.[20][21]

Модель бази даних продемонстрована на рисунок 3.2 в середовищі MySQL Workbench.

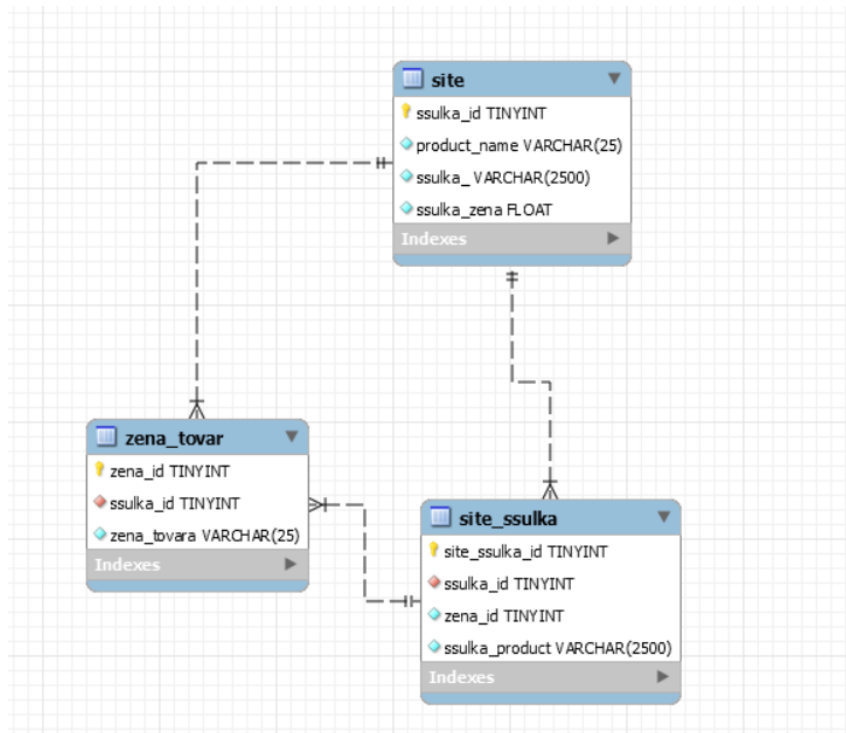


Рисунок 3.2 – EER-діаграма побудована в MySQL Workbench

Отриману базу даних наведено на рисунок 3.3.

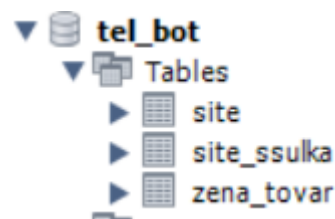


Рисунок 3.3 – База даних побудована в MySQL Workbench

Результат перетворення ER-моделі в конкретну схему бази даних на основі СУБД MySQL та вміст файлу згенерованої моделі даних представлено у «Додатку Г».

Специфікація вхідних та вихідних даних інформаційної системи «Телеграм-боту для порівняння цін на цифрову техніку» представлена на рисунок 3.4-3.6.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ssulka_id	TINYINT	☒	☒	☐	☐	☐	☐	☒	☐	
product_name	VARCHAR(25)	☐	☒	☐	☐	☐	☐	☐	☐	
ssulka_	VARCHAR(2500)	☐	☒	☐	☐	☐	☐	☐	☐	
ssulka_zena	FLOAT	☐	☒	☐	☐	☐	☐	☐	☐	

Рисунок 3.4 – Специфікація таблиці «Site»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
site_ssulka_id	TINYINT	☒	☒	☐	☐	☐	☐	☒	☐	
ssulka_id	TINYINT	☐	☒	☐	☐	☐	☐	☐	☐	
zena_id	TINYINT	☐	☒	☐	☐	☐	☐	☐	☐	
ssulka_product	VARCHAR(2500)	☐	☒	☐	☐	☐	☐	☐	☐	

Рисунок 3.5 – Специфікація таблиці «Site_ssulka»

Comments:										
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
zena_id	TINYINT	☒	☒	☐	☐	☐	☐	☒	☐	
ssulka_id	TINYINT	☐	☒	☐	☐	☐	☐	☐	☐	
zena_tovara	VARCHAR(25)	☐	☒	☐	☐	☐	☐	☐	☐	

Рисунок 3.6 – Специфікація таблиці «Zena_tovar»

3.2 Організаційне забезпечення

Далі на рисунку 3.7 продемонстрована діаграма станів. Діаграма станів допомагає визначити логіку системи і відслідковувати її реакцію на різні події.

Спочатку додаток перебуває в стані idle (бездіяльності), поки користувач не почне взаємодіяти з ним. Після того, переходить до стану «Активний», де проходить створення запиту з використанням додатку, після чого він переходить в стан перевірки «коректності» поданого запиту. Якщо все вірно, то додаток здійснює запит користувача. У стані «Активний», додаток може отримати дві події: скасування запиту або підтвердження. У випадку скасування запиту, додаток повертається в стан idle. У випадку підтвердження, додаток переходить в стан виконання запиту, де вона завершує запит користувача і повертається в стан idle.

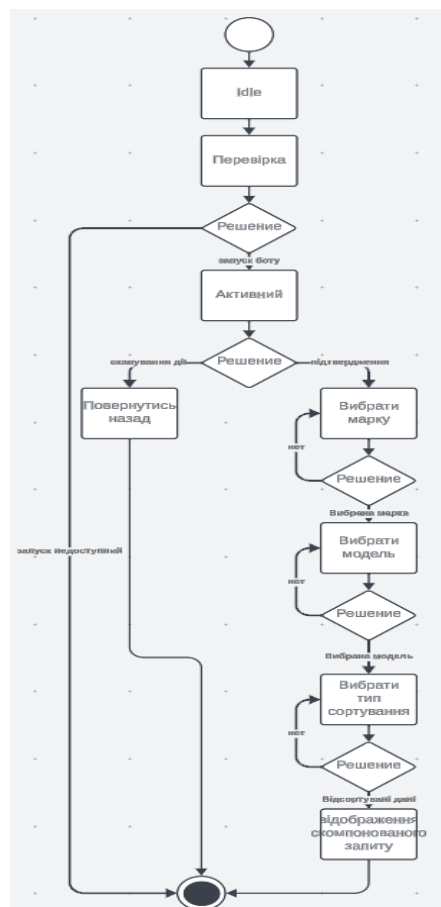


Рисунок 3.7 – Діаграма станів, процесу взаємодії додатку з користувачем «пошук найкращої ціни»

3.3 Програмне забезпечення

На рисунку 3.8 зображена діаграма компонентів системи телеграм-бота для порівняння цін на цифрову техніку первинний метод авторизації де виділено два основних блоки взаємодії: сторона адміністратора та сторона користувача.

Сторона адміністратора містить інструменти по редагування бази даних в ручному форматі: видаленню, додаванню, та заміні.

Сторона клієнта оснащена інтерфейсом, який надає клієнтам інтуїтивно зрозумілий доступ до різноманітних запитів та відображень інформації. Інтерфейс спрощує процес вибору моделі, сортування, марки та забезпечуючи високий рівень задоволеності клієнтів.

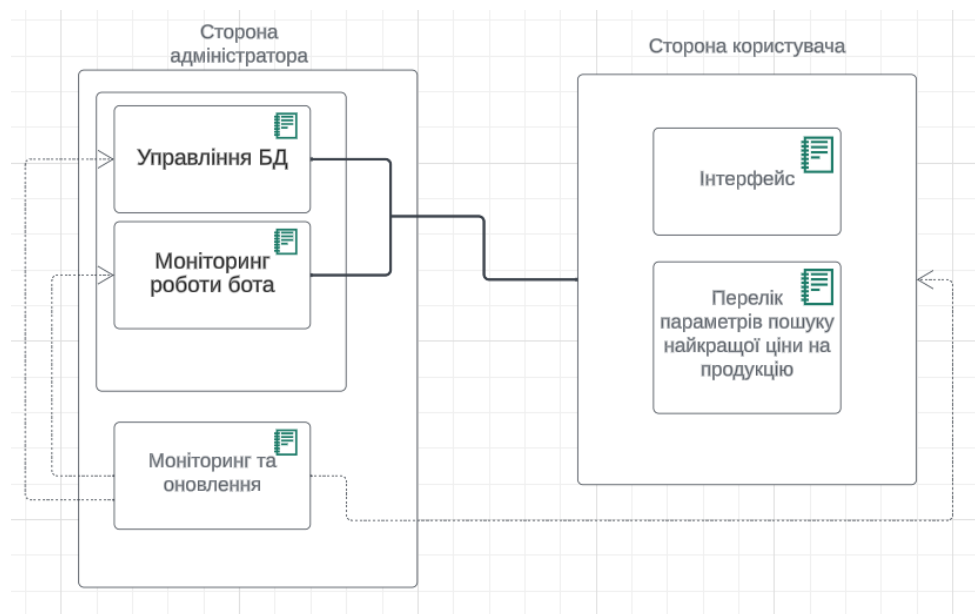


Рисунок 3.8 – Діаграма компонентів телеграм-бота – ілюструє компоненти апаратного та програмного вузла

В цілому, діаграма компонентів відображає взаємозв'язок та інтеграцію між різними компонентами додатку, підкреслюючи структуру та легкість розуміння для різних типів користувачів.[22] Чітке розмежування функціоналу між адміністративною та клієнтською частинами забезпечує ефективність управління телеграм-ботом для порівняння цін на цифрову техніку та високу якість обслуговування клієнтів.

3.4 Технічне забезпечення

На представленій діаграмі розгортання Telegram-bot порівняння цін на цифрову техніку (рисунок 3.9), видно, що вона складається з двох основних компонентів: Telegram Frontend і Host Backend, яке взаємодіє з MySQL сервером для зберігання даних.

Фронтенд-частина, відображається за допомогою самого телеграму, включає в себе різноманітні компоненти та сервіси, які забезпечують інтерфейс користувача. Компоненти Telegram відповідають за відображення інтерфейсу та взаємодію користувача з додатком, тоді як сервіси Telegram обробляють бізнес-логіку на стороні клієнта та комунікацію з бекендом.

Бекенд-частина, створена за допомогою Python, з використанням бібліотеки Telebot.

MySQL сервер, в свою чергу, зберігає базу даних для основних додатків з метою покращення бізнес-процесів та прийняття рішень. Така структура забезпечує міцну основу для створення масштабованого, ефективного та надійного додатку.

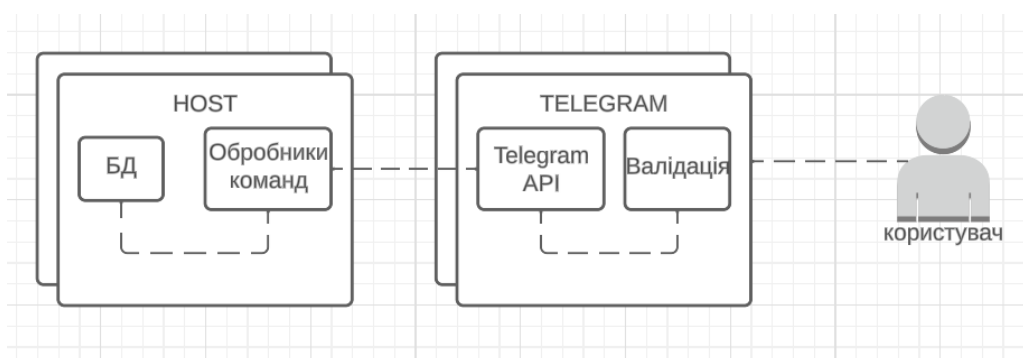


Рисунок 3.9 – Діаграма розгортання Телеграм-боту

В діаграмі розгортання наявні: БД(база даних де міститься інформація щодо сайтів, цін на техніку та назви товару), обробники команд(обробка запиту користувача командами які містяться в боті), Telegram API(використання самого телеграм як інтерфейс), валідація(підтвердження виконаної дії)

3.5 Результати реалізації інформаційної системи

В процесі виконання кваліфікаційної бакалаврської роботи було розроблено інформаційну систему, що призначена для пошуку найкращої ціни в телеграм-боті. ІС демонструє працездатність ключових функціональних модулів, що є основою для створення повноцінної системи більшого масштабу телеграм-боту.

В результаті реалізації ІС було досягнуто наступних результатів:

1. Функціональний модуль створення запитів:

Система дозволяє створювати запити, вибирати потрібні параметри та входження для пошуку, вибирати вже відсортовані сайти з товарами за певним типом. Реалізовано розмежування ролей користувачів: адміністратор має розширені права для управління системою, покупці – обмежені права для вибору товару.

2. Модуль зміни налаштувань системи:

Адміністратор має можливість корегувати послідовність відображення елементів в системі, змінювати налаштування та змінювати товари на актуальні новинки. Покупці можуть переглядати інформацію щодо актуальних новинок, та змін внесених в систему.

Модуль перегляду результатів:

Ключовий модуль системи дозволяє покупцям переглядати вже відфільтровану інформацію за їх запитом який був створений. Після вибору відповідного гаджету, покупець може вибрати тип сортування та отримати інформацію щодо товару уже в готовому вигляді.

Модуль корекції в БД:

Адміністратор має можливість додавати, видаляти та редагувати дані в базі даних, вносити зміни та корекції щодо інформації про товари. Покупці

можуть переглядати інформацію щодо товару, включаючи фотографії, характеристики, ціни.

Для перевірки працездатності системи було проведено тестування з використанням різних типів тестів, включаючи інтеграційні, системні та тести сценарії.

Тестування було проведено в середовищі PyCharm, що дозволило перевірити коректність роботи окремих модулів та системи в цілому. Результати тестування підтвердили відповідність системи поставленим вимогам та задачам.

ІС демонструє потенціал для створення більш масштабного телеграм-боту для порівняння цін на цифрову техніку..

Для подальшого розвитку системи розглядаються такі варіації:

Інтегрувати в систему рекламу товарів відомих марок, для їх популяризації.

Розширити функціонал системи, а саме різновид товарів та зробити широкий функціонал для пошуку найкращих цін будь чого. Це дозволить створити комплексне рішення для управління всіма аспектами телеграм-боту.

Впровадити розширену систему безпеки для захисту конфіденційної інформації. Це може включати двофакторну автентифікацію та інші механізми захисту.

Покращити інтерфейс покупців для забезпечення ще більш зручного та інтуїтивного використання системи, вдосконалення дизайну для кращого сприйняття.

РОЗДІЛ 4

ТЕСТУВАННЯ І ВЕРИФІКАЦІЯ ПРОЕКТНИХ РІШЕНЬ

4.1 Розробка тест-кейсів

Розробка тест-кейсів є важливою складовою процесу тестування будь-якого програмного забезпечення. Кожен вид тест-кейсів має свою функціональну спрямованість та сприяє виявленню певних помилок та недоліків в середині програмного забезпечення.[23]

Характеристика кожного виду тест-кейсів та їх використання у контексті проектування інформаційної системи телеграм-боту для порівняння цін на цифрову техніку:

1. Інтеграційні тест-кейси:

Назва: Перевірка взаємодії внесення змін до бази даних.

Опис: Перевіряє, чи правильно вносяться зміни до бази даних.

Призначення: Перевірка взаємодії з базою даних.

Використання: Виявлення помилок у взаємодії між компонентами системи та бази даних.

2. Системні тест-кейси:

Назва: Перевірка функціональності пошуку найкращої пропозиції.

Опис: Перевіряє, чи працює функція пошуку найкращої ціни, вибраної продукції за різними критеріями.

Призначення: Перевірка функціональності системи як цілісної єдності відповідно до вимог та очікувань покупців.

Використання: Встановлення того, чи відповідає програмне забезпечення вимогам та специфікаціям, а також чи забезпечує воно потрібну продуктивність системи.

3. Тести-сценарії:

Назва: Сценарій на перевірку відображення конкретного результату.

Опис: Симулює реальний процес пошуку найкращої ціни на вибрану продукцію з результатами.

Альтернативний сценарій:

Назва: Сценарій не реалізовується при взаємодії з ботом.

Опис: Симулює випадок, коли кнопки в середині системи не є функціональними.

Призначення: Тестування поведінки системи в конкретних сценаріях використання.

Використання: Перевірка роботи системи в реальних ситуаціях, включаючи різноманітні дії та взаємодію з покупцями.

У «Додатку А» наведено тестова документація, підготовлена засобами Enterprise Architect.

4.2 Трасування вимог до системи

Трасування вимог є важливим кроком у процесі розробки програмного забезпечення. Потрібно для визначення взаємозв'язків між тестовими кейсами, вимогами, прецедентами та елементами користувацького інтерфейсу і допомагає гарантувати, що реалізація функціональності програмного забезпечення є повною і відповідає вимогам користувачів та бізнес-цілям.[24]

Трасування вимог – це процес відстеження вимог від початку до реалізації та тестування. В інформаційній системі телеграм-боту для порівняння цін на цифрову техніку трасування системних вимог допомагає забезпеченню відповідності функціональності системи потребам покупців і бізнес-цілям проекту.

Характеристика використання діаграм відстеження та матриць взаємозв'язків для нашої ІС:

Діаграма трасування дозволяє візуалізувати функціональність системи і як кожна вимога впливає на неї та взаємозв'язок між кожною вимогою.[25] Наприклад, вона може показати, як прецедент «Створення запитів» виконує вимоги «Пошукова функція», «Запит вибору виду товару», «Запит вибору марки товару», «Запит вибору виду сортування товару» та «Запит вибору моделі товару» і які тести необхідно виконати для перевірки цієї функціональності.

Матриці зв'язків надають додаткову інформацію про взаємозв'язки між різними елементами в різних частинах системи. Вони використовуються, для демонстрації відповідності між прецедентами і вимогами, тестовими кейсами і вимогами, а також вимогами і елементами інтерфейсу.[26] Ці матриці можна використовувати для кращого розуміння як вимога відповідає і взаємодіє з різними аспектами системи.

Трасування вимог системи допомагає забезпечити високу якість інформаційної системи та відповідність до її функціональності потребам покупців та бізнесу.

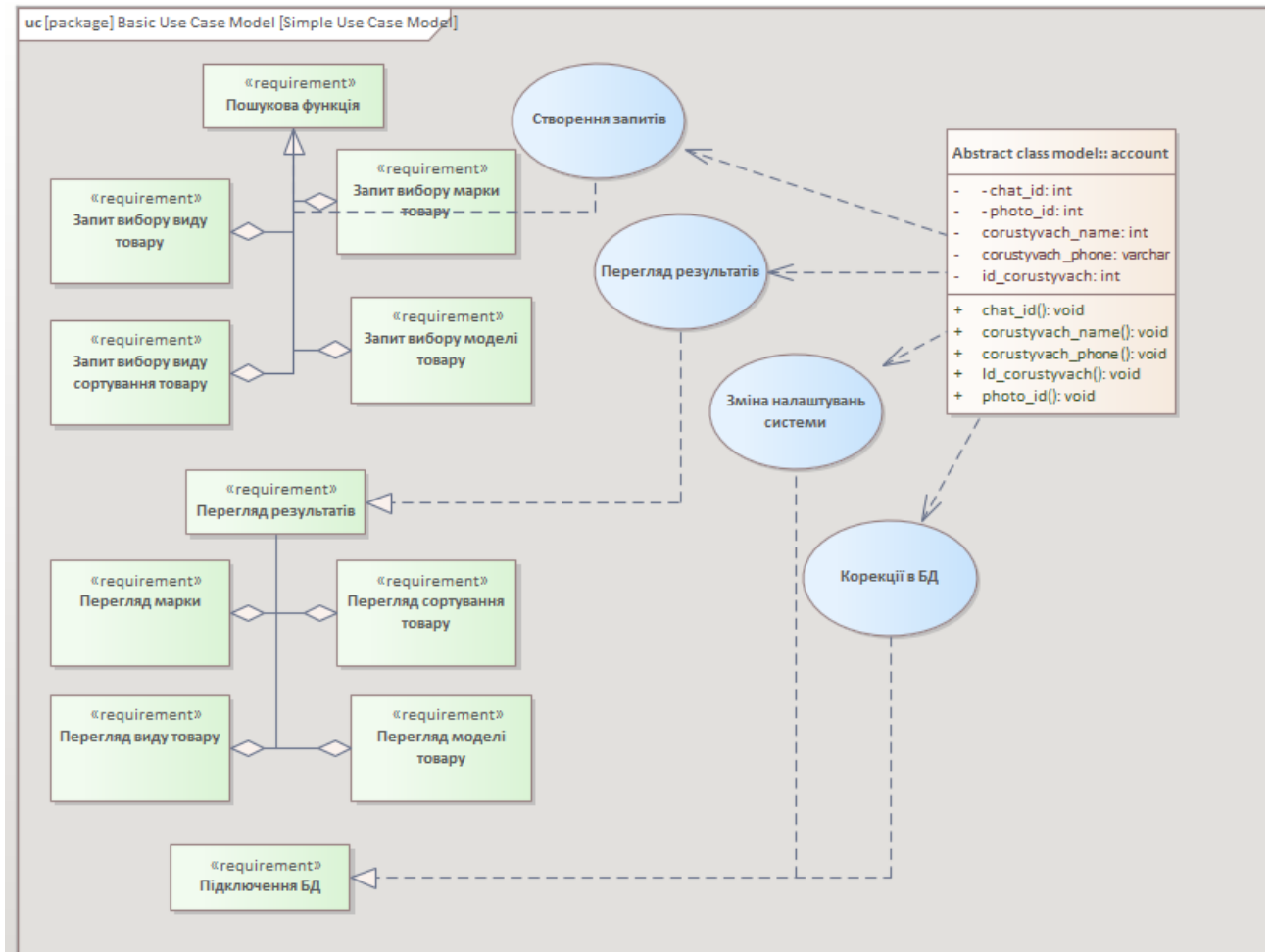


Рисунок 4.1 – Діаграма трасування в системі

Source: Basic Use Case

Set

Type <All>

Link Type: Realization

Profile:

Refresh

Target: Basic Use Case

Set

Type <All>

Direction: Source -> Target

Overlays: <None>

Options

Target +	Actors:Actor A	Actors:Actor C	Actors:System Actor E	Actors:Адміністратор	Actors:Покупець	Basic Use Case Model:Abstract	Basic Use Case Model:Actors	Basic Use Case Model:KanBuy	Basic Use Case Model:Package1	Basic Use Case Model:SV-5: Fur	Basic Use Case Model:Use Case	Basic Use Case Model:Запит ви	Basic Use Case Model:Запит ви	Basic Use Case Model:Запит ви	Basic Use Case Model:Запит ви	Basic Use Case Model:Перегляд	Basic Use Case Model:Перегляд	Basic Use Case Model:Перегляд	Basic Use Case Model:Перегляд	Basic Use Case Model:Підключ	Basic Use Case Model:Пошукує	SV-5: Function to Operational A	SV-5: Function to Operational A	Use Cases:Use Case D
+ Source																								
Basic Use Case Model::Пер...																								
Basic Use Case Model::Пер...																								
Basic Use Case Model::Пер...																								
Basic Use Case Model::Підк...																								
Basic Use Case Model::По...																								
SV-5: Function to Operatio...																								
SV-5: Function to Operatio...																								
Use Cases:Use Case D																								
Use Cases:Зміна налашту...																				↑				
Use Cases:Корекції в БД																				↑				
Use Cases:Перегляд резул...																		↑						
Use Cases:Створення зап...																					↑			

Рисунок 4.2 – Матриця взаємозв'язків в системі «Тгасе»

Relationship Matrix	
Relationships between Basic Use Case Model and Basic Use Case Model	
Source: Basic Use Case Model	Set Type <All>
Target: Basic Use Case Model	Set Type <All>
Link Type: Dependency	
Direction: Source -> Target	
Target +	Actors: Actor A
+ Source	Actors: Actor C
Actors: System Actor E	Actors: Адміністратор
Actors: Покупець	Actors: Покупець
Basic Use Case Model: Abstract	Basic Use Case Model: Abstract
Basic Use Case Model: Actors	Basic Use Case Model: Actors
Basic Use Case Model: KanBuy	Basic Use Case Model: KanBuy
Basic Use Case Model: Package1	Basic Use Case Model: Package1
Basic Use Case Model: SV-5: Fur	Basic Use Case Model: SV-5: Fur
Basic Use Case Model: Use Case	Basic Use Case Model: Use Case
Basic Use Case Model: Запит ви	Basic Use Case Model: Запит ви
Basic Use Case Model: Запит ви	Basic Use Case Model: Запит ви
Basic Use Case Model: Запит ви	Basic Use Case Model: Запит ви
Basic Use Case Model: Запит ви	Basic Use Case Model: Запит ви
Basic Use Case Model: Перегля	Basic Use Case Model: Перегля
Basic Use Case Model: Перегля	Basic Use Case Model: Перегля
Basic Use Case Model: Перегля	Basic Use Case Model: Перегля
Basic Use Case Model: Перегля	Basic Use Case Model: Перегля
Basic Use Case Model: Підключ	Basic Use Case Model: Підключ
Basic Use Case Model: Пошук	Basic Use Case Model: Пошук
SV-5: Function to Operational A	SV-5: Function to Operational A
SV-5: Function to Operational A	SV-5: Function to Operational A
Use Cases: Use Case D	Use Cases: Use Case D
Use Cases: Зміна налаштувань	Use Cases: Зміна налаштувань
Use Cases: Корекції в БД	Use Cases: Корекції в БД
Use Cases: Перегляд результат	Use Cases: Перегляд результат
Use Cases: Створення запитів	Use Cases: Створення запитів
Actors: System Actor E	
Actors: Адміністратор	
Actors: Покупець	
Basic Use Case Model: Abst...	
Basic Use Case Model: Act...	
Basic Use Case Model: Kan...	
Basic Use Case Model: Pack...	
Basic Use Case Model: SV-...	
Basic Use Case Model: Use ...	
Basic Use Case Model: Зап...	
Basic Use Case Model: Зап...	
Basic Use Case Model: Зап...	
Basic Use Case Model: Зап...	
Basic Use Case Model: Пер...	
Basic Use Case Model: Пер...	
Basic Use Case Model: Пер...	
Basic Use Case Model: Пер...	
Basic Use Case Model: Пер...	
Basic Use Case Model: Підк...	
Basic Use Case Model: По...	
SV-5: Function to Operatio...	

Рисунок 4.3 – Матриця взаємозв'язків в системі «Трасе»

ВИСНОВКИ

Кваліфікаційна бакалаврська робота присвячена актуальній проблемі сучасного пошуку найвигідніших цін на продукцію в телеграм-боті – швидкість пошуку та якість результатів відсортованої інформації щодо продукції. В роботі проведено комплексне дослідження предметної області, детальний аналіз існуючих рішень та розроблено концепцію інформаційної системи, що відповідає сучасним вимогам.

У першому розділі детально розглянуто особливості функціонування сучасних телеграм-ботів, включаючи процеси пошуку за цінами на продукцію, сортування сайтів з продукцією та різноманіття функціональності ботів. Проведено порівняльний аналіз популярних ІС, таких як Amazon Search & Pricex, GHop_bot, Volunteer Support. Виявлено їх сильні та слабкі сторони, що дозволило сформулювати вимоги до власної роботи, фокусуючись на ефективності управління та гнучкості. На основі проведеного аналізу сформульовано принципи розробки ІС, що включають використання трьохрівневої архітектури (MVC), об'єктно-орієнтованого підходу (ООП), мови програмування Python, бази даних MySQL та сучасних фронтенд-технології Телеграм.

В другому розділі детально описуються функціональні, нефункціональні та бізнес-вимоги до системи. Також, увагу приділено аспектам зручності використання, масштабованості, можливостям звітності та аналітики. Для реалізації проєкту вибрано методологію Agile яка базується на гнучкості та ітеративності, що дозволяє внести зміни до вимог на будь-якому етапі розробки, що особливо важливо для проєкту, де інформація про товари може часто змінюватися. За допомогою середовища OPCAT було створено модель ІС, а саме ієрархії OPD-діаграм – повний набір процесів і об'єктів разом з їхніми станами для ІС. Проаналізувавши всі вимоги та бізнес-цілі, створено інформаційну модель задачі, що відображає структуру даних, у вигляді таблиць

вхідні та вихідні повідомлення, а також взаємодію між різними компонентами системи. Наступним кроком було розроблено математичне забезпечення та спроектований правильним чином алгоритм функціонування системи, на якому продемонстровано кожний крок покупця, адміністратора телеграм боту. Розроблено діаграму прецедентів (Use case), представлено матриці відповідності вимог і прецедентів, діаграми послідовності (Sequence), класів (Class), визначення блоків для подання внутрішньої структури системи (BDD), процес оформлення замовлення (IBD), які детально описують функціональність системи, взаємодію між користувачами та системою.

Змодельовано приклади користувацького інтерфейсу ІС, які створені в Enterprise Architect та клікабельний прототип інтерфейсу користувача у Figma, що дозволяє наглядно продемонструвати основні функціональні можливості. Сторінки прототипу ІС продемонстровано в Додатку В.

У третьому розділі продемонстровано спроектовану базу даних, а саме структуру бази даних, ключові таблиці, зв'язки між ними. Використання MySQL Workbench забезпечує зручний інструмент для проєктування та управління базою даних. Створено діаграму компонентів, що ілюструє взаємодію між модулями системи, включаючи фронтенд та бекенд компоненти. Також, розроблено діаграму розгортання, що відображає фізичну архітектуру системи. Представлено скріншоти інтерфейсу покупця та результати роботи основних функцій системи.

В четвертому розділі розроблено різні типи тест-кейсів (модульні, системні, тести-сценарії), що охоплюють функціональні можливості системи. Продемонстровано діаграми трасування та матриці зв'язків для перевірки що відповідають поставленим на початку вимогам.

В ході виконання кваліфікаційної бакалаврської роботи було розроблено комплексну інформаційну систему телеграм-боту для порівняння цін на цифрову техніку, яка відповідає сучасним вимогам до функціональності та

зручності використання. Система надає можливості для ефективного пошуку найкращої ціни на продукцію.

Розроблений користувацький інтерфейс є інтуїтивно зрозумілим та зручним у використанні. Його дизайн базується на принципах юзабіліті, що забезпечує легкість та швидку взаємодію з системою. Важливим аспектом проекту є його масштабованість. Завдяки використанню модульної архітектури, систему можна легко збільшувати та додавати все більше та більше функціоналу.

Результати кваліфікаційної бакалаврської роботи можуть бути використані для створення ще більш масштабного телеграм-боту, який буде наділений властивостями нашої системи. Розроблені моделі та алгоритми можуть слугувати базою для створення більш складних та спеціалізованих ІС. Також, робота демонструє поєднання теоретичних знань та практичних навичок у розробці інформаційної системи телеграм-ботів сучасності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1) Структура телеграм-боту, [Online]. Режим доступу: <https://habr.com/ru/company/leader-id/blog/479562/>.
- 2) AmazonGlobalBot телеграм-бот [Електронний ресурс] – Режим доступу до ресурсу: <https://t.me/AmazonGlobalBot>.
- 3) GHop_Bot телеграм-бот [Електронний ресурс] – Режим доступу до ресурсу: https://t.me/GHop_Bot.
- 4) Volunteer Support телеграм-бот [Електронний ресурс] – Режим доступу до ресурсу: https://t.me/volunteersupporthelp_bot?do=open_link.
- 5) Agile, [Online]. Режим доступу: <https://asana.com/ru/resources/agile-methodology>.
- 6) DAO, [Online]. Режим доступу: <https://academy.binance.com/uk/articles/decentralized-autonomous-organizations-daos-explained>.
- 7) PyCharm 2023.1.3, [Online]. Режим доступу: <https://blog.jetbrains.com/pycharm/2023/06/2023-1-3/>.
- 8) ООП, [Online]. Режим доступу: https://fis.tntu.edu.ua/data/elibrary/3/oop_cpp.pdf.
- 9) MySQL Workbench, [Online-стаття]. Режим доступу: https://kpfu.ru/staff_files/F1776545750/mysql.PDF.
- 10) Agile деталі методології, [Online-книга]. Режим доступу: <https://www.atlassian.com/ru/agile/project-management>.
- 11) ООП визначення, [Online-книга]. Режим доступу: https://balka-book.com/files/2023/04_11/11_28/u_files_store_5_37.pdf.
- 12) MVC (Model-View-Controller), [Online-стаття]. Режим доступу: <https://www.geeksforgeeks.org/mvc-design-pattern/>.

- 13) Підхід MVC (Model-View-Controller), [Online-стаття]. Режим доступу:
<https://javarush.com/ua/quests/lectures/ua.questservlets.level14.lecture02>.
- 14) Pro PHP 8 MVC: Model View Controller Architecture-Driven Application Development, [Книга].
- 15) Блок-діаграма, [Online-стаття]. Режим доступу:
<https://uk.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D0%BA-%D0%B4%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0#:~:text=Block%2DDiagramm%20n%20f%C3%BCr%20geologische,%D0%BF%D1%80%D0%BE%D1%94%D0%BA%D1%86%D1%96%D1%97>.
- 16) API, [Online-книга]. Режим доступу: <https://github.com/twirl/The-API-Book/blob/gh-pages/docs/API.ru.pdf>.
- 17) Figma, [Online]. Режим доступу:
<https://www.figma.com/community/file/1291305880342901188/17>.
- 18) Інформаційне забезпечення, [Online-стаття]. Режим доступу:
<https://ru.osvita.ua/vnz/reports/management/13966/>.
- 19) Бази даних MySQL. Навчальний посібник, [Книга].
- 20) Python, [Online]. Режим доступу: <https://uk.wikipedia.org/wiki/Python>.
- 21) Перспективи Python, [Online]. Режим доступу:
<https://beetroot.academy/blog/courses/chomu-tsogo-roku-tobi-varto-vivchiti-python>.
- 22) Діаграма компонентів, [Online-стаття]. Режим доступу:
http://www.znannya.org/?view=uml_component_diagram_10.
- 23) Тест-кейси, [Online-стаття]. Режим доступу:
<https://itvdn.com/ua/video/testing-process>.
- 24) Трасування вимог, [Online-стаття]. Режим доступу:
<https://studfile.net/preview/5740844/page:11/>.

- 25) Діаграма трасування, [Online-книга]. Режим доступу:
<https://mmi.stu.cn.ua/wp-content/uploads/2016/09/MatsenkoKompGrafyka.pdf>.
- 26) Матриці зв'язків, [Online-книга]. Режим доступу:
https://itknyga.com.ua/documents/OOP_final.pdf.

РЕЗУЛЬТАТИ АНТИПЛАГІАТУ



Имя пользователя:
Інформаційних систем в економіці Шкуратовська Те... ID проверки:
1016330597

Дата проверки:
07.06.2024 08:00:26 EEST Тип проверки:
Doc vs Internet + Library

Дата отчета:
07.06.2024 08:09:07 EEST ID пользователя:
100005745

Название файла: Кандала_В_ІН-403

Количество страниц: 76 Количество слов: 10061 Количество символов: 83533 Размер файла: 4.72 MB ID файла: 1016130178

26.5% Совпадения

Наибольшее совпадение: 22.2% с источником из Библиотеки (ID файла: 1016112527)

2.81% Источники из Интернета	275	 Страница 78
26.5% Источники из Библиотеки	523	 Страница 80

0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

0% Исключений

Нет исключенных источников

Модификации

Обнаружены модификации текста. Подробная информация доступна в онлайн-отчете.

Замененные символы 1

ДОДАТКИ

ДОДАТОК А

Тестова документація, підготовлена засобами Enterprise Architect

Тестова документація включає в себе плани тестування продуктивності та надійності, що дозволяє оцінювати, наскільки система відповідає вимогам щодо швидкодії та стабільності в реальних умовах експлуатації.

Документація містить у собі описи різних тестових сценаріїв, які покривають різні аспекти системи, включаючи функціональність, продуктивність тощо. Це допомагає забезпечити повноту тестування системи та виявити потенційні проблеми або недоліки ще на етапі проектування.

Виявлення помилок або недоліків за допомогою тестування дозволяє уникнути проблем на пізніших етапах розробки, що допомагає знизити ризики та витрати на виправлення дефектів.

А.1 Системні тести

Розроблені тестові звіти (Test Report):

The screenshot shows the 'Test details' dialog box. It has a title bar with a close button. The main area is divided into sections. The top section is labeled 'Test' and contains a text box with the description 'Тест кейс з створення стандартного запиту' and an 'Auto' button. Below this is a tabbed interface with tabs for 'Description', 'Input', 'Acceptance Criteria', and 'Results'. The 'Description' tab is active, showing a list of steps: 1) Зайти в бота, 2) Вибрати марку товару, 3) Обрати тип товару (категорію), 4) Вибрати модель товару, 5) Вибрати тип сортування товару, 6) Отримати результат з інформацією про товар, марку, тип, фотографію товару, та посиланням на відсортовані сайти. Below the description is a 'Class' dropdown set to 'Unit' and a 'Type' dropdown set to 'Standard'. The bottom section is labeled 'Run' and contains fields for 'Run Date' (14.04.2024), 'Run By' (Владос), 'Result' (Pass), and 'Checked By' (Владос). At the bottom of the dialog are buttons for '<<', '>>', 'New', 'Apply', 'OK', 'Close', and 'Help'.

Рисунок А.1 — Деталі тест-кейсу

Як бачимо виконання тест-кейсу передбачає 5 пунктів: зайти в бота, вибрати марку товару, обрати тип товару(категорію), вибрати модель товару, вибрати тип сортування товару. При позитивному сценарій буде відображення вибраної моделі та інформації щодо неї з підкріпленою фотографією та посиланнями на відсортовані сайти. При негативному сценарію форма не перейдить до жодного з наступних етапів та зупиниться. Також за допомогою засобів Enterprise Architect було згенеровано відповідну документацію по даному тест-кейсу представлену на рисунку А.2.

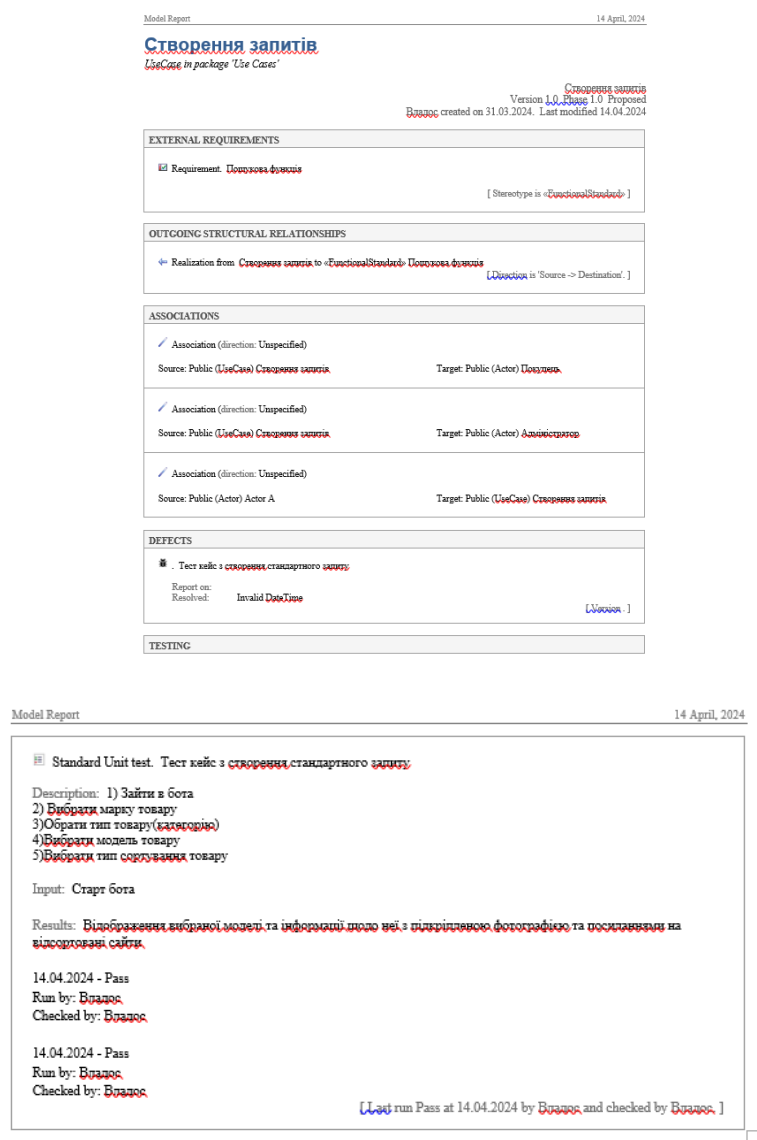


Рисунок А.2 — Документація тест-кейсу запиту користувача

A.2 Інтеграційні тести

Розроблені тестові звіти (Test Report):

The screenshot shows a 'Test details' dialog box with the following elements:

- Test:** Тест внесення корекцій до БД
- Description:** 1) відкрити БД
2) Відкрити таблицю в яку потрібно внести зміни
3) внесення інформації та корегування таблиці
- Class:** Unit
- Type:** Standard
- Run:**
 - Run Date:
 - Run By:
 - Result: Not Run
 - Checked By:
- Buttons:** <<, >>, New, Apply, OK, Close, Help

Рисунок А.3 — Деталі тест-кейсу для корегування БД

Як можемо бачити з рисунка А.3 для проведення даного тест-кейсу внесення змін до БД необхідно виконати наступні 3 кроки, а саме: відкрити БД, відкрити таблицю в яку потрібно внести зміни та внесення інформації з корегуванням таблиці. В результаті адміністратор повинен отримати готову таблицю зі зміненими даними. Також для даного тест-кейсу в Enterprise Architect було згенеровано відповідну документацію представлену на рисунку А.4.

Корекція в бд

Use Case in package 'Use Cases'

Корекція в бд

Version 1.0 Phase 1.0 Proposed

Владос created on 31.03.2024. Last modified 01.04.2024

+

EXTERNAL REQUIREMENTS

Requirement. Підключення БД

[Stereotype is «FunctionalStandard»]

+

OUTGOING STRUCTURAL RELATIONSHIPS

Realization from Корекція в бд to «FunctionalStandard» Підключення БД

[Direction is 'Source -> Destination'.]

+

ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Actor) Адміністратор

Target: Public (Use Case) Корекція в бд

Association (direction: Unspecified)

Source: Public (Actor) Actor C

Target: Public (Use Case) Корекція в бд

+

TESTING

Standard Unit test. Тест внесення корекцій до БД

Description: 1) відкрити БД
2) Відкрити таблицю в яку потрібно внести зміни
3) внесення інформації та корегування таблиці

Input: Старт

Results: Відкорегована база даних в новому форматі
14.04.2024 - Pass
Run by: Владос
Checked by: Владос

14.04.2024 - Pass
Run by: Владос
Checked by: Владос

[Last run Pass at 14.04.2024 by Владос and checked by Владос.]

Рисунок А.4 — Тест-кейс для створення нового запитання

А.3 Тести-сценарії

Розроблені тестові звіти (Test Report):

The screenshot shows a 'Test details' window with the following fields and controls:

- Test:** A text box containing 'Тест кейс на перевірку коректного відображення результатів' and an 'Auto' button.
- Description:** A tabbed interface with 'Description', 'Input', 'Acceptance Criteria', and 'Results' tabs. The 'Description' tab is active, showing a list of steps:
 - 1) Виконати запит покупця
 - 2) Натиснути кнопку сортувати
 - 3) Отримати результат з фотографією продукції та всією інформацією з посиланням на сайти де можна придбати цю продукцію
- Class:** A dropdown menu with 'Unit' selected.
- Type:** A dropdown menu with 'Standard' selected.
- Run:** A section with four fields:
 - Run Date:** A date picker showing '14.04.2024' with a checkmark icon.
 - Run By:** A dropdown menu with 'Владос' selected.
 - Result:** A dropdown menu with 'Pass' selected.
 - Checked By:** A dropdown menu with 'Владос' selected.
- Buttons:** A row of buttons at the bottom: '<<', '>>', 'New', 'Apply', 'OK', 'Close', and 'Help'.

Рисунок А.5 — Деталі тест-кейсу

Як можемо бачити з рисунка А.5 даний тест-кейс складається з 3 кроків: виконання запиту користувача, натискання кнопки сортування та отримання результату з фотографією продукції та всією інформацією з посиланням на сайти де можна придбати цю продукцію. При позитивному сценарію буде здійснено успішне відображення, де покупець зможе вибрати найбільш вигідніший йому сайт. При негативному сценарію форма не буде виконана, та не перейде до наступного пункту що відображати результати, покупець буде повернутий до початкового меню. Також до цього тест-кейсу було згенеровано відповідну документацію що представлена на рисунку А.6.

Перегляд результатів

Use Case in package 'Use Cases'

Перегляд результатів

Version 1.0, Phase 1.0 Proposed

Будує, created on 31.03.2024. Last modified 01.04.2024

EXTERNAL REQUIREMENTS
Requirement. Запит з вибору виду товару [Stereotype is «FunctionalStandard»]
Requirement. Запит з вибору марки товару [Stereotype is «FunctionalStandard»]
Requirement. Перегляд результатів [Stereotype is «FunctionalStandard»]
OUTGOING STRUCTURAL RELATIONSHIPS
Realization from Перегляд результатів to «FunctionalStandard» Перегляд результатів [Direction is 'Source -> Destination'.]
Realization from Перегляд результатів to «FunctionalStandard» Запит з вибору марки товару [Direction is 'Source -> Destination'.]
Realization from Перегляд результатів to «FunctionalStandard» Запит з вибору виду товару [Direction is 'Source -> Destination'.]
ASSOCIATIONS
Association (direction: Unspecified) Source: Public (Use Case) Перегляд результатів Target: Public (Use Case) Зміна налаштувань системи
Association (direction: Unspecified) Source: Public (Actor) Actor A Target: Public (Use Case) Перегляд результатів
Association (direction: Unspecified) Source: Public (Actor) Покупець Target: Public (Use Case) Перегляд результатів

Model Report	14 April, 2024
TESTING	
Standard Unit test. Тест кейс на перевірку коректного відображення результатів Description: 1) Виконати запит покупця 2) Натиснути кнопку сортувати 3) Отримати результат з фотографією продукції та всією інформацією з посиланням на сайт де можна придбати цю продукцію Input: Старт Results: результат з фотографією продукції та всією інформацією з посиланням на сайт де можна придбати цю продукцію 14.04.2024 - Pass Run by: Будує Checked by: Будує [Last run Pass at 14.04.2024 by Будує and checked by Будує.]	

Рисунок А.6 — Документація тест-кейсу перегляд результатів

ДОДАТОК Б

Фрагменти коду, що було створено для реалізації бекенду системи

```
import telebot

import mysql.connector

from telebot import types


db = mysql.connector.connect(

    host="localhost",

    user="root",

    passwd="1234567890ggvladGG",

    port="3306",

    database="tel_bot"

)

cursor = db.cursor()

#ОТОБРАЖАЕНИЕ АЙФОНОВ 15

def defX(message):

    if message.text == '👉 Від дорогих до дешевих👈':

        bot.reply_to(message, "•Кольори: чорний, блакитний, зелений, жовтий, рожевий\n•Ємність: 128 ГБ, 256 ГБ, 512 ГБ\n" + "•Розміри та вага:\n•Довжина: 147,6 мм\n•Ширина: 71,6 мм\n•Товщина: 7,80 мм\n•Вага: 171 г\n"

        "•Дисплей: OLED із повноекранним дизайном, діагональ 6,1 дюйма 2556×1179 пікселів, 460 пікселів на дюйм\n•Захист від бризок, води й пилу: клас захисту IP68 згідно зі стандартом IEC 60529 (занурення щонайбільше на 6 метрів протягом 30 хвилин)\n"

        + "•Чип: Чип A16 Bionic\n•Камера: Основна (48 Мп): 26 мм, діафрагма f/1.6, оптична стабілізація зображення зсувом матриці підтримка Focus Pixels по всій матриці, підтримка фото з неймовірно високою роздільною здатністю (24 Мп та 48 Мп)\n")

        markup_inline = types.InlineKeyboardMarkup()

        cursor.execute("SELECT ssulka_zena FROM site where product_name='Iphone 15' ORDER BY ssulka_zena DESC limit 1")

        myresult = cursor.fetchall()

        for x in myresult:

            bot.send_message(message.chat.id, x)


        cursor.execute("SELECT ssulka_ FROM site where product_name='Iphone 15' ORDER BY ssulka_zena DESC")

        myresult = cursor.fetchall()

        for x in myresult:

            s = str(x)

            s = s[2:len(s) - 3]

            markup_inline.add(types.InlineKeyboardButton(text='👉 Перейти на сайт👈', url=s))

            bot.send_photo(message.chat.id,

'https://estore.ua/media/catalog/product/cache/8/image/650x650/9df78eab33525d08d6e5fb8d27136e95/i/p/iphone-15-finish-select-202309-6-1inch-green_1.png', reply_markup=markup_inline)

    elif message.text == '👈 Від дешевих до дорогих👉':
```

```

markup_inline = types.InlineKeyboardMarkup()

bot.reply_to(message,

    "•Кольори: чорний, блакитний, зелений, жовтий, рожевий\n•Ємність: 128 ГБ, 256 ГБ, 512 ГБ\n" + "•Розміри та вага:\n•Довжина: 147,6 мм\n•Ширина: 71,6 мм\n•Товщина: 7,80 мм\n•Вага: 171 г\n"

    "•Дисплей: OLED із повноекранним дизайном, діагональ 6,1 дюйма 2556×1179 пікселів, 460 пікселів на дюйм\n•Захист від бризок, води й пилу: клас захисту IP68 згідно зі стандартом IEC 60529 (занурення щонайбільше на 6 метрів протягом 30 хвилин)\n"

    + "•Чип: Чип A16 Bionic\n•Камера: Основна (48 Мп): 26 мм, діафрагма f/1.6, оптична стабілізація зображення зсувом матриці підтримка Focus Pixels по всій матриці, підтримка фото з неймовірно високою роздільною здатністю (24 Мп та 48 Мп)\n")

cursor.execute("SELECT ssulka_zena FROM site where product_name='Iphone 15' ORDER BY ssulka_zena ASC limit 1")

myresult = cursor.fetchall()

for x in myresult:

    bot.send_message(message.chat.id, x)

cursor.execute("SELECT ssulka_ FROM site where product_name='Iphone 15' ORDER BY ssulka_zena ASC")

myresult = cursor.fetchall()

for x in myresult:

    s = str(x)

    s = s[2:len(s) - 3]

    markup_inline.add(types.InlineKeyboardButton(text='👉Перейти на сайт👈', url=s))

    bot.send_photo(message.chat.id,
'https://estore.ua/media/catalog/product/cache/8/image/650x650/9df78eab33525d08d6e5fb8d27136e95/i/p/iphone-15-finish-select-202309-6-1inch-green_1.png', reply_markup=markup_inline)

```

#ОТОБРАЖЕНИЕ АЙФОНОВ 14 Про Макс

```

def def11(message):

    if message.text == '👉Від дорогих до дешевих👈':

        bot.reply_to(message, "•Кольори: космічний чорний, сріблястий, золотистий, темно-фіолетовий\n•Ємність: 128 ГБ, 256 ГБ, 512 ГБ, 1 ТБ\n" + "•Розміри та вага:\n•Довжина: 160,7 мм\n•Ширина: 77,6 мм\n•Товщина: 7,85 мм\n•Вага: 240 г\n"

        "•Дисплей: OLED із повноекранним дизайном, діагональ 6,7 дюйма 2796×1290 пікселів, 460 пікселів на дюйм\n•Захист від бризок, води й пилу: Клас захисту IP68 згідно зі стандартом IEC 60529 (занурення щонайбільше на 6 метрів протягом 30 хвилин)\n"

        + "•Чип: Чип A16 Bionic\n•Камера: Основна (48 Мп): 24 мм, діафрагма f/1.78, оптична стабілізація зображення зсувом матриці 2-го покоління, семилінзовий об'єктив, підтримка Focus Pixels по всій матриці\n")

        markup_inline = types.InlineKeyboardMarkup()

        cursor.execute("SELECT ssulka_zena FROM site where product_name='Iphone 14 Pro Max' ORDER BY ssulka_zena DESC limit 1")

        myresult = cursor.fetchall()

        for x in myresult:

            bot.send_message(message.chat.id, x)

        cursor.execute("SELECT ssulka_ FROM site where product_name='Iphone 14 Pro Max' ORDER BY ssulka_zena DESC")

        myresult = cursor.fetchall()

        for x in myresult:

            s = str(x)

            s = s[2:len(s) - 3]

```

```

markup_inline.add(types.InlineKeyboardButton(text='★ Перейти на сайт ★', url=s))

bot.send_photo(message.chat.id,
'https://i.allo.ua/media/catalog/product/cache/3/image/2000x2000/af097278c5db4767b0fe9bb92fe21690/w/w/wwen_iphone14pro_q422_deep-purple_pdp-images_position-1a_7.jpg', reply_markup=markup_inline)

elif message.text == '👉 Від дешевих до дорогих 👈':

    bot.reply_to(message, "•Кольори: космічний чорний, сріблястий, золотистий, темно-фіолетовий\n•Ємність: 128 ГБ, 256 ГБ, 512 ГБ, 1 ТБ\n" + "•Розміри та вага:\n•Довжина: 160,7 мм\n•Ширина: 77,6 мм\n•Товщина: 7,85 мм\n•Вага: 240 г\n"

    "•Дисплей: OLED із повноекранним дизайном, діагональ 6,7 дюйма 2796×1290 пікселів, 460 пікселів на дюйм\n•Захист від брызг, води й пилу: Клас захисту IP68 згідно зі стандартом IEC 60529 (занурення щонайбільше на 6 метрів протягом 30 хвилин)\n"

    + "•Чип: Чип A16 Bionic\n•Камера: Основна (48 Мп): 24 мм, діафрагма f/1.78, оптична стабілізація зображення зсувом матриці 2-го покоління, семілінзовий об'єктив, підтримка Focus Pixels по всій матриці\n")

    markup_inline = types.InlineKeyboardMarkup()

    cursor.execute("SELECT ssulka_zena FROM site where product_name='Iphone 14 Pro Max' ORDER BY ssulka_zena ASC limit 1")

    myresult = cursor.fetchall()

    for x in myresult:

        bot.send_message(message.chat.id, x)


    cursor.execute("SELECT ssulka_ FROM site where product_name='Iphone 14 Pro Max' ORDER BY ssulka_zena ASC")

    myresult = cursor.fetchall()

    for x in myresult:

        s = str(x)

        s = s[2:len(s) - 3]

        markup_inline.add(types.InlineKeyboardButton(text='★ Перейти на сайт ★', url=s))

    bot.send_photo(message.chat.id,
'https://i.allo.ua/media/catalog/product/cache/3/image/2000x2000/af097278c5db4767b0fe9bb92fe21690/w/w/wwen_iphone14pro_q422_deep-purple_pdp-images_position-1a_7.jpg', reply_markup=markup_inline)

```

#ОТОБРАЖАЕНИЕ АЙФОНОВ 15 Про

```
def defXR(message):
```

```

    if message.text == '👉 Від дорогих до дешевих 👈':

        bot.reply_to(message, "•Кольори: чорний титан, білий титан, синій титан, натуральний титан\n•Ємність: 128 ГБ, 256 ГБ, 512 ГБ, 1 ТБ\n" + "•Розміри та вага:\n•Довжина: 146,6 мм\n•Ширина: 70,6 мм\n•Товщина: 8,25 мм\n•Вага: 187 г\n"

        "•Дисплей: OLED із повноекранним дизайном, діагональ 6,1 дюйма 2556×1179 пікселів, 460 пікселів на дюйм\n•Захист від брызг, води й пилу: Клас захисту IP68 згідно зі стандартом IEC 60529 (занурення щонайбільше на 6 метрів протягом 30 хвилин)\n"

        + "•Чип: Чип A17 Pro\n•Камера: Основна (48 Мп): 24 мм, діафрагма f/1.78, оптична стабілізація зображення зсувом матриці 2-го покоління, підтримка Focus Pixels по всій матриці, підтримка фото з неймовірно високою роздільною здатністю (24 Мп та 48 Мп)\n")

        markup_inline = types.InlineKeyboardMarkup()

        cursor.execute("SELECT ssulka_zena FROM site where product_name='Iphone 15 Pro' ORDER BY ssulka_zena DESC limit 1")

        myresult = cursor.fetchall()

        for x in myresult:

            bot.send_message(message.chat.id, x)


        cursor.execute("SELECT ssulka_ FROM site where product_name='Iphone 15 Pro' ORDER BY ssulka_zena DESC")

```

```

myresult = cursor.fetchall()

for x in myresult:

    s = str(x)

    s = s[2:len(s) - 3]

    markup_inline.add(types.InlineKeyboardButton(text='👉Перейти на сайт👈', url=s))

bot.send_photo(message.chat.id, 'https://cdn.tehnoezh.ua/0/0/0/1/4/2/6/0/2/000142602_1054_1054.jpeg', reply_markup=markup_inline)

elif message.text == '👉Від дешевих до дорогих👈':

    bot.reply_to(message, "•👉Кольори: чорний титан, білий титан, синій титан, натуральний титан👈•Ємність: 128 ГБ, 256 ГБ, 512 ГБ, 1
ТБ👈" + "•Розміри та вага:👈•Довжина: 146,6 мм👈•Ширина: 70,6 мм👈•Товщина: 8,25 мм👈•Вага: 187 г👈"

    "•👉Дисплей: OLED із повноекранним дизайном, діагональ 6,1 дюйма 2556×1179 пікселів, 460 пікселів на дюйм👈•Захист від
бризок, води й пилу: Клас захисту IP68 згідно зі стандартом IEC 60529 (занурення щонайбільше на 6 метрів протягом 30 хвилин)👈"

    + "•Чип: Чип A17 Pro👈•Камера: Основна (48 Мп): 24 мм, діафрагма f/1.78, оптична стабілізація зображення зсувом матриці
2-го покоління, підтримка Focus Pixels по всій матриці, підтримка фото з неймовірно високою роздільною здатністю (24 Мп та 48
Мп)👈")

    markup_inline = types.InlineKeyboardMarkup()

    cursor.execute("SELECT ssulka_zena FROM site where product_name='Iphone 15 Pro' ORDER BY ssulka_zena ASC limit 1")

    myresult = cursor.fetchall()

    for x in myresult:

        bot.send_message(message.chat.id, x)


cursor.execute("SELECT ssulka_ FROM site where product_name='Iphone 15 Pro' ORDER BY ssulka_zena ASC")

myresult = cursor.fetchall()

for x in myresult:

    s = str(x)

    s = s[2:len(s) - 3]

    markup_inline.add(types.InlineKeyboardButton(text='👉Перейти на сайт👈', url=s))

bot.send_photo(message.chat.id, 'https://cdn.tehnoezh.ua/0/0/0/1/4/2/6/0/2/000142602_1054_1054.jpeg', reply_markup=markup_inline)

```

#ОТОБРАЖАЕНИЕ АЙФОНОВ 15 Про Макс

```

def defXS(message):

    if message.text == '👉Від дорогих до дешевих👈':

        bot.reply_to(message, "•Кольори: чорний титан, білий титан, синій титан, натуральний титан👈•Ємність: 128 ГБ, 256 ГБ, 512 ГБ, 1
ТБ👈" + "•Розміри та вага:👈•Довжина: 159,9 мм👈•Ширина: 76,7 мм👈•Товщина: 8,25 мм👈•Вага: 221 г👈"

        "•Дисплей: OLED із повноекранним дизайном, діагональ 6,7 дюйма 2796×1290 пікселів, 460 пікселів на дюйм👈•Захист від бризок,
води й пилу: Клас захисту IP68 згідно зі стандартом IEC 60529 (занурення щонайбільше на 6 метрів протягом 30 хвилин)👈"

        + "•Чип: Чип A17 Pro👈•Камера: Основна (48 Мп): 24 мм, діафрагма f/1.78, оптична стабілізація зображення зсувом матриці
2-го покоління, підтримка Focus Pixels по всій матриці, підтримка фото з неймовірно високою роздільною здатністю (24 Мп та 48
Мп)👈")

        markup_inline = types.InlineKeyboardMarkup()

        cursor.execute("SELECT ssulka_zena FROM site where product_name='Iphone 15 Pro Max' ORDER BY ssulka_zena DESC limit 1")

        myresult = cursor.fetchall()

        for x in myresult:

            bot.send_message(message.chat.id, x)

```

```

cursor.execute("SELECT ssulka_ FROM site where product_name='Iphone 15 Pro Max' ORDER BY ssulka_zena DESC")

myresult = cursor.fetchall()

for x in myresult:

    s = str(x)

    s = s[2:len(s) - 3]

    markup_inline.add(types.InlineKeyboardButton(text='Перейти на сайт', url=s))

    bot.send_photo(message.chat.id,
'https://yabloki.ua/media/cache/sylius_shop_product_original/a7/f6/4017ee7a3bc96384a0941b023288.png.webp', reply_markup=markup_inline)

    elif message.text == '👇 Від дешевих до дорогих👆':

        bot.reply_to(message, "•Кольори: чорний титан, білий титан, синій титан, натуральний титан\n•Ємність: 128 ГБ, 256 ГБ, 512 ГБ, 1
ТБ\n" + "•Розміри та вага:\n•Довжина: 159,9 мм\n•Ширина: 76,7 мм\n•Товщина: 8,25 мм\n•Вага: 221 г\n"

        "•Дисплей: OLED із повноекранним дизайном, діагональ 6,7 дюйма 2796×1290 пікселів, 460 пікселів на дюйм\n•Захист від брызг,
води й пилу: Клас захисту IP68 згідно зі стандартом IEC 60529 (занурення щонайбільше на 6 метрів протягом 30 хвилин)\n"

        + "•Чип: Чип A17 Pro\n•Камера: Основна (48 Мп): 24 мм, діафрагма f/1.78, оптична стабілізація зображення зсувом матриці
2-го покоління, підтримка Focus Pixels по всій матриці, підтримка фото з неймовірно високою роздільною здатністю (24 Мп та 48
Мп)\n")

        markup_inline = types.InlineKeyboardMarkup()

        cursor.execute("SELECT ssulka_zena FROM site where product_name='Iphone 15 Pro Max' ORDER BY ssulka_zena ASC limit 1")

        myresult = cursor.fetchall()

        for x in myresult:

            bot.send_message(message.chat.id, x)

cursor.execute("SELECT ssulka_ FROM site where product_name='Iphone 15 Pro Max' ORDER BY ssulka_zena ASC")

myresult = cursor.fetchall()

for x in myresult:

    s = str(x)

    s = s[2:len(s) - 3]

    markup_inline.add(types.InlineKeyboardButton(text='Перейти на сайт', url=s))

    bot.send_photo(message.chat.id,
'https://yabloki.ua/media/cache/sylius_shop_product_original/a7/f6/4017ee7a3bc96384a0941b023288.png.webp', reply_markup=markup_inline)

```

#ОТОБРАЖЕНИЕ АЙФОНОВ 15 Плюс

```

def defXS_MAX(message):

    if message.text == '👇 Від дорогих до дешевих👆':

        bot.reply_to(message,

            "•Кольори: чорний, блакитний, зелений, жовтий, рожевий\n•Ємність: 128 ГБ, 256 ГБ, 512 ГБ\n" + "•Розміри та
вага:\n•Довжина: 160,9 мм\n•Ширина: 77,8 мм\n•Товщина: 7,80 мм\n•Вага: 201 г\n"

            "•Дисплей: OLED із повноекранним дизайном, діагональ 6,7 дюйма
2796×1290 пікселів, 460 пікселів на дюйм\n•Захист від брызг, води й пилу: клас захисту IP68 згідно зі стандартом IEC 60529
(занурення щонайбільше на 6 метрів протягом 30 хвилин)\n"

            + "•Чип: Чип A16 Bionic\n•Камера: Основна (48 Мп): 26 мм, діафрагма f/1.6, оптична стабілізація зображення зсувом
матриці підтримка Focus Pixels по всій матриці, підтримка фото з неймовірно високою роздільною здатністю (24 Мп та 48 Мп)\n")

        markup_inline = types.InlineKeyboardMarkup()

        cursor.execute("SELECT ssulka_zena FROM site where product_name='Iphone 15 Plus' ORDER BY ssulka_zena DESC limit 1")

```

```

myresult = cursor.fetchall()

for x in myresult:

    bot.send_message(message.chat.id, x)


cursor.execute("SELECT ssulka_ FROM site where product_name='iPhone 15 Plus' ORDER BY ssulka_zena DESC")

myresult = cursor.fetchall()

for x in myresult:

    s = str(x)

    s = s[2:len(s) - 3]

    markup_inline.add(types.InlineKeyboardButton(text='🔗Перейти на сайт🔗', url=s))

bot.send_photo(message.chat.id, 'https://files.foxtrot.com.ua/PhotoNew/1_638301567897107760.webp', reply_markup=markup_inline)

elif message.text == '👉Від дешевих до дорогих👈':

    bot.reply_to(message,

        "•Кольори: чорний, блакитний, зелений, жовтий, рожевий\n•Ємність: 128 ГБ, 256 ГБ, 512 ГБ\n" + "•Розміри та вага:\n•□Довжина: 160,9 мм\n•Ширина: 77,8 мм\n•Товщина: 7,80 мм\n•Вага: 201 г\n"

        "•Дисплей: OLED із повноекранним дизайном, діагональ 6,7 дюйма 2796×1290 пікселів, 460 пікселів на дюйм\n•Захист від бризок, води й пилу: клас захисту IP68 згідно зі стандартом IEC 60529 (занурення щонайбільше на 6 метрів протягом 30 хвилин)\n"

        + "•□Чип: Чип A16 Bionic\n•Камера: Основна (48 Мп): 26 мм, діафрагма f/1.6, оптична стабілізація зображення зсувом матриці підтримка Focus Pixels по всій матриці, підтримка фото з неймовірно високою роздільною здатністю (24 Мп та 48 Мп)\n")

    markup_inline = types.InlineKeyboardMarkup()

    cursor.execute("SELECT ssulka_zena FROM site where product_name='Iphone 15 Plus' ORDER BY ssulka_zena ASC limit 1")

    myresult = cursor.fetchall()

    for x in myresult:

        bot.send_message(message.chat.id, x)


cursor.execute("SELECT ssulka_ FROM site where product_name='iPhone 15 Plus' ORDER BY ssulka_zena ASC")

myresult = cursor.fetchall()

for x in myresult:

    s = str(x)

    s = s[2:len(s) - 3]

    markup_inline.add(types.InlineKeyboardButton(text='🔗Перейти на сайт🔗', url=s))

bot.send_photo(message.chat.id, 'https://files.foxtrot.com.ua/PhotoNew/1_638301567897107760.webp', reply_markup=markup_inline)


#2113973109:AAHpjnDYmcP1DgMi6sc8IUkHmahmchrjU5I

TOKEN = "2113973109:AAHpjnDYmcP1DgMi6sc8IUkHmahmchrjU5I"

bot = telebot.TeleBot(TOKEN)

```

```

#@bot.message_handler(commands=['Iphone_X'])

#def get_user_info1(message):

# cursor.execute("SELECT ssulka_ FROM site where product_name='Iphone 15'")

#myresult = cursor.fetchall()

# for x in myresult:

# bot.send_message(message.from_user.id, text=x)

# print(x)

#cursor.close()

#НАЧАЛО БОТА

@bot.message_handler(commands=['start'])

def get_user_info(message):

    bot.reply_to(message, "👋 KanBuy - простой и быстрый способ\n покупки и сравнения техники:👉\n"+"💎 Apple 💎\n"

                                "💎 Samsung 💎")

    markup_inline = types.InlineKeyboardMarkup()

    item_yes = types.InlineKeyboardButton(text='Apple', callback_data='yes')

    item_no = types.InlineKeyboardButton(text='Samsung', callback_data='no')

    markup_inline.add(item_yes, item_no)

    bot.send_message(message.chat.id, '👉Выберите категорию:👈',

                        reply_markup=markup_inline

                    )

@bot.callback_query_handler(func=lambda call: True)

def answer(call):

    if call.data == 'yes':

        markup_reply = types.ReplyKeyboardMarkup(resize_keyboard=True)

        item_id = types.KeyboardButton('📱 Iphone')

        item_username = types.KeyboardButton('📱 Ipad')

        back = types.KeyboardButton('В главное меню')

        markup_reply.add(item_id, item_username, back)

        bot.send_message(call.message.chat.id, 'Выберите категорию: ',

                        reply_markup= markup_reply

                    )

        bot.answer_callback_query(callback_query_id=call.id)

    elif call.data == 'no':

        markup_reply = types.ReplyKeyboardMarkup(resize_keyboard=False)

```

```

bot.send_message(call.message.chat.id, '📱В розробці!',
                  reply_markup=markup_reply
                  )
bot.answer_callback_query(callback_query_id=call.id)

```

```

@bot.message_handler(content_types=['text'])

```

```

def Iphone_1(message):

```

```

    if message.chat.type == 'private':

```

```

        if message.text == '📱 Iphone':

```

```

            markup_reply = types.ReplyKeyboardMarkup(resize_keyboard=True)

```

```

            X = types.KeyboardButton('📱 Iphone 15 📱')

```

```

            X_s = types.KeyboardButton('📶 Iphone 15 Pro Max 📶')

```

```

            X_s_max = types.KeyboardButton('🔋 iPhone 15 Plus 🔋')

```

```

            X_r = types.KeyboardButton('📱 Iphone 15 Pro 📱')

```

```

            X_11 = types.KeyboardButton('★ Iphone 14 Pro Max ★')

```

```

            back = types.KeyboardButton('↩️Назад')

```

```

            markup_reply.add(X_r, X, X_s, X_s_max, X_11, back)

```

```

            bot.send_message(message.chat.id, '📱 IPHONE: 📱', reply_markup=markup_reply)

```

```

        elif message.text == '📱 Ipad':

```

```

            bot.send_message(message.chat.id, '📱В розробці!')

```

```

        elif message.text == '↩️Назад':

```

```

            markup_reply = types.ReplyKeyboardMarkup(resize_keyboard=True)

```

```

            iphone = types.KeyboardButton('📱 Iphone')

```

```

            ipad = types.KeyboardButton('📱 Ipad')

```

```

            back = types.KeyboardButton('В главное меню')

```

```

            markup_reply.add(iphone, ipad, back)

```

```

            bot.send_message(message.chat.id, '↩️Назад', reply_markup=markup_reply)

```

```

        elif message.text == '↩️Повернуться до моделей':

```

```

            markup_reply = types.ReplyKeyboardMarkup(resize_keyboard=True)

```

```

            X = types.KeyboardButton('📱 Iphone 15 📱')

```

```

            X_s = types.KeyboardButton('📶 Iphone 15 Pro Max 📶')

```

```

            X_s_max = types.KeyboardButton('🔋 iPhone 15 Plus 🔋')

```

```

            X_r = types.KeyboardButton('📱 Iphone 15 Pro 📱')

```

```

            X_11 = types.KeyboardButton('★ Iphone 14 Pro Max ★')

```

```

            back = types.KeyboardButton('↩️Назад')

```

```

markup_reply.add(X_r, X, X_s, X_s_max, X_11, back)

bot.send_message(message.chat.id, '📱 IPHONE:📱', reply_markup=markup_reply)

elif message.text == 'В главное меню':

    markup_reply = types.ReplyKeyboardMarkup(resize_keyboard=True)

    apple = types.KeyboardButton('🍏 Apple 🍏')

    samsung = types.KeyboardButton('📱 Samsung 📱')

    markup_reply.add(apple, samsung)

    bot.send_message(message.chat.id, 'Вы в главном меню', reply_markup=markup_reply)


elif message.text == '📱 Samsung 📱':

    bot.send_message(message.chat.id, '🔍 В розробці 🔍')


elif message.text == '🍏 Apple 🍏':

    markup_reply = types.ReplyKeyboardMarkup(resize_keyboard=True)

    iphone = types.KeyboardButton('📱 Iphone')

    ipad = types.KeyboardButton('📱 Ipad')

    back = types.KeyboardButton('В главное меню')

    markup_reply.add(iphone, ipad, back)

    bot.send_message(message.chat.id, 'Виберіть категорію:', reply_markup=markup_reply)

elif message.text == '📱 Iphone 15📱':

    markup_reply = types.ReplyKeyboardMarkup(resize_keyboard=True)

    zena_ot = types.KeyboardButton('📉 Від дорогих до дешевих📉')

    zena_do = types.KeyboardButton('📈 Від дешевих до дорогих📈')

    back = types.KeyboardButton('↩️ Повернутись до моделей')

    markup_reply.add(zena_ot, zena_do, back)

    bot.send_message(message.chat.id, 'Виберіть тип сортування ціни: ', reply_markup=markup_reply)

    bot.register_next_step_handler(message, defX)


# if message.text == '📉 Цена📉':

#     cursor.execute("SELECT ssulka_ FROM site where product_name='Iphone 15' ORDER BY ssulka_zena DESC")

#     myresult = cursor.fetchall()

#     # for x in myresult:

#         # bot.send_message(message.from_user.id, text=x)

#         # print(x)

elif message.text == '📱 Iphone 15 Pro Max📱':

    markup_reply = types.ReplyKeyboardMarkup(resize_keyboard=True)

    zena_ot = types.KeyboardButton('📉 Від дорогих до дешевих📉')

    zena_do = types.KeyboardButton('📈 Від дешевих до дорогих📈')

```

```

back = types.KeyboardButton('↩️Повернутись до моделей')

markup_reply.add(zena_ot, zena_do, back)

bot.send_message(message.chat.id, 'Виберіть тип сортування ціни: ', reply_markup=markup_reply)

bot.register_next_step_handler(message, defXS)

elif message.text == '📱 iPhone 15 Pro 📱':

    markup_reply = types.ReplyKeyboardMarkup(resize_keyboard=True)

    zena_ot = types.KeyboardButton('📉 Від дорогих до дешевих ⬇️')

    zena_do = types.KeyboardButton('📈 Від дешевих до дорогих ⬆️')

    back = types.KeyboardButton('↩️Повернутись до моделей')

    markup_reply.add(zena_ot, zena_do, back)

    bot.send_message(message.chat.id, 'Виберіть тип сортування ціни: ', reply_markup=markup_reply)

    bot.register_next_step_handler(message, defXR)

elif message.text == '📱 iPhone 15 Plus 📱':

    markup_reply = types.ReplyKeyboardMarkup(resize_keyboard=True)

    zena_ot = types.KeyboardButton('📉 Від дорогих до дешевих ⬇️')

    zena_do = types.KeyboardButton('📈 Від дешевих до дорогих ⬆️')

    back = types.KeyboardButton('↩️Повернутись до моделей')

    markup_reply.add(zena_ot, zena_do, back)

    bot.send_message(message.chat.id, 'Виберіть тип сортування ціни: ', reply_markup=markup_reply)

    bot.register_next_step_handler(message, defXS_MAX)

elif message.text == '★ iPhone 14 Pro Max ★':

    markup_reply = types.ReplyKeyboardMarkup(resize_keyboard=True)

    zena_ot = types.KeyboardButton('📉 Від дорогих до дешевих ⬇️')

    zena_do = types.KeyboardButton('📈 Від дешевих до дорогих ⬆️')

    back = types.KeyboardButton('↩️Повернутись до моделей')

    markup_reply.add(zena_ot, zena_do, back)

    bot.send_message(message.chat.id, 'Виберіть тип сортування ціни: ', reply_markup=markup_reply)

    bot.register_next_step_handler(message, def11)

# elif message.text == '📉 Ціна ⬇️':

# if message.text == 'iPhone 15':

# cursor.execute("SELECT ssulka_ FROM site where product_name='iPhone 15' ORDER BY ssulka_zena DESC")

# myresult = cursor.fetchall()

# for x in myresult:

#     bot.send_message(message.from_user.id, text=x)

#     print(x)

# elif message.text == '📈 Ціна ⬆️':

# if message.text == 'iPhone 15':

# cursor.execute("SELECT ssulka_ FROM site where product_name='iPhone 15' ORDER BY ssulka_zena ASC")

# myresult = cursor.fetchall()

```

```
# for x in myresult:

# bot.send_message(message.from_user.id, text=x)

# print(x)

bot.polling(none_stop= True)

# elif message.text == 'Ввести минимальную цену:':

# markup_inline = types.InlineKeyboardMarkup()

# cursor.execute("SELECT ssulka_ FROM site where product_name='Iphone 15'")

# myresult = cursor.fetchall()

# for x in myresult:

# item_yes = types.InlineKeyboardButton(string=x, callback_data='yes')

# item_no = types.InlineKeyboardButton(string=x, callback_data='no')

# markup_inline.add(item_yes, item_no)

# bot.send_message(message.chat.id, '👉Виберіть категорію:👈', reply_markup=markup_inline)
```

ДОДАТОК В

Сторінки прототипу системи зроблені у Figma

Розглянемо прототипи сторінок інформаційної системи «Телеграм-бот для порівняння цін на цифрову техніку» що представлені на рисунок В.1 – В.6.

Головна сторінка як і будь яка інша – містить одразу інформацію з можливим розвитком ситуації. Після входу до телеграм-боту «KanBuy», показано його головне меню на рисунку В.1, у телеграм-боті є кнопки «Apple» та «Samsung», до того ж ми можемо побачити коротку інформацію про бота.

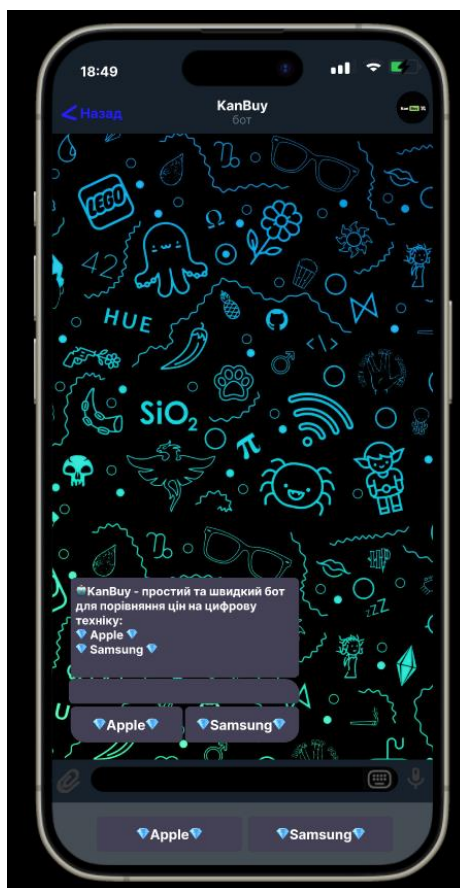


Рисунок В.1 – Головне меню телеграм-бота «KanBuy»

Відображено дві компанії, які містяться на даний момент в нашому боті: «Apple» та «Samsung».

На рисунку В.2 представлено вибір кнопки «Apple», що переносить нас до вибору категорій, де представлені у нас клавіші «Iphone» та «Ipad», це першочергові категорії з подальшою розробкою, вибір буде ставати дедалі більшим, також кнопка «В головне меню», яка виконує однойменну дію з назвою кнопки.

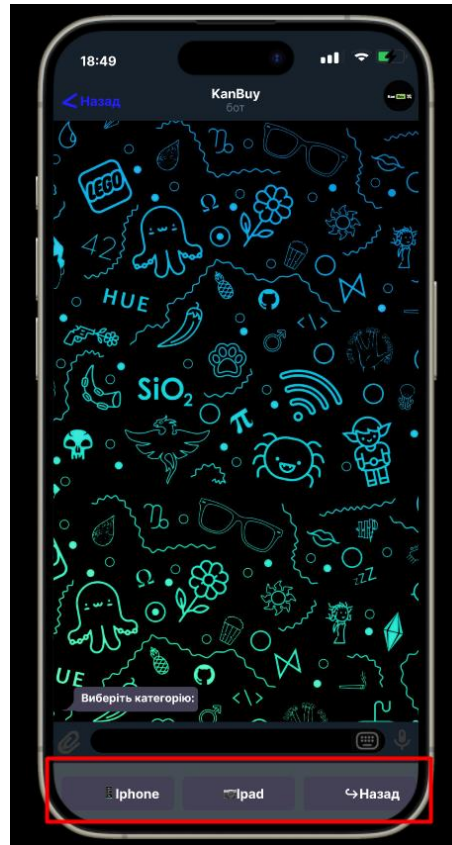


Рисунок В.2 – Меню категорій товарів

На рисунку В.3 представлено вибір кнопки «Iphone», що переносить нас до вибору моделі, де ми можемо обрати цікаву сьогоднішню для нас модель, наприклад, такі як: «Iphone 15», «Iphone 15 Plus», «Iphone 15 Pro», «Iphone 15 Pro Max» кожна з яких виконана в індивідуальному стилі, для спрощення ідентифікування її, а також, там є клавіша «Назад», яка повертає користувача до вибору категорії. В боті будуть відображатись тільки останні моделі цифрової техніки.

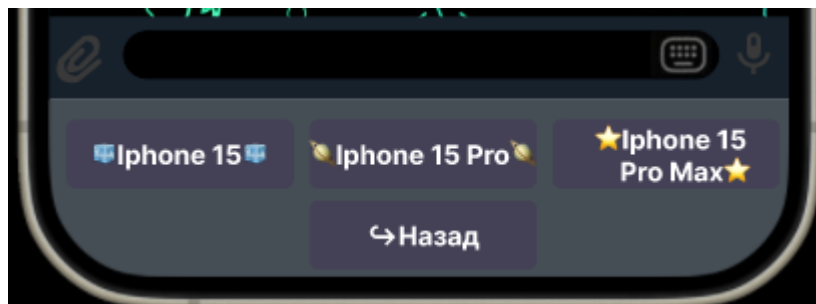


Рисунок В.3 – Вибір моделей в телеграм-боті «KanBuy»

На рисунку В.4 представлено вибір одної з зазначених кнопок на (рисунку В.3) кнопка «iPhone 15», при натисканні якої ми можемо побачити кнопки: «Від дорогих до дешевих» та «Від дешевих до дорогих» — скориставшись ними, ми можемо відсортувати ціну за спаданням або зростанням, до того ж там представлена ще одна кнопка «Назад» — яка відправляє нас до списку кнопок де є у нас моделі телефону.

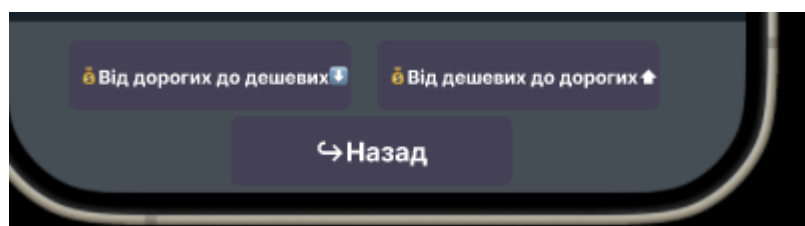


Рисунок В.4 – Вибір моделі «iPhone 15» телеграм-бота «KanBuy»

На рисунку В.5 представлено вибір одної з зазначених кнопок на (рисунку В.4) кнопка «Від дорогих до дешевих», при натисканні якої у нас відображаються відомості про товар за спаданням, де ми можемо перейти на сайти з актуально вибраною продукцією, натиснувши відповідне посилання.

До того ж, ми можемо подивитись мінімальну та максимальну ціну у вибраній категорії товару, що дозволить нам більш простіше орієнтуватися в цінах вибраного товару.

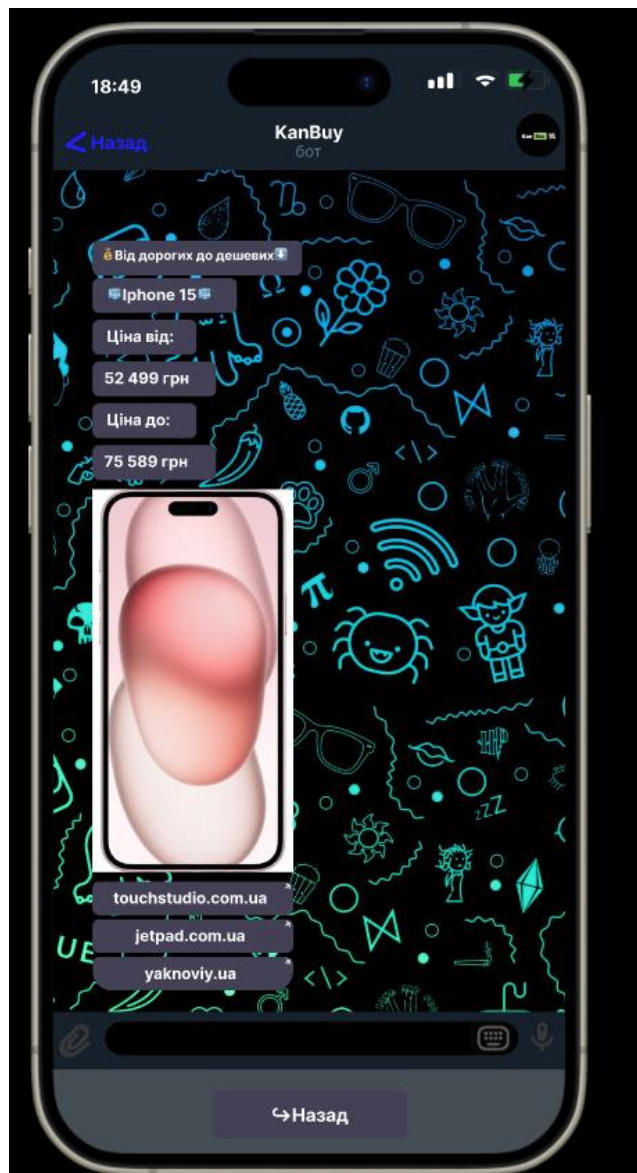


Рисунок В.5 – Сортуння магазинів за параметром «Від дорогих до дешевих» в телеграм-боті «KanBuy», ціни відображаються актуальним діапазоном, серед якого й може користувач зробити вибір

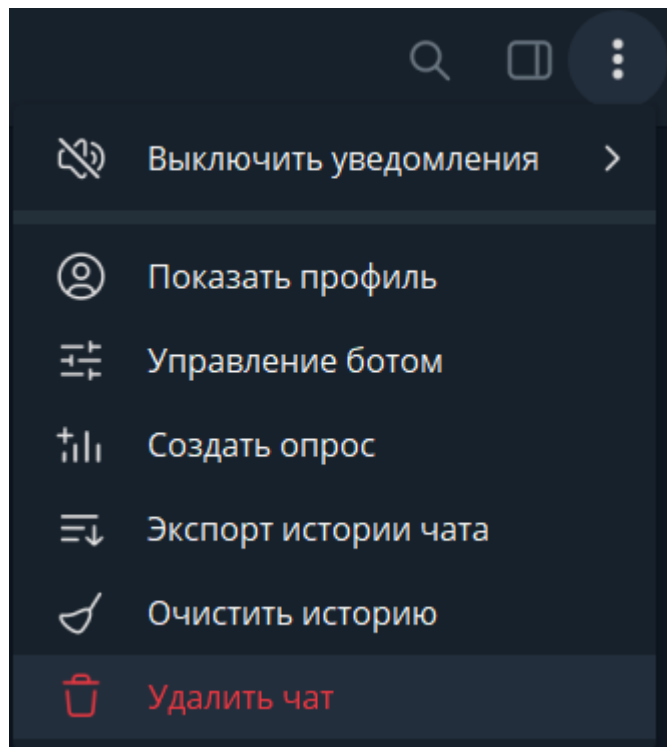


Рисунок В.6 – Відмова покупця від роботи з ботом на будь-якому етапі взаємодії з ботом

ДОДАТОК Г

Результат перетворення ER-моделі в модель даних СУБД MySQL

Згенерований код:

```
site create database if not exists tel_bot character set utf8;
```

```
use tel_bot;
```

```
create table site(
```

```
ssulka_id tinyint(4) auto_increment primary key,
```

```
product_name varchar(25) not null,
```

```
ssulka_ varchar(2500) not null,
```

```
ssulka_zena float(6) not null
```

```
) character set utf8;
```

```
create table obshiy(
```

```
obshiy_id tinyint(4) auto_increment primary key,
```

```
ssulka_id tinyint(4) NOT NULL,
```

```
zena_id tinyint(4) NOT NULL,
```

```
site_ssulka_id tinyint(4) NOT NULL
```

```
) character set utf8;
```

```
create table zena_tovar(
```

```
zena_id tinyint(4) auto_increment primary key,
```

```
ssulka_id tinyint(4) NOT NULL,
```

```
zena_tovara varchar(25) not null,
```

```
constraint FK_zena foreign key (ssulka_id) references  
site(ssulka_id)
```

```
on delete cascade on update cascade,
```

```
constraint FK_site_ssulka foreign key (ssulka_id) references  
site_ssulka(ssulka_id)
```

```
on delete cascade on update cascade
```

```
) character set utf8;
```

```
create table site_ssulka(
```

```
site_ssulka_id tinyint(4) auto_increment primary key,
```

```
ssulka_id tinyint(4) NOT NULL,
```

```
zena_id tinyint(4) NOT NULL,
```

```
ssulka_product varchar(2500) not null,
```

```
constraint FK_ssulka foreign key (ssulka_id) references  
site(ssulka_id)
```

```
on delete cascade on update cascade
```

```
) character set utf8;
```

Заповнення БД:

```
USE tel_bot;
```

```
SET CHARACTER SET utf8;
```

```
DELETE FROM site;
```

```
INSERT INTO site VALUES (1,'Iphone  
X','https://jetpad.com.ua/p1104918007-apple-iphone-  
64gb.html?utm_medium=cpc&utm_source=hotline&utm_campaig  
n=%D0%A1%D0%BC%D0%B0%D1%80%D1%82%D1%84%D  
0%BE%D0%BD%D1%8B+%D0%B8+%D0%BC%D0%BE%D0  
%B1%D0%B8%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5  
+%D1%82%D0%B5%D0%BB%D0%B5%D1%84%D0%BE%D0  
%BD%D1%8B&utm_term=Apple+iPhone+X+64GB+Space+Gra  
y+%28MQAC2%29&utm_content=1104918007',10954);
```

```
INSERT INTO site VALUES (2,'Iphone  
X','https://yaknoviy.ua/smartfon-apple-iphone-x-64gb-space-gray-  
mqac2-vitrinnyj-variant-  
item?utm_medium=cpc&utm_source=hotline&utm_campaign=%  
D0%A1%D0%BC%D0%B0%D1%80%D1%82%D1%84%D0%B  
E%D0%BD%D1%8B+%D0%B8+%D0%BC%D0%BE%D0%B1  
%D0%B8%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5+%D  
1%82%D0%B5%D0%BB%D0%B5%D1%84%D0%BE%D0%BD  
%D1%8B&utm_term=Apple+iPhone+X+64GB+Space+Gray+%2  
8MQAC2%29&utm_content=220223',10299);
```

```
INSERT INTO site VALUES (3,'Iphone  
X','https://touchstudio.com.ua/product/apple-iphone-x-64gb-space-  
gray-mqac2',17063);
```

```
DELETE FROM site;
```

```
INSERT INTO `tel_bot`.`site_ssulka` (`site_ssulka_id`,  
`ssulka_id`, `zena_id`, `ssulka_product`) VALUES ('1', '1', '1',  
'https://www.istore.ua/ua/item/apple-iphone-15-128-gb-  
pink/?gad_source=1&gclid=CjwKCAiAjrArBhAWEiwA2qWdCH  
RC840ytwf6rRRsP7mnHx4q3QOJm5E2j8l_MAZf_bCpCXiyKI20  
GxoCsI4QAvD_BwE');
```

```
INSERT INTO `tel_bot`.`site_ssulka` (`site_ssulka_id`,  
`ssulka_id`, `zena_id`, `ssulka_product`) VALUES ('2', '2', '2',  
'https://estore.ua/iphone-15-128gb-  
green/?gad_source=1&gclid=CjwKCAiAjrArBhAWEiwA2qWdCC  
kcnv9_enhLoe8yk7N1HPJbRGqB81gTJ0ybsxPQZG7Wdkp2Fb5T  
4RoCtLUQAvD_BwE');
```

```
INSERT INTO `tel_bot`.`site_ssulka` (`site_ssulka_id`,  
`ssulka_id`, `zena_id`, `ssulka_product`) VALUES ('3', '3', '3',  
'https://www.istore.ua/catalog/item/apple-iphone-15-128-gb-
```

```
black/?gad_source=1&gclid=CjwKCAiAjrArBhAWEiwA2qWdCO  
0fgvwMyKrb58H0o2cxQ7QkdOORzovjL-  
cvWDTAZqAqzBzlfytllBoCfJgQAvD_BwE');
```

```
DELETE FROM site;
```

```
INSERT INTO `tel_bot`.`zena_tovar` (`zena_id`, `ssulka_id`,  
`zena_tovara`) VALUES ('1', '1', '54000');
```

```
INSERT INTO `tel_bot`.`zena_tovar` (`zena_id`, `ssulka_id`,  
`zena_tovara`) VALUES ('2', '2', '27000');
```

```
INSERT INTO `tel_bot`.`zena_tovar` (`zena_id`, `ssulka_id`,  
`zena_tovara`) VALUES ('3', '3', '30000');
```

ДОДАТОК Д

Фрагменти ескізів форм подання вихідної інформації:

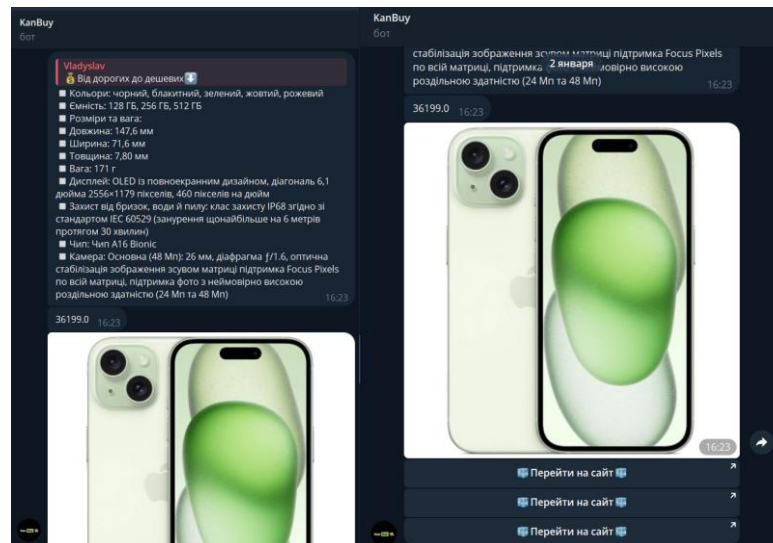


Рисунок Д-1 Ескіз форми вихідних даних



Рисунок Д-2 – Показ відсортованих сайтів за формулою