

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ПОЛТАВСЬКИЙ ПОЛІТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Циклова комісія дисциплін програмної інженерії

Затверджую:

Голова циклової комісії

_____ / Олександр БАБИЧ

“ ____ ” _____ 2024 р.

КУРСОВИЙ ПРОЄКТ

з дисципліни: “Проєктування автоматизованих інформаційних систем”
на тему _____ Розробка автоматизованої інформаційної системи
_____ для магазину одягу (телеграм бот)

КП 000.000000.045

Виконав: здобувач освіти 4 курсу,
групи ____ 44 ____
спеціальність 121 «Інженерія
програмного забезпечення»
спеціалізація «Розробка програмного
забезпечення»

_____ Артем ПРИХОДЬКО

(прізвище та ініціали)

Керівник _____ Світлана ГРИЦЕНКО
(підпис) (прізвище та ініціали)

Полтава – 2024

ЗМІСТ

ВСТУП	5
1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАВДАННЯ	7
1.1. Постановка задачі	7
1.2. Розробка прототипу інтерфейсу	8
1.3. Опис та побудова діаграми прецедентів роботи з системою	12
1.4. Інфологічне проєктування – схема бази даних	18
2. ОПИС ПРОГРАМНИХ ЗАСОБІВ	21
3. РЕАЛІЗАЦІЯ РОЗВ’ЯЗАННЯ ЗАДАЧІ ПРОЕКТУВАННЯ	24
3.1. Алгоритм програми, архітектура програми	24
3.2. Структура програми з функцій складових частин і зв’язків між ними	24
3.3. Виклик та завантаження	25
3.4. Інструкція користувачу	25
ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	31
ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ	32
ДОДАТОК Б. ЛІСТИНГ ПРОГРАМИ	36
ДОДАТОК В. РЕПОЗИТОРІЙ ДО ПРОГРАМИ	38

ВСТУП

Актуальність роботи: у сучасному світі автоматизація є ключовим фактором підвищення ефективності та продуктивності бізнес-процесів. Інтернет-торгівля є одним із найдинамічніших сегментів ринку, що вимагає впровадження новітніх технологій для забезпечення конкурентоспроможності та зручності для користувачів. В епоху цифровізації навіть найпростіші операції, такі як покупка товару або реєстрація користувача, можуть бути значно спрощені за допомогою автоматизованих інформаційних систем. У зв'язку з цим зростає попит на розробку таких систем, які дозволяють компаніям або приватним особам надавати своїм клієнтам зручний сервіс із будь-якого місця, що має доступ до Інтернету. Фактично, Telegram, як одна з найпопулярніших платформ для миттєвого обміну повідомленнями, є потужним інструментом для реалізації таких інформаційних систем. Боти на платформі Telegram дозволяють створювати автоматизовані помічники, які можуть виконувати широкий спектр функцій: від відповіді на прості питання до виконання складних завдань, таких як управління базами даних та проведення транзакцій. Завдяки своїй доступності, Telegram-боти стають особливо привабливими для малого та середнього бізнесу, який не має великих ресурсів для розробки власних додатків.

Метою роботи є розробка Telegram-бота для автоматизації процесу продажу одягу, взуття та аксесуарів в інтернет-магазині. Основне завдання бота — надання користувачам можливості перегляду асортименту, збереження інформації про користувачів, забезпечення функцій реєстрації та створення замовлень. Це дозволяє значно спростити процес покупки як для клієнтів, так і для адміністрації магазину, яка може ефективно керувати товарами та замовленнями через інтегровану базу даних.

Для досягнення поставленої мети Telegram-бот забезпечує такі основні функціональні можливості:

— реєстрація користувача з введенням контактної інформації, зокрема номеру телефону та повного імені;

- перегляд каталогу товарів, що розділений на категорії: верхній одяг (верх), низ, взуття та аксесуари;
- можливість ознайомитися з розмірними сітками для одягу та взуття;
- інформація про умови доставки, що дозволяє користувачам вибирати між декількома варіантами доставки;
- можливість оформлення замовлення на обрані товари з подальшою обробкою замовлення менеджерами магазину;

Для реалізації проекту було обрано мову програмування Python, яка є одним із найпоширеніших інструментів для роботи з Telegram API. Додатково використовується SQLite як зручне та легке у налаштуванні рішення для локальної бази даних, що зберігає інформацію про користувачів, товари та замовлення. Така архітектура дозволяє створити автономну систему, що не вимагає значних витрат на серверні ресурси, але при цьому здатна обробляти запити користувачів у реальному часі.

Об'єктом дослідження є програмні рішення для реалізації функціоналу боту. *Предметом дослідження* — можливості автоматизації процесу продажів та реєстрації замовлень через базу даних.

Обсяг та структура роботи. Проєкт складається зі вступу, трьох розділів, висновків, списку використаної літератури та додатків.

1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАВДАННЯ

1.1. Постановка задачі

Потрібно розробити автоматизовану інформаційну систему у вигляді Telegram-бота для магазину одягу. Метою системи є автоматизація обслуговування клієнтів, надання інформації про асортимент товарів, прийом замовлень, а також спрощення процесів управління запасами та комунікації з клієнтами. Система повинна включати такі основні функції:

Каталог товарів:

— Асортимент товарів, розподілений за категоріями (чоловічий, жіночий, дитячий одяг, аксесуари тощо).

Кожен товар характеризується наступними параметрами:

- Назва
- Опис
- Розмірний ряд
- Доступна кількість на складі
- Ціна
- Зображення товару

Реєстрація користувачів:

Користувачі можуть зареєструватися, надаючи контактні дані (ім'я, прізвище, номер телефону) для швидшого оформлення замовлень.

Замовлення:

Користувачі можуть додавати товари до кошика, переглядати його вміст, оформляти замовлення, обираючи спосіб доставки (самовивіз чи кур'єрська доставка) та оплати (готівкою при отриманні або онлайн-оплата).

Після оформлення замовлення клієнт отримує підтвердження із деталями замовлення та приблизною датою доставки.

Системні обмеження:

- Максимальна кількість товарів в одному замовленні – 20 позицій.
- Реєстрація користувачів є обов'язковою для здійснення покупок.

Система має підтримувати можливість одночасної обробки великої кількості запитів.

З системою будуть працювати такі групи користувачів:

— Адміністратори магазину:

1. Додавати нові товари, редагувати інформацію про них (ціна, кількість на складі, розміри тощо).
2. Переглядати та обробляти замовлення, змінювати статус замовлення.
3. Мати доступ до звітів щодо продажів, запасів, популярних товарів.

— Клієнти:

1. Переглядати каталог товарів.
2. Оформляти замовлення.
3. Отримувати повідомлення про статус замовлення та можливість відслідковування доставки.

1.2. Розробка прототипу інтерфейсу

Прототипування – один з ранніх етапів створення сайту.

Прототип полегшує спілкування між розробниками, конкретизує техзавдання і допомагає в плануванні подальшої роботи.

Прототип – це інтерактивний начерк, чернетка майбутнього програмного продукту. На ньому схематично зображуються основні елементи і їх відгук на дії користувача.

Прототип використовують на ранніх стадіях розробки для презентації ідеї зацікавленим сторонам.

Для проектування прототипів був використаний сервіс Figma.

Figma – це хмарний багатоплатформовий сервіс для дизайнерів інтерфейсів і web-розробників, з яким можна працювати безпосередньо в браузері.

Результатом є прототипи програмного продукту. Проаналізувавши вимоги до зовнішнього вигляду програмного продукту, було визначено, що додаток буде складатися з таких компонентів:

Початковий екран взаємодії користувача з ботом (Рисунок 1.1).

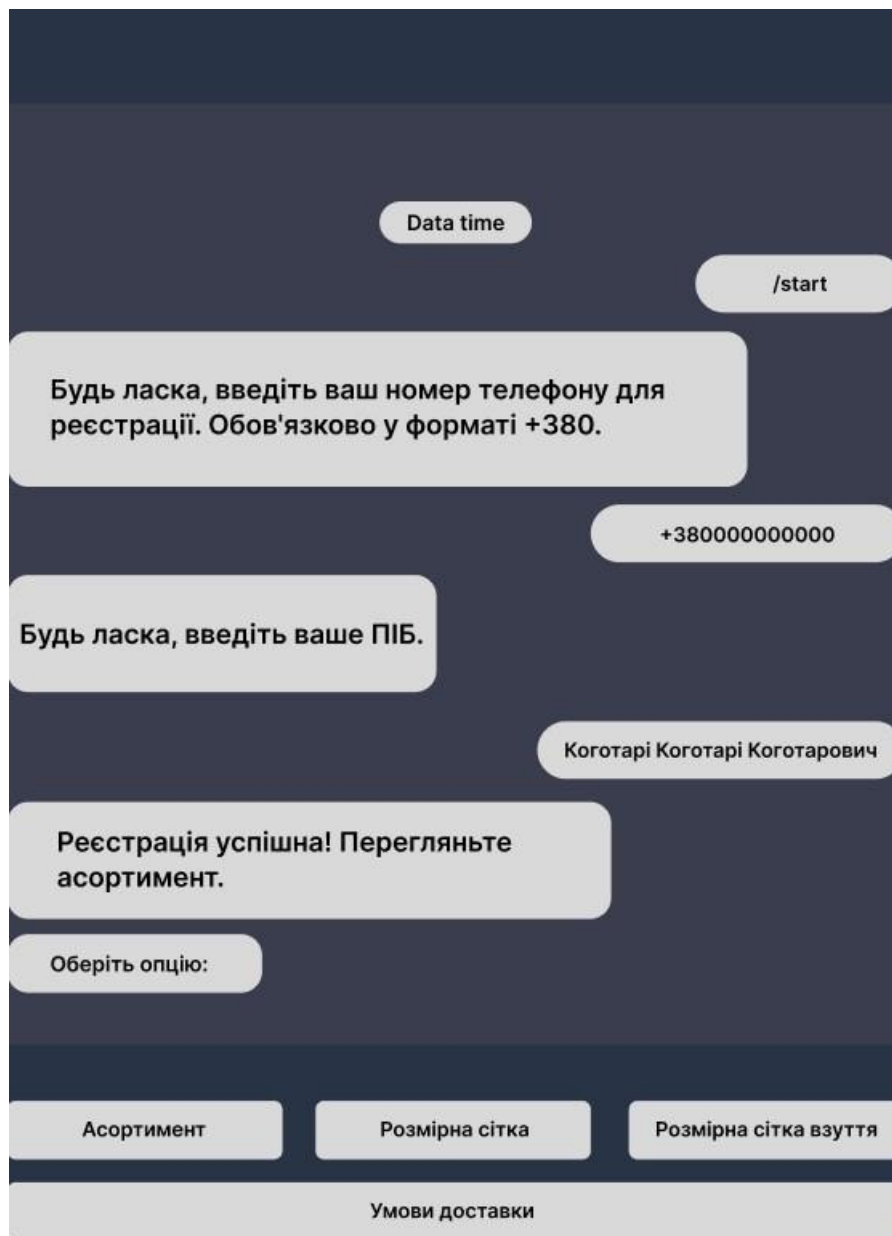


Рисунок 1.1 – Перші кроки для взаємодії

Натискання на кнопку «Умови доставки» (Рисунок 1.2).

Будь ласка, введіть ваш номер телефону для реєстрації. Обов'язково у форматі +380.

+3800000000000

Будь ласка, введіть ваше ПІБ.

Коготарі Коготарі Коготарович

Реєстрація успішна! Перегляньте асортимент.

Оберіть опцію:

Умови доставки

Умови доставки:

1. Укрпошта: стандартна доставка (3-7 днів).
2. Нова Пошта: швидка доставка (1-2 дні).
3. Міст Пошта: доставка до відділення (2-5 днів).

Вартість доставки залежить від компанії та місця призначення.

Асортимент Розмірна сітка Розмірна сітка взуття

Умови доставки

Рисунок 1.2 – Опис «Умови доставки»

Натискання на кнопку «Розмірна сітка» (Рисунок 1.3).

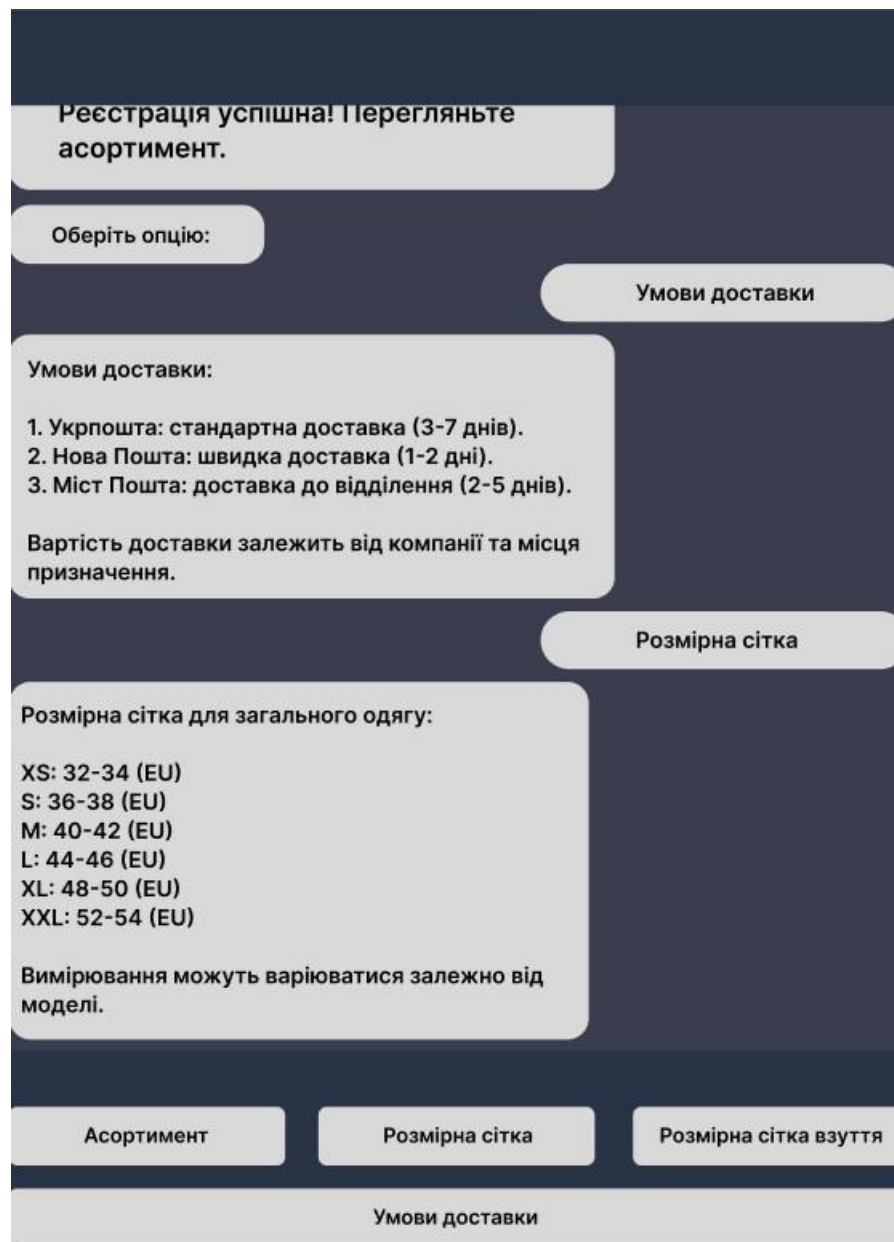


Рисунок 1.3 – Опис «Розмірна сітка»

Натискання на кнопку «Асортимент», висвітлення нових кнопок для перегляду одягу, а саме – верхній одяг, нижній одяг, взуття та аксесуари. Наявність кнопки замовлення товару під повідомленням з описом та фотографією товару (Рисунок 1.4).

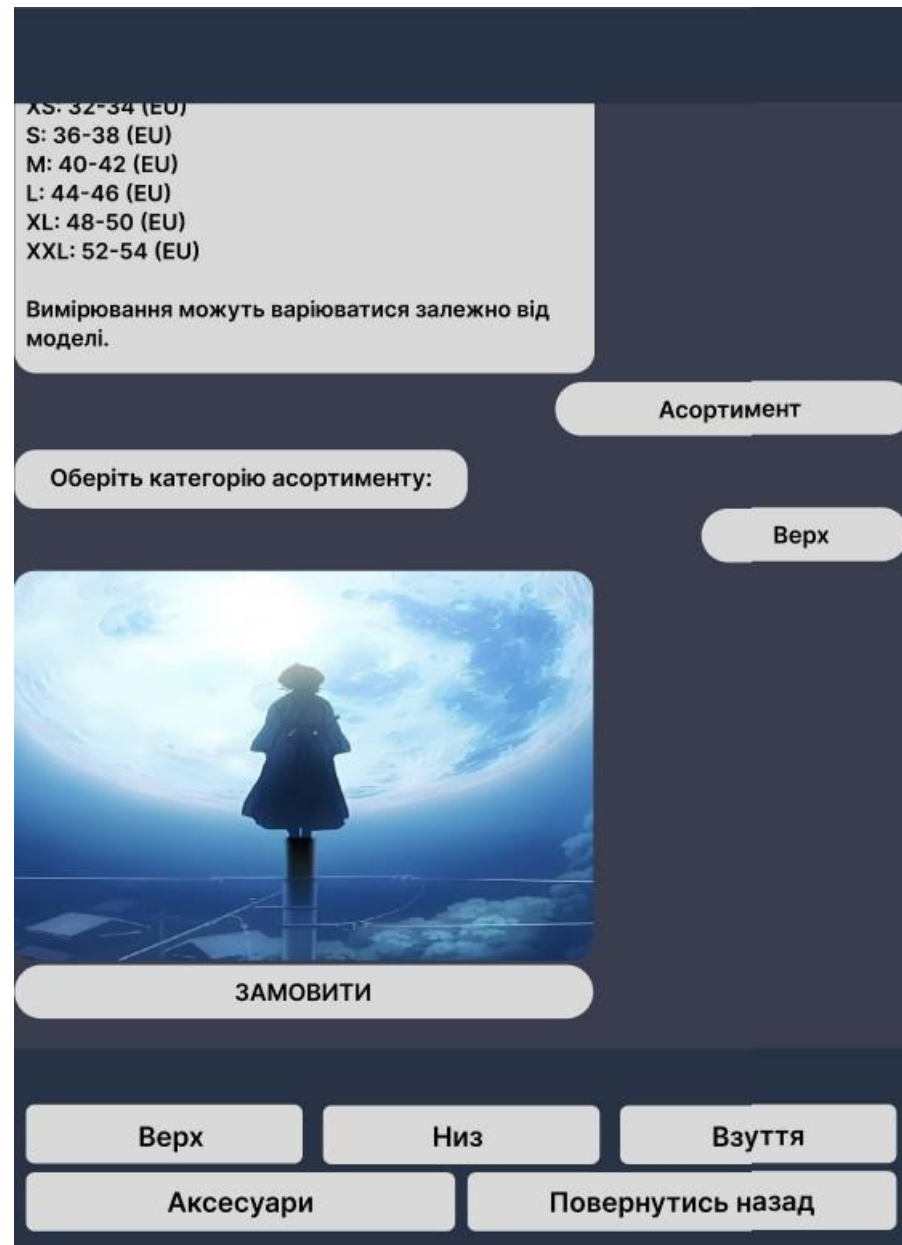


Рисунок 1.4 – Нові кнопки після натискання на асортимент, наявність кнопки повернення назад і кнопка замовлення під товаром.

1.3. Опис та побудова діаграми прецедентів роботи з системою

Нижче зображена діаграма прецедентів, яка описує взаємодію акторів – користувача, адміністратора та бота – із системою Telegram Bot. Користувач

та адміністратор виконують різні дії в межах функціоналу бота, який забезпечує обробку команд, реєстрацію користувачів, показ товарів та інші операції. Користувач має доступ до функцій, пов'язаних з переглядом та оформленням замовлень, тоді як адміністратор має можливість управляти замовленнями та наповненням бази даних (Рисунок 1.5)

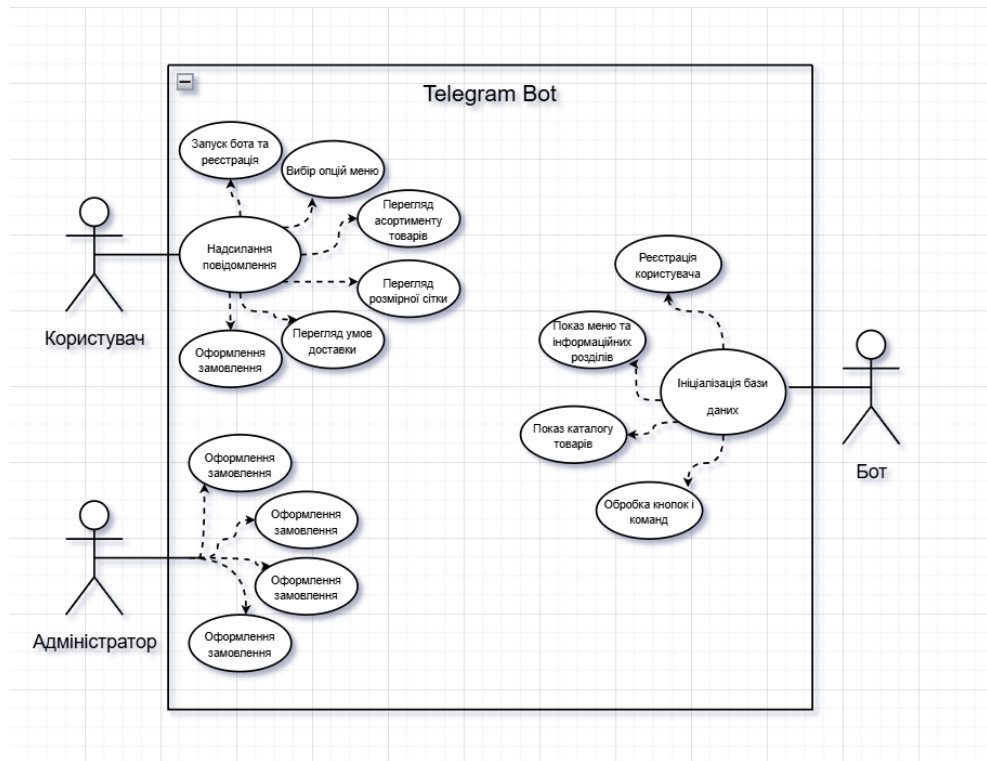


Рисунок 1.5 – Діаграма прецедентів

ПРЕЦЕДЕНТИ ДЛЯ КОРИСТУВАЧА:

1. Запуск бота та реєстрація

- Передумова: Користувач не зареєстрований у системі.
- Післяумова: Користувач зареєстрований і може продовжувати користуватися ботом.
- Сценарій:

1. Користувач надсилає команду /start.
2. Бот перевіряє реєстрацію користувача.
3. Якщо користувача не зареєстровано, бот просить надати номер телефону.
4. Користувач надсилає свій номер у форматі +380....
5. Бот запитує повне ім'я користувача.

6. Користувач вводить своє повне ім'я, і бот завершить реєстрацію, додаючи інформацію до бази даних.

2. Вибір опцій меню

— Передумова: Користувач зареєстрований і готовий до подальшої взаємодії.

— Післяумова: Користувач отримав доступ до потрібної інформації.

— Сценарій:

1. Користувач бачить меню з різними опціями: «Асортимент», «Розмірна сітка», «Розмірна сітка для взуття», «Умови доставки».
2. Користувач обирає одну з опцій, і бот надає відповідну інформацію або запитує подальший вибір.

3. Перегляд асортименту товарів

— Передумова: Користувач знаходиться у головному меню та обирає «Асортимент».

— Післяумова: Користувач переглядає товари вибраної категорії.

— Сценарій:

1. Користувач обирає «Асортимент» у меню.
2. Бот показує категорії: «ВЕРХ», «НИЗ», «ВЗУТТЯ», «АКСЕСУАРИ».
3. Користувач обирає категорію, яка його цікавить.
4. Бот надсилає список товарів у цій категорії з фото, описом і ціною.
5. Користувач може переглянути деталі кожного товару.

4. Перегляд розмірної сітки

— Передумова: Користувач хоче дізнатися розмірну сітку одягу або взуття.

— Післяумова: Користувач отримав потрібну інформацію для вибору розміру.

— Сценарій:

1. Користувач обирає «Розмірна сітка» або «Розмірна сітка для взуття».
2. Бот надсилає розмірну таблицю з детальними параметрами розмірів.

3. Користувач аналізує дані та визначає потрібний розмір.

5. Перегляд умов доставки

— Передумова: Користувач хоче дізнатися умови доставки.

— Післяумова: Користувач отримав детальну інформацію про можливі варіанти доставки.

— Сценарій:

1. Користувач обирає «Умови доставки».

2. Бот надсилає перелік доступних способів доставки, строки та вартість.

3. Користувач обирає найбільш підходящий варіант для себе.

6. Оформлення замовлення

— Передумова: Користувач обрав товар і хоче його замовити.

— Післяумова: Користувач отримує підтвердження про додавання товару до кошика та очікує на зворотний зв'язок.

— Сценарій:

1. Користувач обирає товар із категорії та натискає на кнопку «Замовити».

2. Бот підтверджує додавання товару до кошика і надсилає повідомлення з подальшими інструкціями.

3. Користувач очікує на зворотний зв'язок від менеджера.

ПРЕЦЕДЕНТИ ДЛЯ АДМІНІСТРАТОРА:

1. Додавання нових товарів

— Передумова: Потрібно додати нові товари до бази даних, щоб користувачі могли їх переглядати.

— Післяумова: Нові товари з'являються в базі даних та доступні для користувачів через меню бота.

— Сценарій:

1. Адміністратор формує нові записи для кожного товару із зазначенням категорії, назви, фото, опису та ціни.

2. Адміністратор оновлює базу даних, додаючи нові записи для товарів.

3. Користувачі можуть переглядати додані товари через меню бота.

2. Оновлення інформації про товари

— Передумова: Потрібно внести зміни в інформацію про товари, наприклад, змінити ціну, опис, або зображення товару.

— Післяумова: Оновлена інформація відображається у бота.

— Сценарій:

1. Адміністратор знаходить товар, інформацію про який потрібно оновити, в базі даних.

2. Адміністратор вносить необхідні зміни, наприклад, оновлює ціну або додає нове фото.

3. База даних зберігається з актуальною інформацією, яку бот відображатиме користувачам.

3. Видалення товарів

— Передумова: Потрібно видалити товар, що більше не доступний, щоб уникнути замовлень.

— Післяумова: Товар більше не відображається користувачам у бота.

— Сценарій:

1. Адміністратор визначає товар, який потрібно видалити.

2. Адміністратор видаляє запис про товар з бази даних.

3. Користувачі не бачать видалений товар у списку доступних.

4. Обробка замовлень

— Передумова: Користувачі зробили замовлення через бота, і їх потрібно опрацювати.

— Післяумова: Замовлення користувачів оброблені, і надано відповідь з підтвердженням або уточненням.

— Сценарій:

1. Адміністратор переглядає нові замовлення, що зберігаються в базі даних.

2. Адміністратор зв'язується з користувачем для підтвердження деталей замовлення.
3. Адміністратор змінює статус замовлення в базі даних, якщо потрібно.

ПРЕЦЕДЕНТИ ДЛЯ БОТА:

1. Ініціалізація бази даних

- Передумова: Система запускається вперше, або базу даних потрібно оновити.
- Післяумова: Усі таблиці створені, а зразки товарів додані до каталогу.
- Сценарій:
 1. Бот викликає функцію `init_db()`, яка перевіряє наявність таблиць `users`, `products`, і `orders`.
 2. Якщо таблиць немає, вони створюються з потрібною структурою.
 3. Функція `add_sample_products()` додає зразки товарів у таблицю `products`.
 4. База даних готова для подальшої роботи з користувачами.

2. Перевірка реєстрації користувача

- Передумова: Користувач надсилає команду `/start`.
- Післяумова: Бот визначив, чи користувач вже зареєстрований, і діє відповідно.
- Сценарій:
 1. Бот отримує команду `/start` від користувача.
 2. Він перевіряє наявність користувача у таблиці `users`.
 3. Якщо користувач є і зареєстрований, бот пропонує продовжити роботу.
 4. Якщо користувач не зареєстрований, бот запитує номер телефону та ПІБ.

3. Показ меню та інформаційних розділів

- Передумова: Користувач уже зареєстрований і взаємодіє з ботом.

— Післяумова: Користувач отримав запитану інформацію або доступ до каталогу.

— Сценарій:

1. Бот пропонує користувачеві основне меню, яке містить опції для перегляду каталогу, таблиці розмірів та умов доставки.
2. Користувач обирає одну з опцій, і бот надсилає відповідну інформацію.

4. Показ каталогу товарів

— Передумова: Користувач обирає категорію з асортименту.

— Післяумова: Користувач отримує перелік товарів із фото, описом і ціною.

— Сценарій:

1. Бот отримує запит на перегляд асортименту і надсилає категорії товарів.
2. Користувач обирає категорію, і бот витягує відповідні записи з бази.
3. Бот надсилає фото товарів, описи та ціни, додаючи кнопку для замовлення.

5. Обробка кнопок і команд

— Передумова: Користувач натискає кнопку чи вводить команду.

— Післяумова: Бот надає відповідь або виконує необхідну дію.

— Сценарій:

1. Бот отримує команду або натискання кнопки.
2. Відповідно до команди викликається відповідна функція, наприклад, для перегляду розмірної сітки, умов доставки чи повернення до головного меню.
3. Бот надсилає відповідь на запит.

1.4. Інфологічне проєктування – схема бази даних

Для створення інформаційної системи, що автоматизує процес обслуговування клієнтів у контексті телеграм-бота для інтернет-магазину,

дані, що використовуватимуться в системі, зберігатимуться у базі даних (БД). У БД виділено п'ять основних інформаційних об'єктів (відношень): користувачі, товари, замовлення, варіанти доставки та методи оплати.

Розглянемо структуру таблиць, що описують ці сутності, і розробимо модель даних «сутність-зв'язок».

- Таблиця «Користувачі» зберігає відомості про користувачів, які зареєструвались у системі. Вона містить такі поля: ідентифікатор користувача, номер телефону, ПІБ, а також статус реєстрації для перевірки, чи завершена реєстрація користувача. Первинний ключ – поле Ідентифікатор користувача.
- Таблиця «Товари» описує товари, що доступні в каталозі. Поля таблиці включають категорію, назву, фото товару, опис, ціну. Кожен товар має унікальний ідентифікатор, який виступає первинним ключем для забезпечення цілісності даних.
- Таблиця «Замовлення» зберігає інформацію про зроблені замовлення. Вона містить ідентифікатор замовлення, ідентифікатор користувача, ідентифікатор товару, статус замовлення (за замовчуванням "очікує") та інформацію про опції доставки. Первинним ключем є ідентифікатор замовлення, а зовнішні ключі – ідентифікатори користувача і товару, що вказують на зв'язок між користувачем, який зробив замовлення, і замовленим товаром.

Такий склад інформаційних об'єктів забезпечує можливість обробляти інформацію про користувачів, асортимент товарів, а також зберігати дані про замовлення та їх статуси.

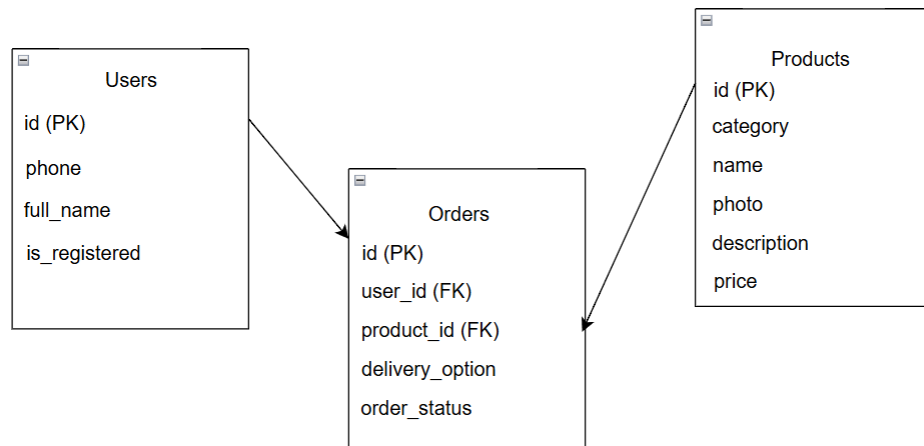


Рисунок 1.6 – Схема бази даних

Схема бази даних (Рисунок 1.6) показує взаємозв'язок між таблицями та відображає ключові атрибути кожної з них.

2. ОПИС ПРОГРАМНИХ ЗАСОБІВ

Мова програмування Python. Python був обраний для реалізації проекту завдяки своїй простоті, надійності та великому вибору бібліотек, які полегшують розробку. Мова ідеально підходить для написання телеграм-ботів завдяки таким бібліотекам, як *telebot* та інші, що дозволяють легко обробляти повідомлення, працювати з API та базами даних.

Бібліотека TeleBot (pyTelegramBotAPI). Призначення: TeleBot використовується для створення телеграм-бота, який може приймати команди від користувачів, обробляти повідомлення, відправляти відповіді та працювати з інтерактивними кнопками.

Застосування в проекті:

- Обробка команд, таких як */start*.
- Взаємодія з користувачем для отримання даних (номер телефону, ім'я тощо).
- Відображення меню, асортименту товарів та умов доставки.

База даних SQLite. Призначення: SQLite обрана як проста та зручна у використанні вбудована база даних для зберігання інформації про користувачів, товари та замовлення.

Застосування в проекті:

- Зберігання даних користувачів для їх ідентифікації та відстеження реєстрації.
- Зберігання товарів із зазначенням категорії, опису, фото та ціни.
- Ведення інформації про замовлення, що дозволяє фіксувати товари, обрані користувачами.

Бібліотека SQLite3. Призначення: стандартна бібліотека для роботи з базою даних SQLite, яка дозволяє виконувати SQL-запити до бази даних, створювати таблиці та маніпулювати даними.

Застосування в проекті:

- Ініціалізація бази даних та створення таблиць (users, products, orders).
- Виконання запитів для зчитування, додавання та оновлення інформації в базі даних, а також управління інформацією про користувачів та товари.

Бібліотека Time. Призначення: стандартна бібліотека Python для роботи з часом.

Застосування в проекті: використовується для створення пауз між повідомленнями, що покращує сприйняття інформації користувачем (плавність відображення товарів при перегляді асортименту).

Середовище розробки Visual Studio Code (VS Code). Призначення: VS Code обрано для розробки завдяки його зручному інтерфейсу, підтримці плагінів для роботи з Python, системами контролю версій (Git) та інтеграції з платформою для телеграм-ботів.

Застосування в проекті: розробка коду бота, налагодження, тестування та організація структури проекту.

Дизайн та інтерфейс. Для інтерфейсу бота використовувалися вбудовані інструменти Telegram (кнопки меню та inline-кнопки), що дозволяють користувачеві взаємодіяти з ботом без необхідності ручного введення команд.

Інструменти Telegram API. Використовуються для обробки команд, налаштування інтерфейсу, створення клавіатур, інлайн-кнопок для зручної взаємодії з ботом та перегляду товарів.

Google Forms (замість платіжної системи і кнопки замовлення). Призначення: використовується для обробки замовлень, оскільки в проекті немає інтеграції з платіжною системою і не фізична inline кнопка замовлення у проекті. Google Forms дозволяє користувачам заповнити форму для замовлення та надання контактних даних.

Застосування в проекті: посилання на форму вбудовується в бота під кінець перегляду асортименту, як спосіб замовлення товару.

Ці програмні засоби створюють надійну основу для розробки телеграм-бота, дозволяючи зберігати інформацію в базі даних, приймати дані від користувачів, керувати асортиментом товарів та обробляти замовлення.

3. РЕАЛІЗАЦІЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ ПРОЕКТУВАННЯ

3.1. Алгоритм програми, архітектура програми

Алгоритм програми

Програма призначена для обробки замовлень у Telegram через бот, який взаємодіє з базою даних SQLite та Google-формою для завершення замовлення. Основний алгоритм програми можна поділити на кілька етапів:

- Запуск та ініціалізація.
- Реєстрація користувача.
- Відображення асортименту.
- Оформлення замовлення.

Архітектура програми складається з кількох основних компонентів:

- Telegram-бот.
- База даних SQLite.
- Інтеграція з Google-формою.

3.2. Структура програми з функцій складових частин і зв'язків між ними

Структура програми, функції, складові частини та зв'язки показані на діаграмі класів (Рисунок 1.7).

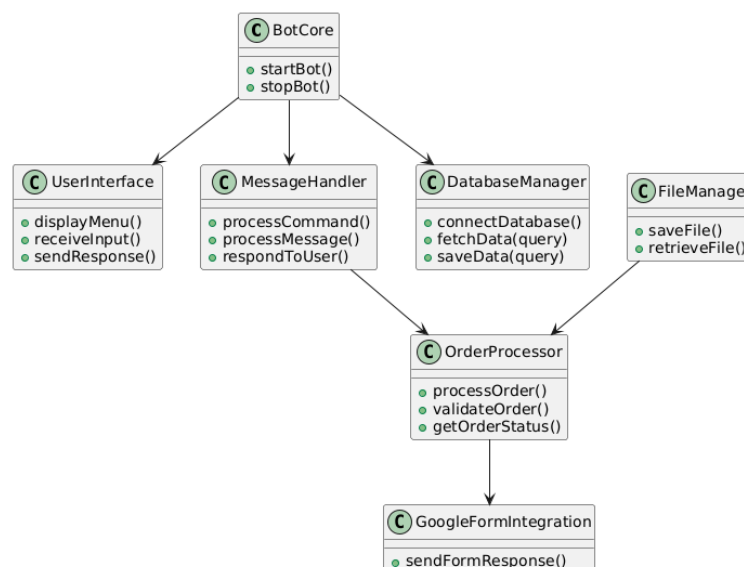


Рисунок 1.7 – Діаграма класів

3.3. Виклик та завантаження

Програма розроблена як Python-скрипт, який виконується з використанням середовища Python або Visual Studio Code, а не як окремий виконувальний файл. Для зручності використання, скрипт разом із необхідними залежностями можна зберегти в архіві, що полегшить його розповсюдження та захистить від випадкових змін компонентів програми. Користувач повинен завантажити та розпакувати архів, після чого програма готова до запуску. Активувати самого бота потрібно запустивши код, який буде у .ру файлі. Поки активне середовище, поки й функціонує бот.

3.4. Інструкція користувачу

Інструкція по роботі з програмою

Запускаємо бота. У користувача знизу буде команда /start. Після натискання, почнеться процес реєстрації. Потрібно буде ввести номер телефону (Рисунок 1.8).

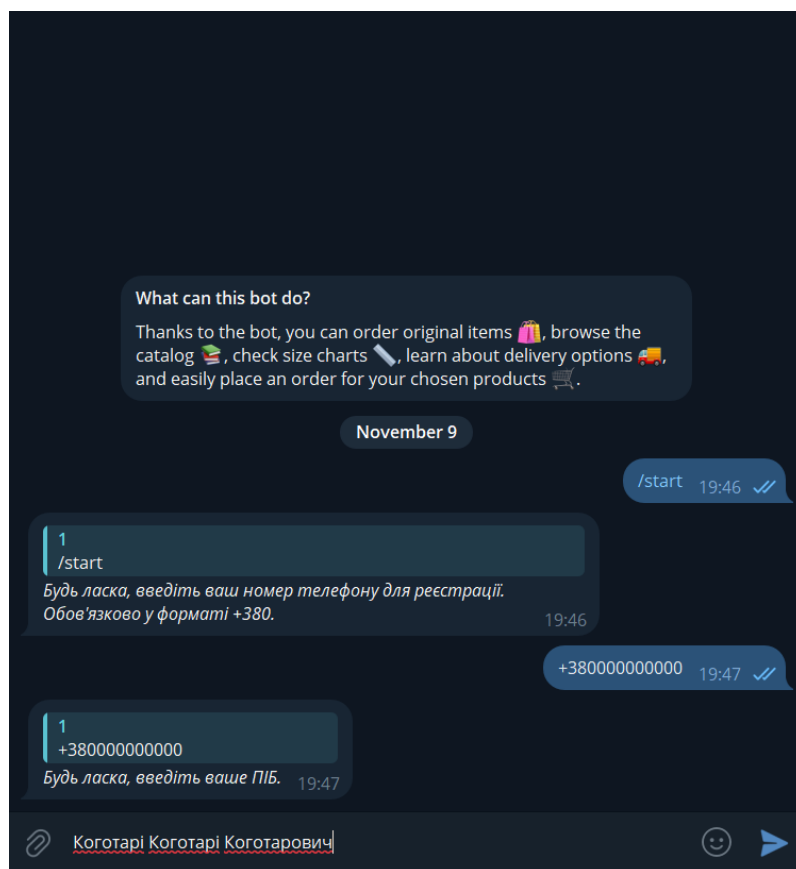


Рисунок 1.8 – Початок роботи, запуск бота, введення номеру

Після введення номера телефона, клієнту потрібно буде ввести своє ПІБ. Обмежень немає, тому клієнт може ввести що забажає, бо все одно для підтвердження замовлення з клієнтом буде зв'язуватися менеджер магазину (Рисунок 1.9).

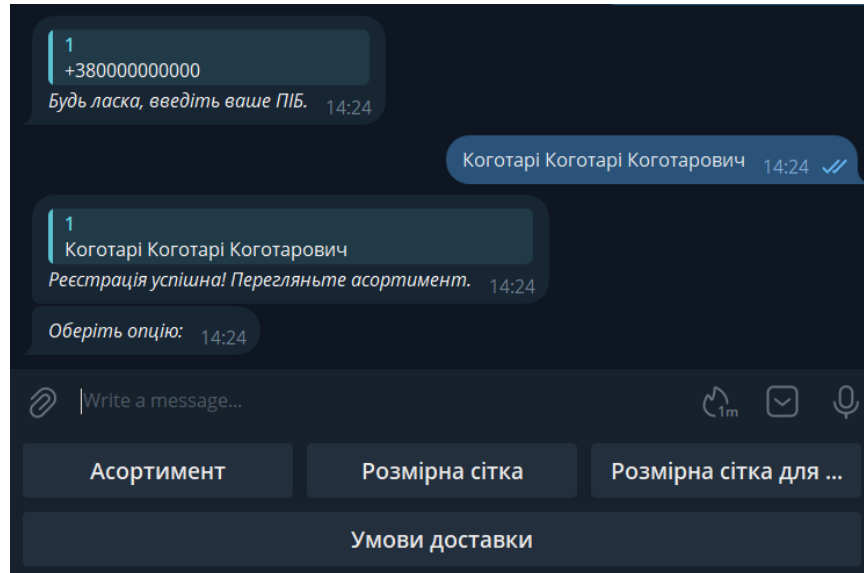


Рисунок 1.9 – Введення ПІБ, повідомлення щодо реєстрації, виведення кнопок інтерфейсу користувачу, обирання опції Вибір опції «Умови доставки» (Рисунок 1.10).

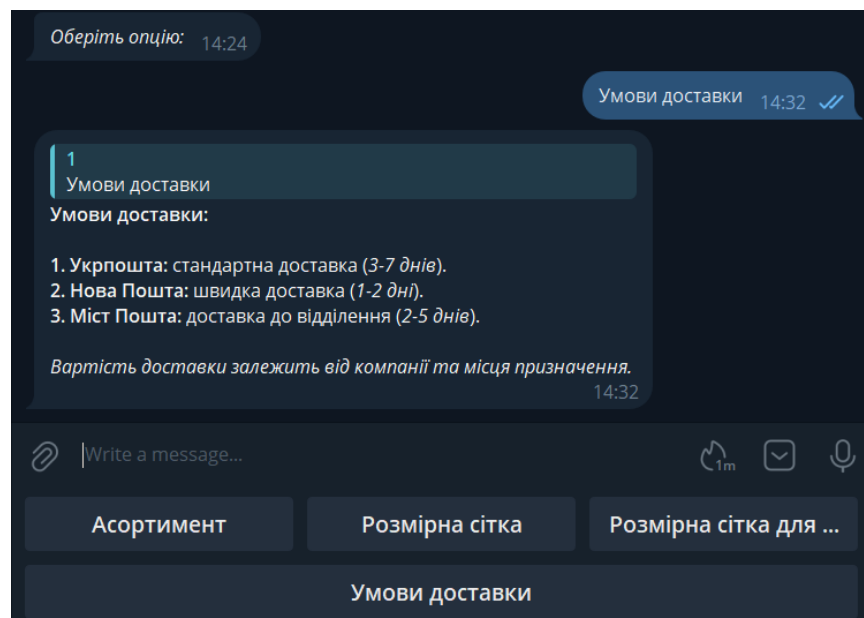


Рисунок 1.10 – Кнопка «Умови доставки»

Вибір опції «Розмірна сітка» (Рисунок 1.11).

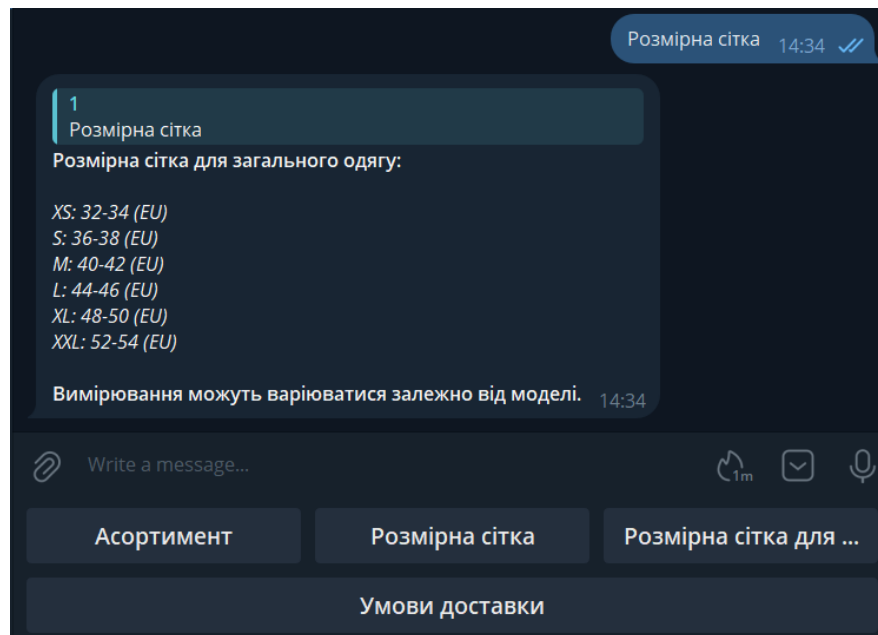


Рисунок 1.11 – Кнопка «Розмірна сітка»

Вибір опції «Асортимент». Після натискання з'являються нові кнопки для перегляду бажаної категорії товару, можливість повернутись назад до головного меню (Рисунок 1.12).

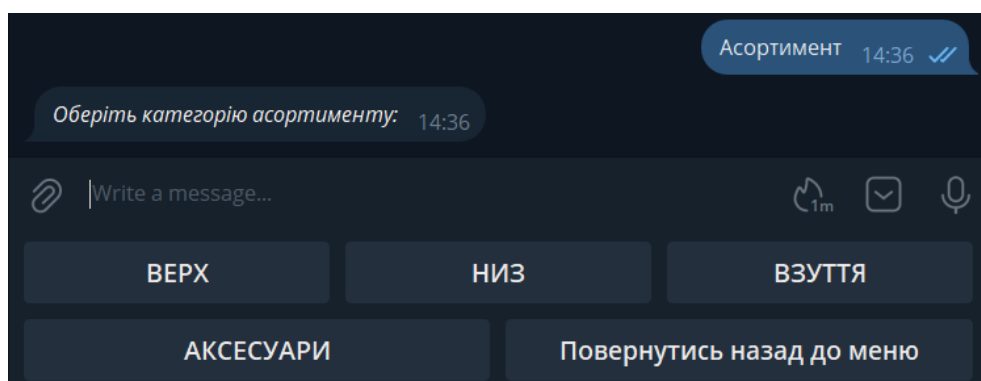


Рисунок 1.12 – Кнопка «Асортимент»

Натискання на бажану категорію товару, інлайн-кнопка замовлення (Рисунок 1.13).



Рисунок 1.13 – перегляд товару, його вартість, опис, інлайн кнопка замовлення

ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

Було завдання створити телеграм-бота для онлайн-замовлень, який має забезпечити користувачам можливість переглядати асортимент товарів, вибирати необхідні позиції, та робити замовлення через Google Forms. Розроблений бот, який включає функціональність реєстрації користувачів, відображення каталогу товарів з категоріями, замовлення товару, взаємодію через зручний інтерфейс у Telegram, а також інтеграцію з базою даних SQLite для зберігання інформації про користувачів.

Цей бот може використовуватися невеликими бізнесами (ФОП або ж приватні особи, які тільки починають працювати у сфері торгівлі одягу) для обробки замовлень на товари чи послуги, де зручний інтерфейс замовлення через месенджер полегшує процес взаємодії для користувачів. Бот надає можливість легко створювати замовлення і має просту інфраструктуру, що робить його привабливим рішенням для малого бізнесу.

Завдання було виконане повністю – бот працює, забезпечуючи необхідні функції, та дозволяє користувачам переглядати асортимент, вибирати товари й оформлювати замовлення, але за допомогою анкети Google Forms. Система надає злагоджену функціональність, зручний інтерфейс для користувача й базові можливості для управління замовленнями з боку адміністрації.

Під час роботи застосовано знання з таких дисциплін, як:

- Програмування на Python – для створення основного функціоналу бота.
- Базы даних – для роботи з SQLite, зокрема структурування даних і управління запитам.
- Інтерфейс користувача та UX-дизайн – для створення інтуїтивного інтерфейсу в Telegram.
- Мережеві технології – знання основ API, що допомогло ефективно працювати з TeleBot. Вдосконалено навички роботи з базами даних, інтеграції сторонніх сервісів (Google Forms), а також навчився ефективно розробляти інтерфейс користувача для ботів, що покращило мої знання з

UX. Також отримано практичний досвід розробки проекту з інтеграцією бібліотек для Telegram.

Наступними кроками може бути розширення функціональності бота, зокрема, додавання модуля для прийому платежів, повна функціональність inline-кнопки замовлення товару, а також додаткових функцій для адміністрування замовлень. Розширення можливостей бота може відкрити йому доступ до більш широкого спектру користувачів та підвищити його корисність для малого бізнесу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Python [Електронний ресурс] – Режим доступу до ресурсу: <https://www.python.org/>
2. SQLAlchemy [Електронний ресурс] – Режим доступу до ресурсу: <https://www.sqlalchemy.org/>
3. SQLite Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://sqlite.org/docs.html>
4. Figma [Електронний ресурс] – Режим доступу до ресурсу: <https://www.figma.com>
5. pyTelegramBotAPI [Електронний ресурс] – Режим доступу до ресурсу: <https://pypi.org/project/pyTelegramBotAPI/>
6. Python Telegram Bot Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://python-telegram-bot.readthedocs.io/>
7. Telegram API Support in Python [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/python-telegram-bot>
8. W3Schools SQL Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com/sql/>

ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ

1. ПРИЗНАЧЕННЯ І МЕТА СТВОРЮВАНОЇ СИСТЕМИ

1.1. Загальні відомості

Повна назва системи: Автоматизована інформаційна система для обробки замовлень у магазині "Vivienne_shop". Коротка назва: АІС "Vivienne_shop".

Робота виконується на основі завдання до курсового проекту з дисципліни Проектування автоматизованих інформаційних систем.

Термін виконання роботи: 1 вересня 2024 року – 10 листопада 2024 року.

Роботи по створенню АІС "Vivienne_shop" здаються поетапно згідно з календарним планом.

1.2. Призначення системи

АІС "Vivienne_shop" повинна обробляти інформацію про користувачів, товари, замовлення, а також забезпечувати автоматизований процес реєстрації користувачів, вибір товарів через Telegram-бота.

Функціональність АІС:

- реєстрація користувачів, збір контактної інформації (телефон, ПІБ);
- перегляд асортименту товарів;
- можливість вибрати товар;
- інтерфейс користувача через Telegram-бота для вибору товарів, перегляду розмірних сіток і умов доставки;
- можливість повернення до основного меню через інтерфейс бота;
- можливість замовлення товару.

1.3. Мета створення

Мета створення АІС "Vivienne_shop" — це автоматизація процесу управління магазином, включаючи:

- реєстрацію та зберігання інформації про клієнтів;
- зручність користувачів через доступ до асортименту, інформації про товар, розмірні сітки та умови доставки;

2. ХАРАКТЕРИСТИКА ОБ'ЄКТІВ АВТОМАТИЗАЦІЇ ТА ВИМОГИ ДО СИСТЕМИ

2.1. Відомості про об'єкт автоматизації

Об'єктом автоматизації є:

- процес реєстрації користувачів магазину, включаючи введення номера телефону та ПІБ;
- процес взаємодії з користувачами для отримання інформації про умови доставки, розмірні сітки та асортимент товарів;
- процес формування замовлень клієнтів через Telegram-бота.

Особливих умов до експлуатації АІС не висувається

2.2. Вимоги до системи в цілому

АІС "Vivienne_shop" повинна бути централізованою, тобто всі дані про користувачів, товари та замовлення повинні зберігатися в єдиній базі даних, яка доступна через інтеграцію з Telegram-ботом. Система складається з наступних підсистем:

- підсистема збору, обробки та завантаження даних: збір інформації від користувачів (реєстрація, замовлення), обробка запитів від бота;
- підсистема зберігання даних: зберігання інформації про користувачів в базі даних SQLite.;
- підсистема формування та візуалізації звітності: забезпечення безперервної взаємодії з користувачами через Telegram-бота для перегляду асортименту.

2.3. Вимоги до функцій (задач), що виконуються системою

Система повинна забезпечувати:

- роботу користувачів режимі - 24 годин на день, 7 днів на тиждень (24x7);
- виконання своїх функцій – обробка та зберігання даних, включаючи реєстрацію користувачів та товарів;

2.4. Вимоги до надійності

Програмний продукт повинен працювати стабільно в середовищі Telegram. У разі збоїв або перевантажень на сервері, інформація про

користувачів і замовлення повинна зберігатися в базі даних і не втрачається, але можливе тимчасове відключення бота.

2.5. Вимоги до інформаційної і програмної сумісності

Для розробки використовувати базу даних SQLite, об'єктно-орієнтоване середовище Visual Studio Code, мову програмування Python, бібліотеку pyTelegramBotAPI і обов'язково для взаємодії з користувачами, система повинна бути сумісною з Telegram.

2.6. Спеціальні вимоги

Інтерфейс:

- Бот повинен мати простий і зручний інтерфейс для користувачів, який забезпечує зрозуміле меню для вибору товарів, перегляду розмірних сіток та умов доставки.
- Масштабованість: Система повинна бути здатна обробляти великий обсяг замовлень одночасно без значного уповільнення роботи.

3. СКЛАД І ЗМІСТ РОБІТ ПО СТВОРЕННЮ СИСТЕМИ

Роботи по створенню системи виконуються в три етапи:

- технічне завдання;
- розробка проекту і проектної документації;
- захист проекту.

На етапі «Технічне завдання» має бути виконаний етап розробки, і етап затвердження технічного завдання:

- постановка завдання;
- визначення і уточнення вимог до технічних засобів;
- визначення вимог до програми;
- визначення стадій, етапів і термінів розробки програми і документації на неї;
- вибір мов програмування;
- узгодження і затвердження технічного завдання.

На етапі «Розробка проекту і проектної документації» мають бути виконані перераховані етапи робіт:

- розробка програми (програмування (кодування) і відладки програми);
- розробка програмної документації (розробка пояснювальної записки, технічного завдання, опису програми відповідно до вимог чинних методичних рекомендацій);
- випробування програми (тестування програми; коригування програми і програмної документації за результатами тестування).

На етапі захисту проекту має бути виконана розробка презентації курсового проекту і представлення його комісії.

4. ПОРЯДОК КОНТРОЛЮ І ПРИЙМАННЯ СИСТЕМИ

При прийманні курсового проекту перевіряється – комплектність, зміст і оформлення програмної документації.

Захист проводиться перед комісією з трьох викладачів у присутності студентів поточної академічної групи.

Курсовий проект оцінюється за такими критеріями:

- якість аналізу предметної області та побудови концептуальної моделі;
- рівень якості поданої програмної розробки (повнота реалізованих функцій, рівень інтерфейсу, наявність можливостей налаштування, стійкість та надійність функціонування, наявність засобів допомоги тощо);
- якість побудови ER-діаграми, діаграми прецедентів;
- якість побудови логічної моделі бази даних та проведеної нормалізації;
- практична цінність проектних рішень (відповідність реальним умовам об'єкта, універсальність та оригінальність прийнятих рішень);
- відповідність оформлення курсового проекту встановленим вимогам, дотримання встановлених стандартів;
- своєчасність виконання графіка робіт при проектуванні та поданні курсового проекту;
- виступ та презентація за основними результатами;
- відповіді на запитання.

ДОДАТОК Б. ЛІСТИНГ ПРОГРАМИ

```

import sqlite3
# функція для ініціалізації бази даних
def init_db():
    with sqlite3.connect('database.db') as db:
        cursor = db.cursor()
        cursor.executescript("""
            CREATE TABLE IF NOT EXISTS users (
                id INTEGER PRIMARY KEY,
                phone TEXT,
                full_name TEXT,
                is_registered INTEGER DEFAULT 0
            );
            CREATE TABLE IF NOT EXISTS products (
                id INTEGER PRIMARY KEY,
                category TEXT,
                name TEXT,
                photo TEXT,
                description TEXT,
                price REAL
            );
            CREATE TABLE IF NOT EXISTS orders (
                id INTEGER PRIMARY KEY,
                user_id INTEGER,
                product_id INTEGER,
                delivery_option TEXT,
                payment_method TEXT,
                order_status TEXT DEFAULT 'pending',
                FOREIGN KEY (user_id) REFERENCES users(id),
                FOREIGN KEY (product_id) REFERENCES products(id)
            );
        """)
        add_sample_products() # зразкові товари
    db.commit()

def add_sample_products():
    with sqlite3.connect('database.db') as db:
        cursor = db.cursor()
        cursor.executescript("""
            delete from products;
            INSERT INTO products (category, name, photo, description, price) VALUES
                -- ваші товари тут...
        """)
    db.commit()

# команда /start, перевірка реєстрації користувача
@bot.message_handler(commands=['start'])
def start_command(message):
    user_id = message.from_user.id
    with sqlite3.connect('database.db') as db:
        cursor = db.cursor()

```

```

        cursor.execute("SELECT * FROM users WHERE id=?", (user_id,))
        user = cursor.fetchone()
        # обробка даних користувача

# збереження даних користувача
def handle_full_name(message, phone):
    full_name = message.text
    user_id = message.from_user.id

    with sqlite3.connect('database.db') as db:
        cursor = db.cursor()
        cursor.execute("INSERT OR REPLACE INTO users (id, phone, full_name,
is_registered) VALUES (?, ?, ?, ?)",
                        (user_id, phone, full_name, 1))
        db.commit()

# вибір категорії та показ товарів
@bot.message_handler(func=lambda message: message.text in ["БЕРХ", "НИЗ",
"ВЗУТТЯ", "АКСЕСУАРИ"])
def show_products(message):
    category = message.text
    with sqlite3.connect('database.db') as db:
        cursor = db.cursor()
        cursor.execute("SELECT * FROM products WHERE category=?", (category,))
        products = cursor.fetchall()
        # обробка даних товарів

```

ДОДАТОК В. РЕПОЗИТОРІЙ ДО ПРОГРАМИ



Рисунок В.1 – Посилання на репозиторій до програми