

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА СИСТЕМИ ЗАМОВЛЕННЯ МОВОЮ
ПРОГРАМУВАННЯ PYTHON ВИКОРИСТОВУЮЧИ TELEGRAM API
»**

Виконав: студент 4 курсу, групи ПД-41
спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності/спеціалізації)

Фурман К. Д.
(прізвище та ініціали)

Керівник _____ Гаманюк І. М.
(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти -«Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ _____ ” _____ 2021 року

З А В Д А Н Н Я НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА

ФУРМАНА КИРИЛА ДМИТРОВИЧА

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка системи замовлення мовою програмування Python використовуючи Telegram Арі»

Керівник роботи: Гаманюк І. М., старший викладач кафедри ІІЗ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «12» березня 2021 року №65.

2. Строк подання студентом роботи «01» червня 2021 року

3. Вхідні дані до роботи

Методи розробки комерційних проєктів;

Науково-технічна література з питань, пов'язаних з програмним забезпеченням щодо розробки комерційних проєктів; Visual Studio Code; Документація Python;

4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити).

4.1 Система обробки замовлень в месенджері Telegram

4.2 Вимоги та оцінка якості системи.

4.3 Моделювання системи.

4.4 Опис проектування системи.

4.5 Опис використаних технологій.

5. Перелік демонстраційного матеріалу (назва основних слайдів)

1. Мета, об'єкт та предмет дослідження
 2. Аналоги
 3. Технічне завдання
 4. Програмні засоби реалізації
 5. Архітектура бази даних
 6. Апробація результатів дослідження
 7. Висновки
6. Дата видачі завдання «19» квітня 2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури та програмних засобів	19.04-21.04	Виконано
2	Створення вимог до системи та моделювання UML діаграмами	22.04-23.04	Виконано
3	Створення концепції та архітектури програмного забезпечення	24.04-25.04	Виконано
4	Розробка програмного забезпечення та усунення недоліків	26.04-28.04	Виконано
5	Моделювання та розробка БД	29.04-30.05	Виконано
6	Вступ, висновки, реферат	01.05-08.05	Виконано
7	Розробка обов'язкових демонстраційних матеріалів	09.05	Виконано
8	Попередній захист роботи	11.05	
9	Здача роботи	01.06	

Студент _____
(підпис)

Фурман К. Д.
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Гаманюк І. М.
(прізвище та ініціали)

Реферат

Текстова частина бакалаврської роботи 45 с., 24 рис., табл. 4, 18 джерел.

Ключеві слова: AIOGRAM, PYTHON, VISUAL STUDIO CODE, GIT, POSTGRESQL, CHAT-BOT, TELEGRAM, TELEGRAM BOT API

Об'єктом дослідження - розробка системи замовлення мовою програмування Python з використанням Telegram Api. Дана система повинна бути реалізована в середині месенджера Telegram, надавати змогу користувачам оформляти та відправляти замовлення, та надсилати до менеджера.

Предмет дослідження - можливість реалізації даної системи в месенджері, з використанням передових інформаційних технологій, та аналіз впливу виходу на нову методику збуту товарів.

Мета роботи – розробити систему замовлень в месенджері для подальшого обчислення показників продажу товарів, виявити можливі недоліки та усунути їх.

Наукова новизна даної роботи полягає в наступному:

- Розроблено алгоритм для ефективної обробки замовлень.
- Проаналізовано ринок месенджерів.
- Встановлено, що для досягнення поставленої задачі доцільно використовувати мову програмування Python та асинхронну бібліотеку aiogram.

Упровадження розробленого застосунку надасть користувачам можливість без завантаження додаткових програмних засобів в зручному режимі оформляти та залишати замовлення. В той час як для замовника це інноваційна поки ще не розповсюджена методика збуту товарів, що дасть змогу отримати додатковий прибуток, збільшити швидкість обробки замовлень, та удосконалити процес обслуговування клієнтів.

В роботі було виконано аналіз існуючих застосунків, встановлено переваги та недоліки. В результаті аналізу було визначено основні потреби користувачів та спроектовано додаток для подальшої розробки.

Зміст

Вступ.....	8
РОЗДІЛ 1 ЕЛЕКТРОНА КОМЕРЦІЯ ТА ВИДИ РЕАЛІЗАЦІЇ.....	13
1.1 Електрона комерція.....	13
1.2 Варіанти реалізації системи замовлень	14
1.2.1 Web-site	14
1.2.2 Мобільні додатки	15
1.2.3 Месенджер	16
1.3 Месенджер	17
РОЗДІЛ 2 АНАЛІЗ НАЯВНИХ ЗАСОБІВ ТА ТЕХНОЛОГІЙ ОРГАНІЗАЦІЇ ПРОДАЖУ ТОВАРІВ ЧЕРЕЗ МЕРЕЖУ ІНТЕРНЕТ.....	20
2.1 Вибір середовища розробки.....	20
2.2 Visual Studio Code.....	22
2.3 Вибір мови програмування для розробки додатку	23
2.4 Вибір бібліотеки для роботи з Telegram Bot API.....	26
2.5 PostgreSQL	27
2.6 Використання системи контролю версій Git та Github	29
2.7 Unified Modeling Language	31
РОЗДІЛ 3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	33
3.1 Модель предметної галузі	33
3.2 Модель прецедентів	34
3.3 Модель діяльності застосунку	36
3.4 Моделювання Бази даних.....	39
РОЗДІЛ 4 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	42
4.1 Ініціалізація чат-бота за допомогою BotFather	42
4.2 Використання бази даних Postgresql	44
4.3 Тестування додатку.....	47
4.4 Інструкція користувача.....	50

Висновки	53
Перелік посилань	54
Додаток А.....	56

Вступ

Технологічний вибух ХХІ сторіччя який лише нещодавно відбувся, але відіграє важливу роль в житті людства, не є новиною ні для кого, адже приніс у світ нові технології та можливості, а з тим збільшились і потреби щодо комфорту звичайних людей.

Роботи, розумні будинки, смартфони те що в минулому сторіччі люди вважали фантастикою вже сьогодні стає буденністю, але програмні та технічні засоби розвиваються з різними швидкостями, те що повинно зробити життя простішим може ускладнювати в іншому.

Так використовуючи смартфон та відповідні технічні та програмні засоби можна значно підвищити комфорт, але смартфон має велику кількість недоліків які на сьогоднішній день ще не вирішили. Саме тому розробники програмного забезпечення мусять адаптуватися та робити усе можливе зі своєї сторони щоб не зробити лише гірше.

Саме тому кожен заклад, розробляє свій мобільний додаток що дає можливість за декілька хвилин залишити замовлення, та збільшують швидкість їх обробки. Через що у кожного на смартфоні можна знайти значну кількість додатків які більшу частину часу лише займають місце, а іноді ще працюють в фоновому режимі що потребляє досить обмежені ресурси смартфона.

Один з методів вирішення цієї проблеми це інтеграція одного додатку до складу іншого, а саме системи замовлення до месенджеру з використанням вже існуючого АРІ для обробки запитів. Месенджер це невід'ємна складова життя багатьох людей, саме тому це дасть змогу користувачеві не покидаючи вже звичного та комфортного середовища задовольнити свої потреби, а закладам які спеціалізуються на продажі товару отримати досить бюджетний вихід на нову платформу продажу товарів.

Об'єктом дослідження є розробка системи замовлення мовою програмування Python з використанням Telegram Арі. Дана система повинна бути реалізована в

середині месенджера Telegram, надавати змогу користувачам оформляти за відправлення замовлення, та надсилати до менеджера для наступної обробки та виконання замовлення, після чого зберігати данні в БД для подальшого ведення обліку продаж.

Предметом дослідження є можливість реалізації даної системи в месенджері, з використанням передових інформаційних технологій, та аналіз впливу виходу на нову методику збуту товарів.

Мета роботи – розробити систему замовлень в месенджері для подальшого обчислення показників продажу товарів, виявити можливі недоліки та усунути їх.

В процесі дослідження вирішується наступні завдання:

1. Проаналізувати ринок на методи збуту товарів та методи продажу в мережі інтернет;
2. Дослідити можливості використання мови програмування Python для розробки додатку;
3. Встановити вимоги, котрим має відповідати розроблений додаток;
4. Здійснити моделювання та проектування інформаційної моделі;
5. Розробити програмне забезпечення для обробки замовлень в месенджері з використанням мови програмування Python

Методи дослідження – методи структурного аналізу і проектування, аналіз ринку та потреб користувачів та замовників, методи розробки програмного забезпечення, методи тестування, валідації та верифікації програмного забезпечення.

Наукова новизна даної роботи полягає в наступному:

1. Розроблено алгоритм для ефективного обробки замовлень.
2. Проаналізовано ринок месенджерів.

3. Встановлено, що для досягнення поставленої задачі доцільно використовувати мову програмування Python та асинхронну бібліотеку aiogram.

4. Визначено, що мова програмування Python зручна у використанні та надає можливість ефективної розробки додатку на рівень вище інших мов програмування.

Практична значущість - результатів досліджень полягає у вирішенні важливої практичної задачі – розробки програмного забезпечення для обробки замовлень з використанням Telegram API на мові програмування Python, що дозволяє спросити та удосконалити процес створення та обробки замовлення для кожної з сторін, заощадити кошти для замовника та надати користувачам більш комфортну можливість покупки товарів

Результати дослідження бакалаврської роботи два рази апробовані на Всеукраїнській науково-технічній конференції «Застосування програмного забезпечення в інфокомунікаційних технологіях»

Скорочення та умовні позначки

Лінтінг - це процес запуску програми, яка аналізує код на предмет можливих помилок.

Сніпети - фрагмент вихідного тексту або коду програми

IDE - Integrated development environment - Інтегроване середовище розробки

ACID - Atomicity Consistency Isolation Durability

API - Application Programming Interface

pip - Python package manager

РОЗДІЛ 1 ЕЛЕКТРОНА КОМЕРЦІЯ ТА ВИДИ РЕАЛІЗАЦІЇ

1.1 Електрона комерція

Електронна комерція (e-commerce) – це організація більшості бізнесів–процесів підприємства за рахунок їхнього проведення електронним чином. Сюди відносять електронний обмін інформацією (Electronic Data Interchange, EDI), електронний рух капіталу (Electronic Funds Transfer, EDF), електронну торгівлю, обіг електронних грошей, електронний маркетинг, електронний банкінг, електронні страхові послуги тощо. Кожен з видів електронної комерції включає наступні операції:

- встановлення контактів між потенційними контрагентами;
- обмін інформацією;
- купівлю–продаж товарів та послуг;
- електронні платежі;
- управління доставкою;
- бізнес–адміністрування.

Таким чином, електронну комерцію можна характеризувати як ведення бізнесу через Інтернет.

Електронна комерція має багато переваг:

- значно збільшується оперативність одержання інформації;
- значно скорочується цикл виробництва та продажу;
- знижується імовірність виникнення помилок під час введення документів;
- значно знижуються витрати, пов'язані з обміном інформацією за рахунок використання більш дешевих засобів комунікацій;
- використання інтернет–технологій електронної комерції дозволяє легко і швидко інформувати партнерів і клієнтів про продукти і послуги.

1.2 Варіанти реалізації системи замовлень

На сьогоднішній день все нікого не здивуєш використанням електронних систем ведення бізнесу та продажу товарів за останні роки вони показали свою ефективність тому існує декілька реалізацій систем замовлення в мережі інтернет.

1.2.1 Web-site

Web-site це один з найпопулярніших видів реалізації та продажу товарів в мережі інтернет, який набув популярності на початку двадцять першого сторіччя. Це зручний метод реалізації товарів та послуг як для користувачів так і для замовників рис.1.1

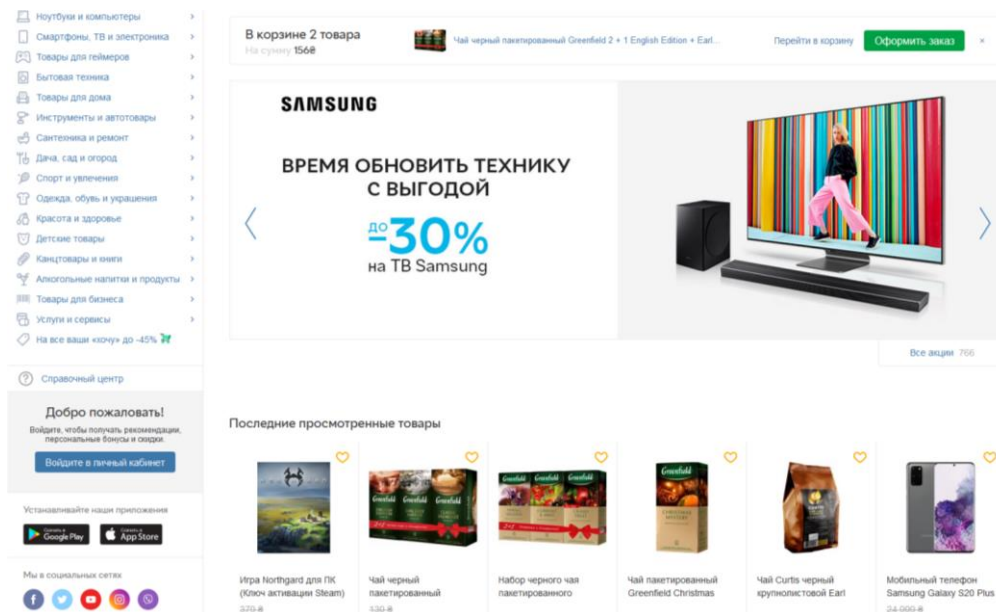


Рисунок 1.1 – Приклад веб-сайту для продажу товарів.

Цей метод майже не має недоліків, але невпинний розвиток технологій та популяризація смартфонів, навіть не зважаючи на швидку адаптацію під велику кількість різноманітних гаджетів не дала змогу утримати першу позицію в реалізації товару, та поступилося мобільним додаткам.

1.2.2 Мобільні додатки

Метод реалізації інтернет торгівлі що виник та набрав популярність з часу популяризації смартфонів та займає першу позицію на сьогоднішній день.

Основна перевага даного методу в тому що смартфон завжди під рукою та користувачам набагато легше робити замовлення використовуючи спеціально розроблений додаток, а не використовувати браузер та робити замовлення на сайті.



Рисунок 1.2 – Приклад мобільного додатку.

Хоча головна перевага також й головний недолік, для того щоб використати даний метод необхідно завантажити відповідний додаток використовуючи маркет на смартфоні. Зазвичай більшість користувачів для замовлення їжі, або іншого невеличкого замовлення ігнорують додатки та використовують звичайний дзвінок на номер телефона закладу. Бо

користувачі які досить не часто замовляють повинні встановити додаток, який займає ресурси телефону, а іноді ще збирає дані о користувачах.

1.2.3 Месенджер

Останній на сьогоднішній день метод ведення інтернет торгівлі де реалізація товарів здійснюється з використанням месенджерів. Виник даний метод з моменту популяризації месенджерів.

Цей метод не має недоліків попередніх, бо для оформлення замовлення можна використовувати інтерфейс розроблений спеціально для смартфонів, не потрібно використовувати браузер та не потрібно завантажувати додатковий додаток рис 1.3 .

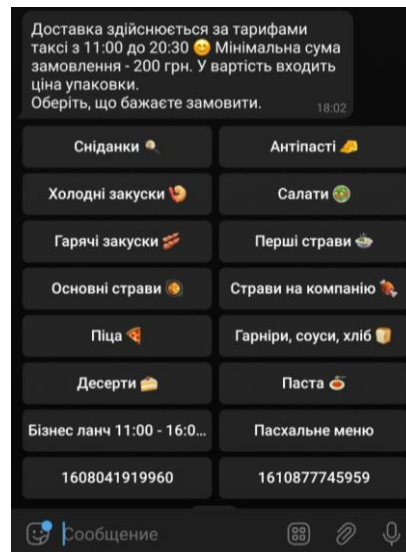


Рисунок 1.3 – Приклад реалізації в месенджері.

Месенджер це те що зараз є в кожного, від маленької дитини до похилого віку людей, хоча основне завдання це надання можливості підтримувати зв'язок між людьми за останні роки їх функціонал стрімко розвивається. Так вже сьогодні багато закладів використовують месенджери як основне місце для маркетингу та реалізації товарів

Хоча даний метод також має і недоліки, де головним є несумісність з великими інтернет-магазинами так як мінімалістичний інтерфейс месенджера в даному випадку буде не зручним для користувачів.

1.3 Месенджер

Месенджер - це програмний додаток для обміну повідомленнями. Миттєвий обмін повідомленнями - це вид онлайн-чату, який пропонує передачу тексту в режимі реального часу через інтернет.

З моменту появи смартфонів і подальшого вибухового зростання мобільних додатків, чати і додатки для обміну повідомленнями в соціальних мережах зарекомендували себе як дешева, а частіше за все безкоштовна альтернатива текстових повідомлень за допомогою операторів через SMS.

Основний функціонал месенджерів:

- групові чати;
- обмін графікою, відео;
- навіть звуковими повідомленнями;
- наліпками й емоціями;
- групові голосові чати;
- безкоштовне збереження даних;

Хоча основний функціонал у всіх месенджерів однаковий кожен має свої особливості що в купі с іншими факторами впливає на популярність. Так світова тенденція на популярність месенджерів відрізняється рис

Месенджер це не лише місце де люди можуть підтримувати контакт з родичами, сьогодні це основний засіб зв'язку для багатьох великих компаній, навчальних закладів та державних установ. Це місце де користувачі можуть дивитися фільми, читати новини та проводять більшу частину вільного часу.

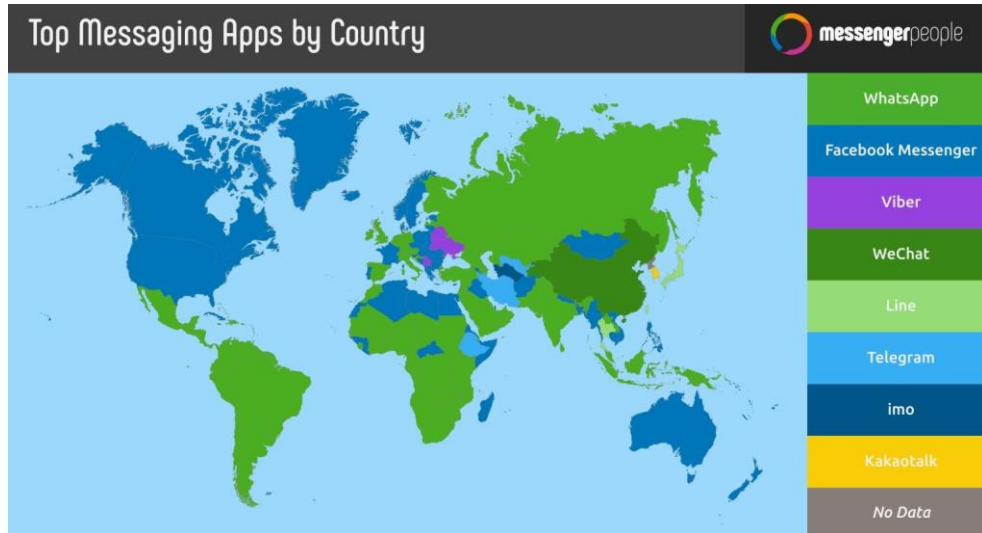


Рисунок 1.4 - Популярність месенджерів у країнах світу.

Саме тому WhatsApp, що найшвидше адаптувався під потреби ринка є лідером за розповсюдженням та щоденної кількості активних користувачів, та має майже на 700 мільйонів більше користувачів ніж найближчий конкурент, що можна побачити на рис1.2

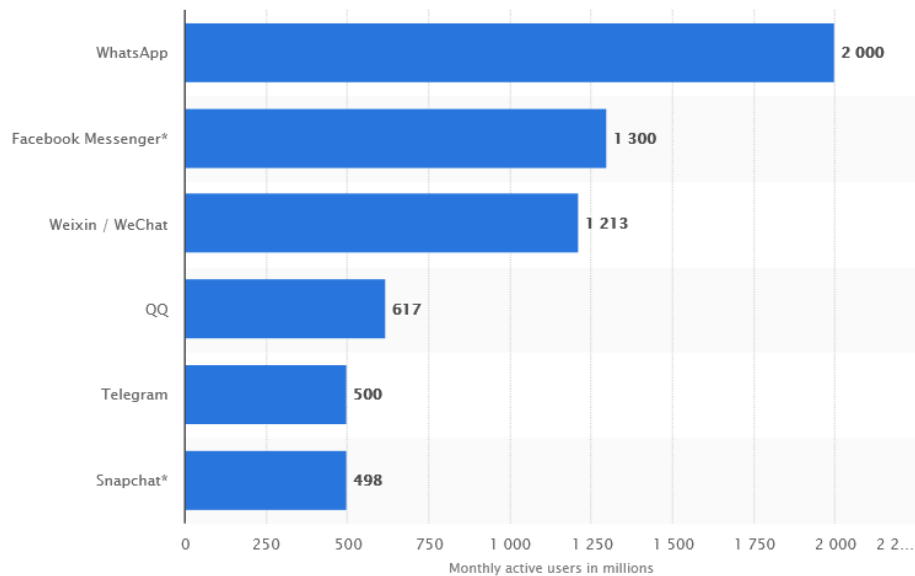


Рисунок 1.5 - Кількість активних користувачів світі

Для повноти роботи потрібно навести короткі відомості для найпопулярніших месенджерів.

WhatsApp - 2 млрд. Користувачів, це додаток для обміну повідомленнями, що використовується людьми в більш ніж 180 країнах. Спочатку WhatsApp використовувався людьми тільки для спілкування з родиною і друзями, але поступово люди почали вести і бізнес через нього. У світі всього 25 країн, де WhatsApp не є лідером ринку.

Facebook Messenger - 1,3 млрд користувачів, раніше був функцією обміну повідомленнями в Facebook, а з 2011 року Facebook перетворив Messenger в окремий додаток і значно розширив його можливості

WeChat - 1,213 млрд. Користувачів.. Крім обміну повідомленнями та дзвінків, користувачі можуть використовувати його для здійснення покупок в Інтернеті і здійснення платежів в автономному режимі, переказу грошей, бронювання, замовлення таксі і багато чого іншого.

QQ - це відома платформа, яка налічує 731 мільйон користувачів. Основні користувачі китайці, але також використовується в 80 країнах світу. Крім функцій миттєвого обміну, також дає можливість користувачам прикрашати свої автарки, дивитися фільми, грати в онлайн-ігри, робити покупки в інтернеті, вести блог і здійснювати платежі.

Telegram - 400 млн. Користувачів. Telegram схожий на більшість додатків для обміну повідомленнями та досить відомий своєю безпекою. Також набув популярності завдяки простоті створення чат-ботів для платформи, а також завдяки функцію каналу Telegram для трансляції повідомлень необмеженій кількості передплатників.

Хоча Telegram не є найпопулярнішим додатком як у світі так і в Україні, але завдяки своїй захищеності та іншим перевагам цільова аудиторія кожного року збільшується, також активна підтримка розробників програмного забезпечення та розвинення програмної бази для розробки чат-ботів в перспективі надає змогу зайняти лідируючу позиці серед месенджерів.

РОЗДІЛ 2 АНАЛІЗ НАЯВНИХ ЗАСОБІВ ТА ТЕХНОЛОГІЙ ОРГАНІЗАЦІ ПРОДАЖУ ТОВАРІВ

2.1 Вибір середовища розробки

Для раціонального вибору середовища розробки, розглянемо три найбільш популярних середовища: PyCharm, Visual Studio Code, Sublime Text

PyCharm - це інтегроване середовище розробки спеціально для Python. Розроблено компанією JetBrains. Має широкий набір можливостей, таких як підсвічування помилок, виправлення автозавершення та інспекції коду, налагодження, система контролю версій і рефакторинг.

IDE доступна на всіх операційних системах. Існує безкоштовна і платна версії. Основний недолік даної середовища розробки це повільна робота на слабких комп'ютерах, особливо коли оперативної пам'яті менше 8 Гб.

Visual Studio Code - безкоштовний редактор коду від Microsoft для Windows, Linux і MacOS. Його можливості - налагодження, підсвічування синтаксису, завершення коду, зумовлені фрагменти коду, рефакторинг.

Підтримуються різні мови програмування. Для початку роботи з Python може знадобитися кілька додаткових пакетів, але встановити їх досить просто. Редактор постійно оновлюється, середовище розробки безкоштовне. Така ж універсальність стосується і команд, що мають можливість працювати над проектом як від однієї людини так і до великої команди, використовуючи, систему контролю версій – Git.

Завдяки багатьом перевагам та можливостям що надає дана IDE отримала значну підтримку та популярність серед розробників з усього світу, про що свідчить опитування проведене StackOverflow в 2019 році де до найближчий конкурент також розробки Microsoft відстає на 20% що можна побачити на рис 2.1

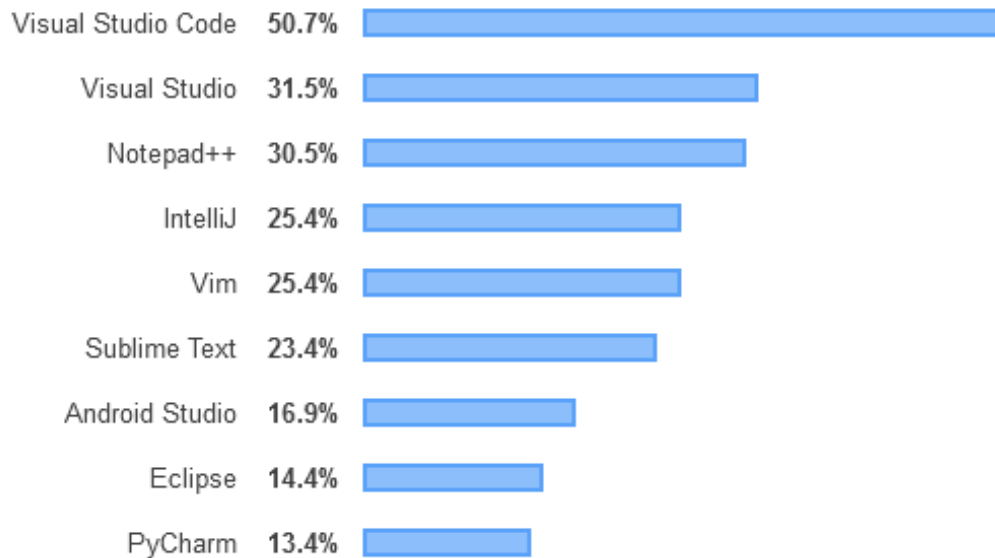


Рисунок 2.1 - Рейтинг популярності IDE серед розробників

Sublime Text - вільне програмне забезпечення з широкою підтримкою спільноти. Редактор працює з декількома мовами програмування. У більшості випадків можна використовувати незареєстрований Sublime Text, але час від часу ви будете отримувати спливаюче вікно з проханням зареєструвати продукт і придбати ліцензію. Sublime Text тонко настроюється і доповнюється пакетами для налагодження, автозавершення коду, лінтинга і т.д.

Основна перевага Sublime Text це швидкість, простота та надійність, досить важкі проекти він здатний відкрити за декілька секунд, але по функціоналу поступається аналогам зазначеним вище.

Таким чином розглянув найпопулярніші середовища розробки, для розробки додатку на мові програмування Python, було обрано Visual Studio Code. Воно відповідає усім необхідним умовам, надає необхідні можливості, зручне в налаштуванні та здатне збільшити ефективність розробки. Хоча в розробці також приймає участь Sublime Text саме завдяки швидкості роботи це ідеальне середовище коли потрібно швидко виправити помилки та не потрібно надавати будь-яку увагу на особливості та можливості.

2.2 Visual Studio Code

VS Code від Microsoft - легкий і зручний редактор коду, доступний на всіх платформах і неймовірно гнучкий. Продукт щомісяця оновлюється і покращується. У нього з одразу після встановлення вбудована підтримка декількох мов програмування та зручна модель розширення. Інтерфейс досить простий і інтуїтивно зрозумілий .

Основні можливості редактора:

- підтримка Python 3.4 і вище, а також Python 2.7;
- автоматичне доповнення коду за допомогою IntelliSense;
- лінійні;
- налагодження;
- сніппети;
- модульне тестування;
- автоматичне використання conda і віртуальних середовищ;
- редагування коду в середовищах Jupyter і Jupyter Notebooks.

У редакторі є і корисні особливості, не пов'язані безпосередньо з мовою:

- набори гарячих клавіш для інших IDE;
- налагоджуються теми оформлення;
- мовні пакети для багатьох мов;

І ще кілька крутих можливостей для повного щастя:

- GitLens - безліч корисних функцій Git прямо в редакторі;
- автозбереження (File - Auto Save) і зручна настройка його використання.
- синхронізація параметрів редактора між різними пристроями за допомогою GitHub;
- зручна робота з Docker.

Щоб знайти і встановити необхідні розширення і теми, достатньо зробити декілька кліків також існує пошук за ключовими словами і можливість сортувати результати пошуку.

У Visual Studio Code можна налаштувати все під себе рис. 2.2. Так як є параметри користувача, які є глобальними, і параметри робочої області - локальні для конкретних папок або проектів. Локальні настройки зберігаються в вигляді .json-файлів в папці .vscode.

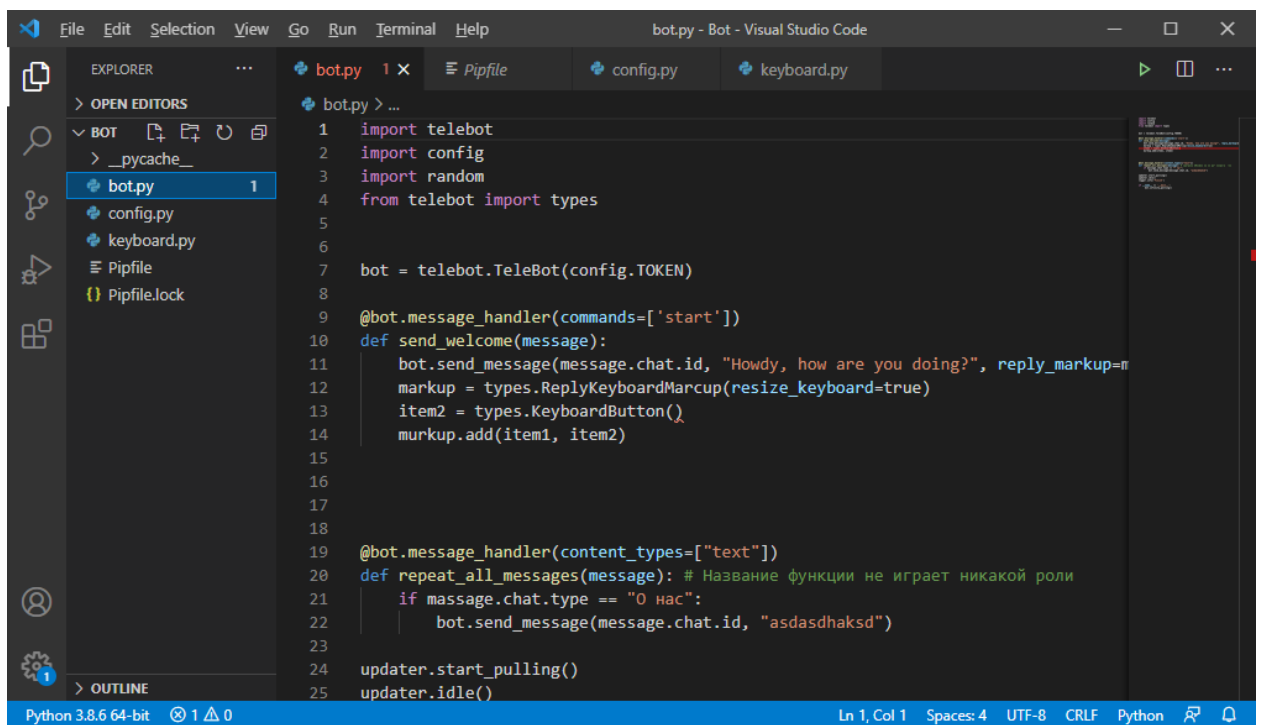


Рисунок 2.2 – Інтерфейс Visual Studio Code

2.3 Вибір мови програмування для розробки додатку

Вибір мови програмування здійснювався між Python та JavaScript. Обидві мови чудово підходять для виконання поставленої задачі та мають вже написані бібліотеки що взаємодіють з Telegram API та значно спрощують розробку, бо не потрібно фокусуватися на методах обробки запитів та повідомлень з серверами Telegram і можна сфокусуватися на розробці додатку.

Python – динамічна строго типізована мова програмування, що спеціалізується на комфорті написання, читання та якості коду, орієнтований на підвищення продуктивності розробника програмного забезпечення. Завдяки мінімалістичному синтаксису рис.2.3 та варіативності використання займає топові положення в рейтингу мов програмування, запит на розробників збільшується кожного року.



Рисунок 2.3 – Порівняння синтаксису Python з іншими мовами програмування

Також потрібно зазначити що Python це open sours тому кожен бажаючий має змогу відкрити код інтерпретатора та модифікувати його тим самим прийняти участь удосконалені проекту.

Головним недоліком Python вважається швидкість роботи, це пов'язано з принципами роботи ядра мови, хоча існують декілька способів вирішення цієї проблеми, де найпопулярніше реалізація функціоналу що потребує швидкої обробки іншою мовою програмування, звідки виникли такі різновиди мови:

- CPython компілює в байт-код та виконує в віртуальній машині за основу взят C;
- Jython реалізація Python коду в віртуальній машині Java;
- IronPython реалізація з використанням компілятора та методів C#;

- PyPy альтернатива CPython з використанням JIT-компіляції.

JavaScript – динамічна, слабо типізована об'єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб сторінок, що надає можливість на боці клієнта взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб сторінки.

При порівнянні зазначених мов програмування важко визначити однозначного фаворита, обидві мови по праву займають топові позначки в рейтингу популярності серед розробників, та запиту серед компаній. Хоча останнім часом кількість запитів що стосувалися Python серед розробників на спеціалізованих ресурсах в перше за наздогнав та перегнав JavaScript що можна побачити на рисунку 2.4 .



Рисунок 2.4 графік порівняння кількості запитів за останні роки

В цьому проєкті перевага була надана саме Python через активну та зручну в використанні бібліотекам, синтаксису та завдяки навичкам що були отриману під час навчання в університеті.

2.4 Вибір бібліотеки для роботи з Telegram Bot API

Python дуже популярна мова що має активне ком'юніті що постійно розвивається, налічує велетенську кількість фреймворків що також активно розвиваються. Для розробки чат-ботів та взаємодії з Telegram Bot API є дві досить відомі бібліотеки:

- Aiogram;
- PyTelegramBotApi;
- Telebot.

Основне завдання даних фреймворків це обробка повідомлень що користувачі надсилають через чат, та контроль логіки бота що контролюється за допомогою HTTPS запитів до Bot API.

Хоча існує набагато більше фреймворків для розробки під Telegram виділити можна лише перші два, так як найбільш розповсюджені та мають активне співтовариство та активно удосконалюється.

PyTelegramBotApi чудово підходить для тих хто тільки починає знайомитися з розробкою ботів та хоче зрозуміти основи правці з Bot API, має чудову документацію та багато прикладів що можна використовувати для навчання.

Основним недоліком даного фреймворку є застарілість та синхронне виконання запитів користувачів через що не підходить для активних та важких проєктів.

Aiogram повинена бібліотека яка чудово підходить для розробки високопродуктивних ботів, постійно обновляється та має багато ексклюзивних особливостей як приклад FSM, встановлення стандартного типу

розмітки, асинхронність та багато іншого, корисного при розробці. Хоча дана бібліотека вийшла набагато пізніше за першу, вона активно розвивається та все більше людей поступово переводять свої проекти саме на неї.

2.5 PostgreSQL

PostgreSQL - це об'єктно-реляційна система управління базами даних ОРСУБД, ORDBMS, заснована на POSTGRES Version 4.2 - програмою, розробленою на факультеті комп'ютерних наук Каліфорнійського університету в Берклі. У POSTGRES з'явилося безліч нововведень, які були реалізовані в деяких комерційних СУБД набагато пізніше. Це СУБД з відкритим вихідним кодом, основою якого був код, написаний в Берклі.

СУБД відрізняється високою надійністю і хорошою продуктивністю, підтримує ACID, реплікація реалізована вбудованими механізмами. При цьому система розширюється - можна створювати свої типи даних і індексів, а також розширювати поведінку за допомогою мов програмування.

База даних входить до топу найпопулярніших СУБД та кожного року зміцнює свої позиції рис. 2.5

З додаткових можливостей PostgreSQL хочеться окремо відзначити той факт, що ця СУБД дозволяє працювати не тільки зі структурованими і нормалізованими, а й зі слабо-структурованими даними, при цьому ці дані індексуються і робота з ними дійсно зручна. Особливу цінність ці можливості знаходять при розробці систем, де моделі даних повинні мати можливість редагуватися в режимі реального часу на рівні GUI

Вона підтримує більшу частину стандарту SQL і пропонує безліч сучасних функцій:

- складні запити
- зовнішні ключі
- тригери

- змінювані уявлення
- цілісність транзакцій

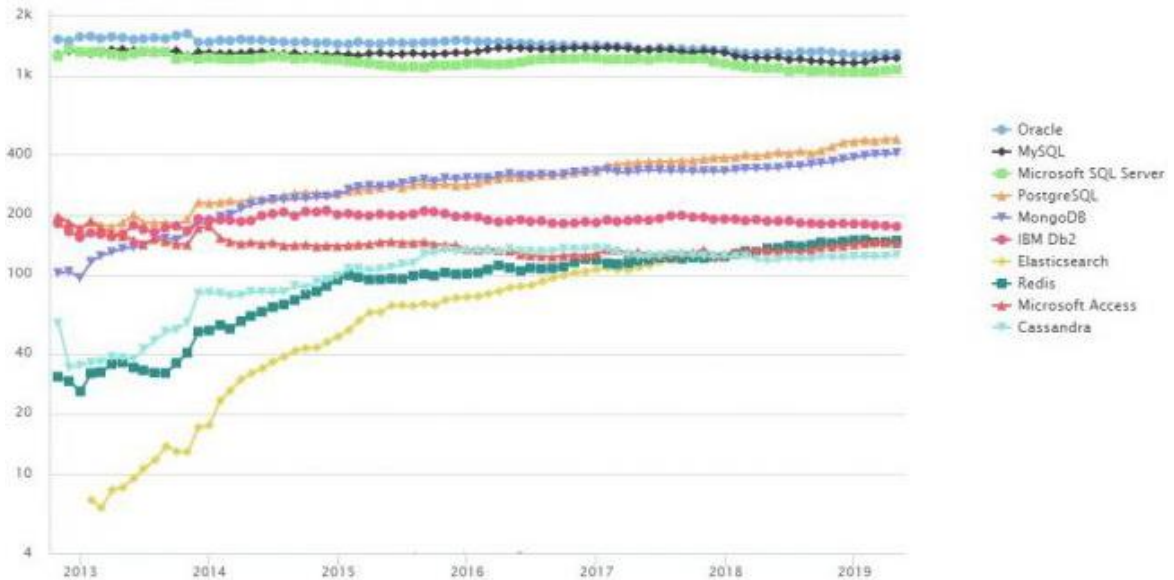


Рисунок 2.5 – Регресія популярності СУБД

Крім того, користувачі можуть всіляко розширювати можливості PostgreSQL, наприклад створюючи свої:

- типи даних
- функції
- оператори
- агрегатні функції
- методи індексування
- процедурні мови

Також ця СУБД має вільну ліцензію, що дозволяє безкоштовно використовувати, редагувати та розповсюджувати для будь-яких цілей, як комерційних так і для навчання, власних проєктів.

Усі наведені переваги зіграли основну роль в виборі бд для даного проєкту саме тому було прийнято рішення використовувати PostgreSQL

2.6 Використання системи контролю версій Git та Github

Система контролю версій (СКВ) дозволяє розробникам зберігати всі зміни, котрі були внесені в код програми. Збереження змін, дає можливість відновити код більш старих версій, в разі виявлення не коректної поведінки програми, для заощадження часу та інших ресурсів.

СКВ дає змогу працювати над одним проектом одразу декільком розробникам без шкоди один одному. Існують три типи СКВ: локальна, централізована та розподілена. Далі розглянемо детально розподілену систему контролю версій.

Розподілена система контролю версій – система де користувачі повинні повністю копіювати репозиторій. Тобто, у кожного розробника є копія усього початкового коду проекту та усіх змін, котрі були внесені під час розробки проекту.

Це означає, якщо один із серверів тимчасово вийде з ладу, будь-який інший репозиторій може бути скопійований на сервер для відновлення роботи в штатному режимі, без втрати даних. Схема розподільної системи контролю версій зображено на рис 2.6 .

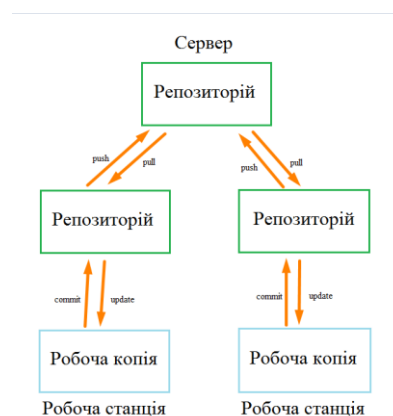


Рисунок 2.6 - Схема розподільної системи контролю версій

Git – це розподілена система контролю версій, що надає можливість декільком розробникам одночасно відстежувати зміни в файлах, вносити зміни самостійно та сумісно працювати над одним проектом із можливістю відстеження змін. Найбільш відомі особливості Git – це простий дизайн, швидкодія, підтримка нелінійної розробки, повна децентралізація та, за необхідності, можливість роботи з великими проектами.

GitHub – сервіс хостинга репозиторіїв, який вмщує в собі всі функції, що підтримує Git, а саме функції розподільного контролю версій і функціональність управління початковим кодом. Використання GitHub разом з Git дає розробникам можливість збереження коду в хмарі та взаємодіяти з іншими розробниками, що в свою чергу активно використовуються в open source project . Важлива та зручна функція це історія комітів, завдяки чому є можливість просто відстежувати які саме зміни були внесені. рис.2.7

В історії комітів виділено файли та частини коду, в які було внесені зміни, існує можливість порівняти частини коду, що спрощує відладку коду та пошук помилок.

Використання Git необхідне для проекту для мінімізації можливих ризиків з втратою проекту через технічні фактори.

```

21 21
22 22     def runshell(self, parameters):
23 23         args, env = self.settings_to_cmd_args_env(self.connection.settings_dict, parameters)
24 -         if env:
25 -             env = {**os.environ, **env}
24 +         env = {**os.environ, **env} if env else None
26 25         subprocess.run(args, env=env, check=True)

```

```

@@ -51,7 +51,7 @@ def settings_to_cmd_args_env(cls, settings_dict, parameters):
51 51         env['PGSSLKEY'] = str(sslkey)
52 52         if passfile:
53 53             env['PGPASSFILE'] = str(passfile)
54 -         return args, env
54 +         return args, (env or None)
55 55
56 56     def runshell(self, parameters):
57 57         sigint_handler = signal.getsignal(signal.SIGINT)

```

```

@@ -65,3 +65,6 @@ Bugfixes
65 65     admin changelist now uses ``Exists()`` instead ``QuerySet.distinct()``
66 66     because calling ``delete()`` after ``distinct()`` is not allowed in Django

```

Рисунок 2.7 - Вигляд коміту на GitHub

2.7 Unified Modeling Language

Уніфікована мова моделювання (Unified Modeling Language, UML) - це графічна мова для візуалізації, специфікація, конструювання та документування систем, в яких головна роль належить програмному забезпеченню. За допомогою UML можна розробити детальний план створюваної системи, що містить не тільки її концептуальні елементи, такі як системні функції і бізнес процеси, а й конкретні особливості, наприклад класи, написані на будь-якій мові програмування, схеми баз даних і повторно використовувані програмні компоненти .

Для швидкої і ефективної розробки програмного додатку з мінімальною кількістю помилок потрібно залучити робочу силу, підібрати правильні інструменти та визначити напрям роботи. Щоб впоратися з поставленим завданням, беручи до уваги витрати на життєвий цикл системи, необхідно, щоб процес розробки програми був ретельно продуманий і міг бути адаптований до потреб бізнесу та технологій.

Центральним елементом діяльності, що веде до створення першокласного програмного забезпечення, є моделювання. Моделі дозволяють наочно продемонструвати бажану структуру і поведінку системи. Вони також необхідні для візуалізації і управління її архітектурою. Моделі допомагають добитися кращого розуміння майбутньої системи, що дозволяє її спростити та забезпечити можливість повторного використання. А також моделювання потрібно для мінімізації ризику.

Моделювання дозволяє вирішити чотири різні завдання:

- Візуалізувати систему в її поточному або бажаному стані;
- Описати структуру або поведінку системи;
- Отримати шаблон, що дозволяє сконструювати систему;
- Документувати прийняті рішення, використовуючи отримані моделі;

Отже, уніфікована мова моделювання стандартний інструмент для розробки креслень програмного забезпечення. Його можна використовувати для візуалізації, специфікації, конструювання та документування артефактів програмних систем

Діаграма - це графічне представлення набору елементів, найчастіше зображеного у вигляді зв'язного графа вершин і шляхів Ви малюєте діаграми для візуалізації системи з різних точок зору, тому окрема діаграма - це проекція системи.

UML включає 13 видів діаграм:

- діаграма класів;
- діаграма об'єктів;
- діаграма компонентів;
- діаграма складовою структури;
- діаграма варіантів використання;
- діаграма послідовності;
- діаграма комунікацій;
- діаграма станів;
- діаграма діяльності;
- діаграма розміщення;
- діаграма пакетів;
- тимчасова діаграма;
- Діаграма огляду взаємодії.

Кожна з перерахованих діаграм є компонентом моделювання системи, де кожна наступна діаграма доповнює попередню та надає можливість комплексно змодельовати систему. Моделювання діаграм можна порівняти з роботою інженера будівельника, який перед початком будівництва будинку аналізує всі можливі фактори та моделює майбутній будинок.

РОЗДІЛ 3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Модель предметної галузі

Розробка інформаційної системи це розробка засобів для управління інформацією. Де ІС приймають інформацію, по строгим правилам, обробляють її та видають результат користувачам.

Для того щоб розробити інформаційну систему потрібно не лише розуміти бізнес-процеси та потреби замовника. Потрібно розуміти які об'єкти потрапляють до предметної області проєктованої інформаційної системи та які логічні зв'язки між ними існують. Для формування такого розуміння використовують логічні моделі предметної області.

Головна ціль моделювання предметної галузі – змоделювати суть та взаємовідносини між ними. Де логічні моделі предметної області це можливість графічного представлення логічної структури досліджуваної предметної області.

Суті описують об'єкти предметної області, а також суб'єкти, що здійснюють діяльність в рамках предметної області. Властивості об'єктів і суб'єктів реального світу описуються за допомогою атрибутів.

Відносини між сутностями ілюструються за допомогою зав'язків. Правила та обмеження взаємин описуються за допомогою властивостей зав'язків. Зазвичай зв'язки визначають якої залежності між сутностями, вплив однієї сутності на іншу.

На рисунку 3.1 змодельована модель предметної галузі для даного проєкту, моделювання суті та зав'язків між ними стало основою проєкту та активно використовувалось під час подальшого моделювання.

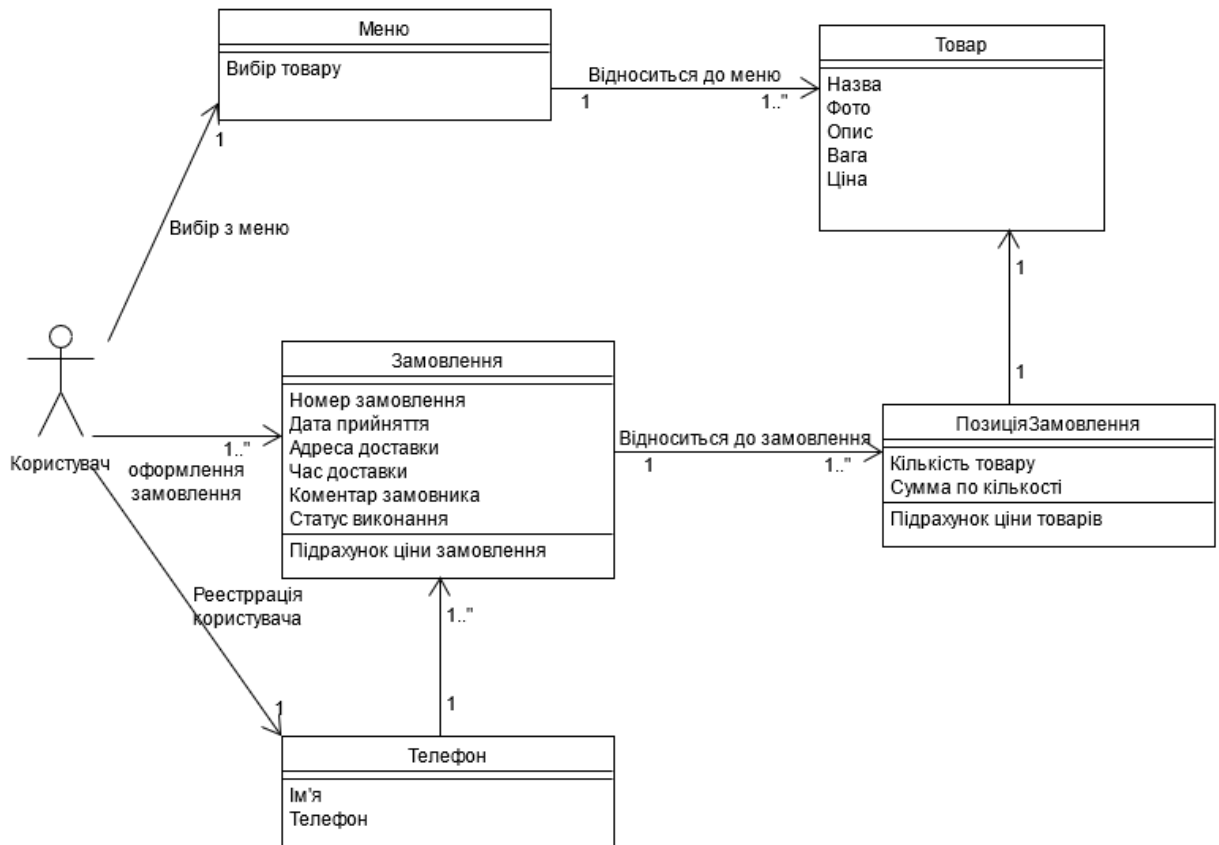


Рисунок 3.1 - Модель предметної галузі

3.2 Модель прецедентів

Діаграми варіантів використання або альтернативна назва діаграма прецедентів це один з видів діаграм UML, призначених для моделювання динамічних аспектів систем, а також це основний вид діаграм при моделюванні поведінки системи, підсистеми або класу. Кожна з них показує набір варіантів використання і дійових осіб в їх взаємодії.

Діаграми варіантів використання застосовуються для моделювання уявлення системи з точки зору варіантів використання. Здебільшого це має на увазі моделювання Контекст системи, підсистеми або класу або моделювання вимог до цих елементів.

Діаграма варіантів використання важливі для візуалізації, специфікації та документування поведінки елементів. Вони забезпечують доступність і

зрозумілість систем, підсистем і класів за рахунок зовнішнього уявлення того, як ці елементи можуть бути використані в контексті. Крім того, такі діаграми важливі для тестування працюючих систем за допомогою прямого проектування і для забезпечення їх розуміння за допомогою проектування.

Для повноти моделювання системи потрібно змодельювати дві можливих діаграми прецедентів, а саме для користувачів та менеджерів, так як головний функціонал розрізняється, але всі працюють в межах однієї системи для досягнення однієї мети.

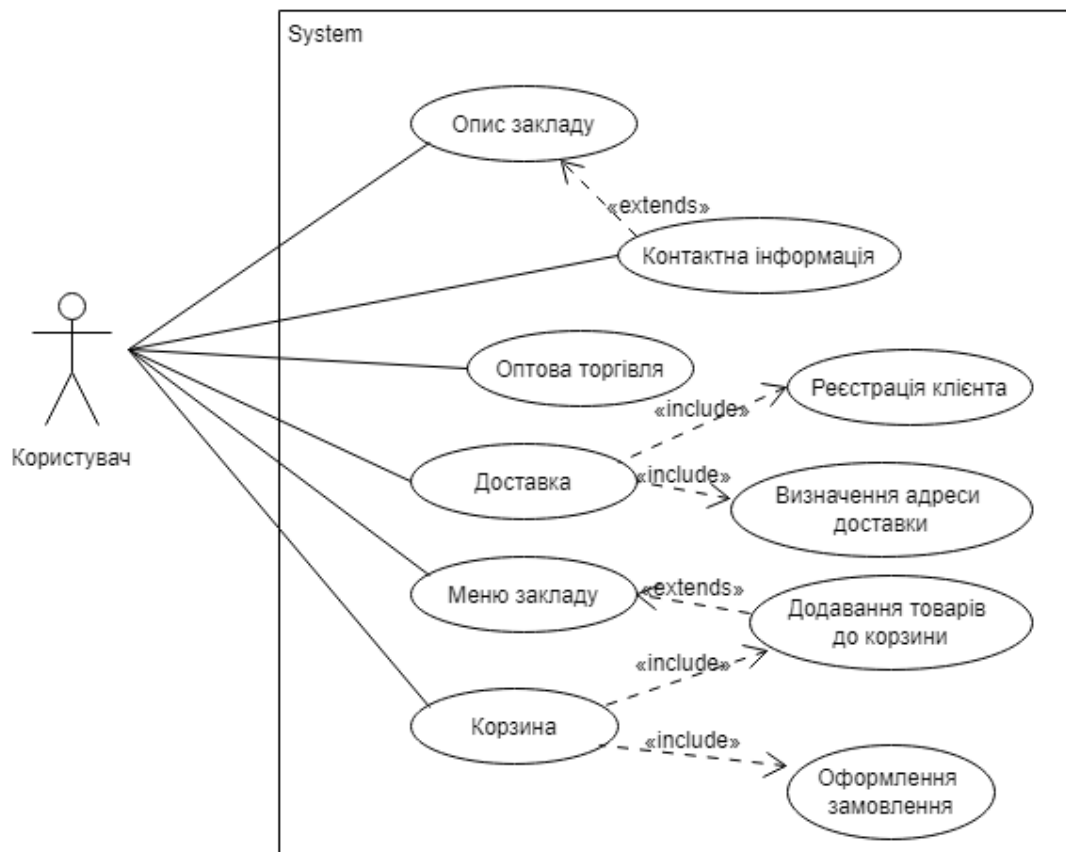


Рисунок 3.2 - Модель прецедентів користувачів

В моделі прецедентів де автором виступає Користувач можна побачити основний функціонал що доступний користувачам та зв'язки які існують між елементами системи, що можна побачити на рис. 3.2

На моделі прецедентів менеджера рис. 3.3 можна побачити головний функціонал який доступний для використання, звязки між елементами системи та головним функціоналом.

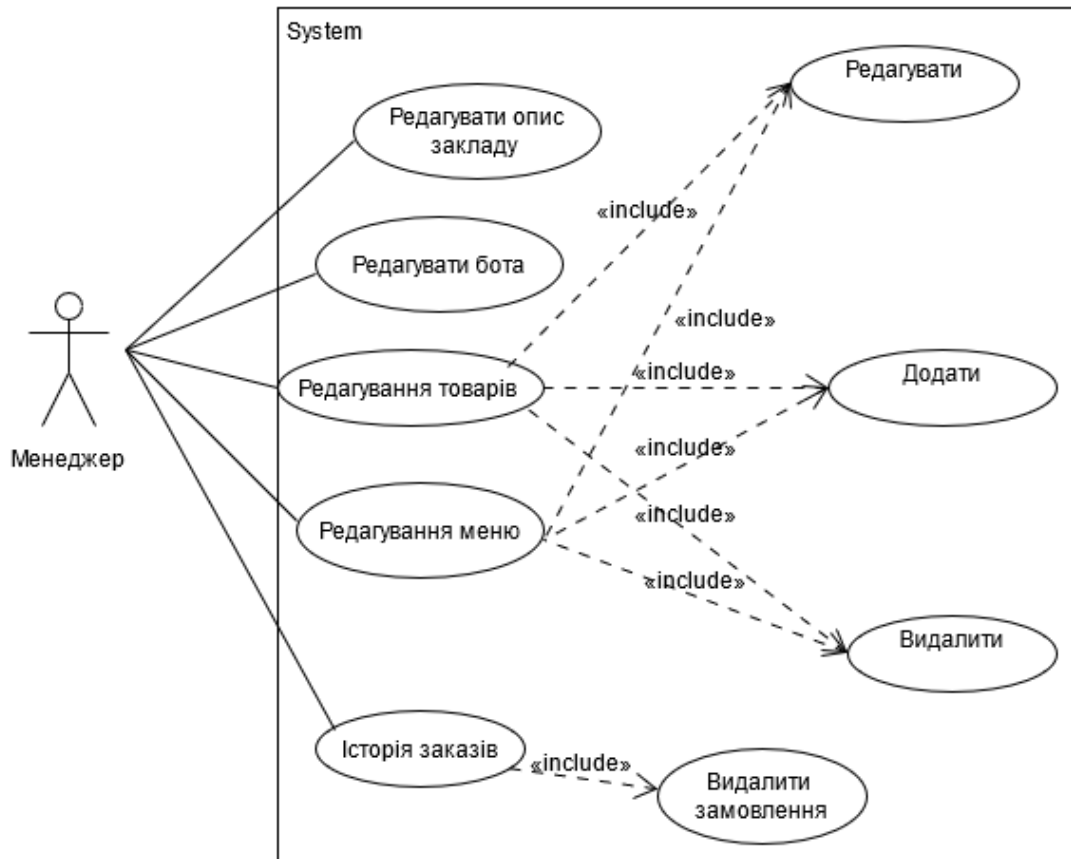


Рисунок 3.3 - Модель прецедентів для Менеджера

Було змодельовано дві моделі прецедентів для поверхневого опису основного функціоналу користувачів, що надає змогу спросити наступне моделювання системи, яку потрібно розробити та звести до мінімуму можливі помилки.

3.3 Модель діяльності застосунку

Діаграми діяльності - це один з п'яти видів діаграм, які використовують в UML для моделювання динамічних аспектів систем.

Діаграма діяльності являє собою блок-схему, яка показує як потік управління переходить від однієї діяльності до іншої. На відміну від традиційної блок-схеми діаграма діяльності показує паралелізм так само добре, як і розгалуження потоку управління

Діаграми діяльності можуть використовуватися окремо для візуалізації, специфікації, конструювання і документування динаміки зав'язків об'єктів або для моделювання потоку управління в операції. Якщо в діаграмах взаємодії акцент робиться на переходи потоку управління від одного об'єкта до іншого, то діаграми діяльності описують переходи потоку управління від одного кроку процесу до іншого.

Діаграми діяльності застосовуються для моделювання динамічних аспектів поведінки системи. Ці динамічні аспекти можуть включати діяльність на будь-якому рівні абстракції в будь-якому поданні системної архітектури, включаючи класи (в тому числі активні), інтерфейси, компоненти і вузли.

Діаграми діяльності важливі не тільки для моделювання динамічних аспектів системи, але і для конструювання виконуваних систем за допомогою прямого і зворотного проектування.

Далі було змодельована діаграма діяльності для поетапного проектування можливостей для користувачів, що можна побачити на рис.3.4 де основний функціонал розробленого додатку починається з *Меню закладу* де користувач дивиться на існуючі в продажу товари та додає до кошика, після чого користувач має можливість ще додати товар, або перейти до *Корзини* та продовжити оформлення замовлення. Після чого користувач вводить контактні дані, що перевіряє системою на коректність введення, обрати спосіб доставки, само вивіз або доставку та в другому випадку користувачеві буде представлена можливість ввести адресу для доставки замовлення. Користувач

отримує повідомлення, що замовлення оформлене, після чого настає логічний вихід.

Потрібно додати, що в зв'язку з специфікою месенджера вихід з додатку можливий після кожного етапу, але для того щоб не навантажувати модель було прийнято рішення вказати вихід лише після виконання основного функціоналу, тому в даному випадку вихід показує отримання бажаного результату для користувача.

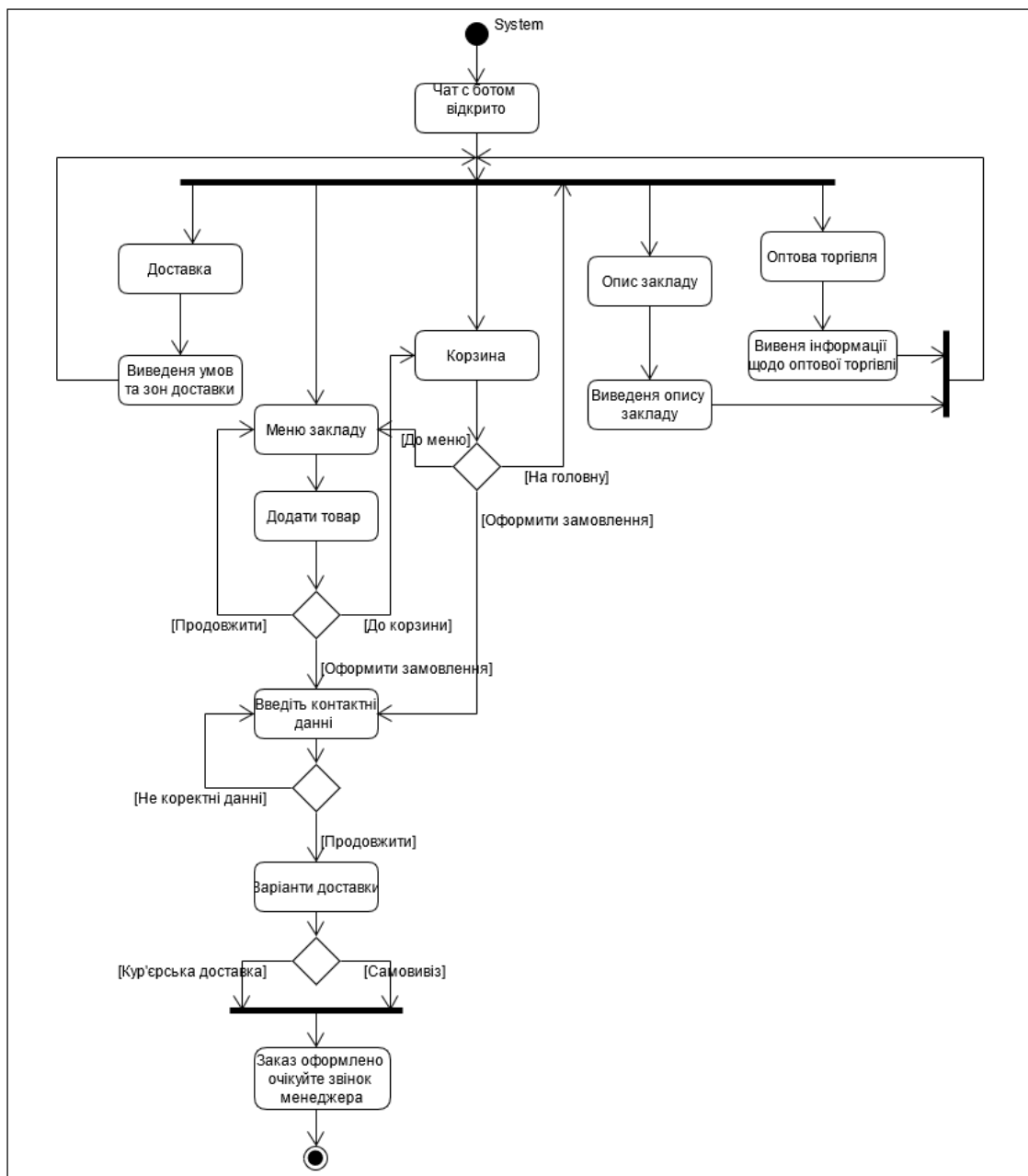


Рисунок 3.4 - Діаграма діяльності користувачів

Для повноти моделювання та розуміння проєкту також було змодельовано діаграму діяльності менеджерів рис. 3.5, саме вони займаються адмініструванням системи, мають усі необхідні для цього елементи, а також отримують оформлені замовлення для подальшої обробки та виконання.

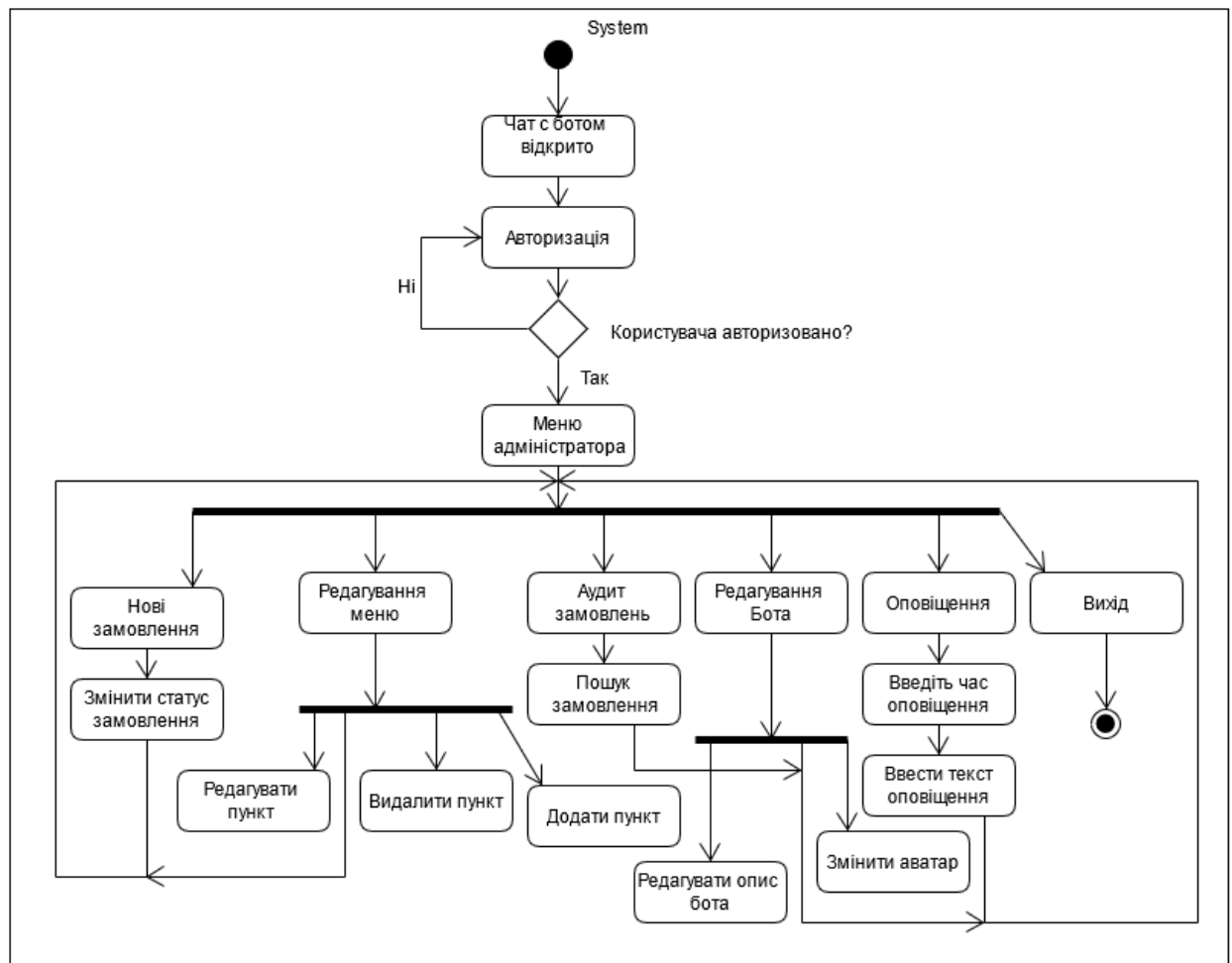


Рисунок 3.5 – Діаграма діяльності менеджера

3.4 Моделювання Бази даних

Методологія проектування баз даних - це сукупність принципів, методів, інструментів та засобів, що застосовуються для послідовного розроблення структури бази даних. Оскільки система баз даних складається з

програм и даних, методологія проектування баз даних розглядається як невід'ємна частина загальної методології проектування програмних систем.

До методології проектування баз Даних висуваються певні вимоги. Прийнятною вважається база даних, яка відповідає вимоги користувачів ефективність, адаптивність, незалежність, захищеність, цілісність і вимоги до апаратного забезпечення.

Моделювання даних це процес розробки моделі для подальшої розробки та зберігання даних в базі даних. Модель даних являє собою концептуальне уявлення об'єктів даних, зв'язків між різними об'єктами даних і правилами.

Моделювання даних допомагає візуально представляти дані і забезпечує дотримання правил, нормативних вимог та державних політик щодо даних. Моделі даних забезпечують узгодженість в угодах про імена, значеннях за замовчуванням, семантиці, безпеки при забезпеченні якості даних.

Розробка моделі даних перед безпосередньою розробкою бази даних це не простий, але дуже важливий етап в розробці, що дозволяє:

- мінімізувати помилки;
- завчасно знайти помилки та виправити;
- оптимізувати та удосконалити;
- отримати наглядний приклад та спростити подальшу розробку;

Етап проектування БД вважається одним із самих складних етапів створення БД, який не має явно вираженого початку й закінчення. У порівнянні з аналізом вимог до БД або розробкою додатків. Якщо всі етапи створення БД перекриваються один з одним у своїй послідовності, то етап проектування перекривається з усіма іншими етапами. Проектування починається з моменту прийняття стратегічних рішень і триває на етапах реалізації й тестування.

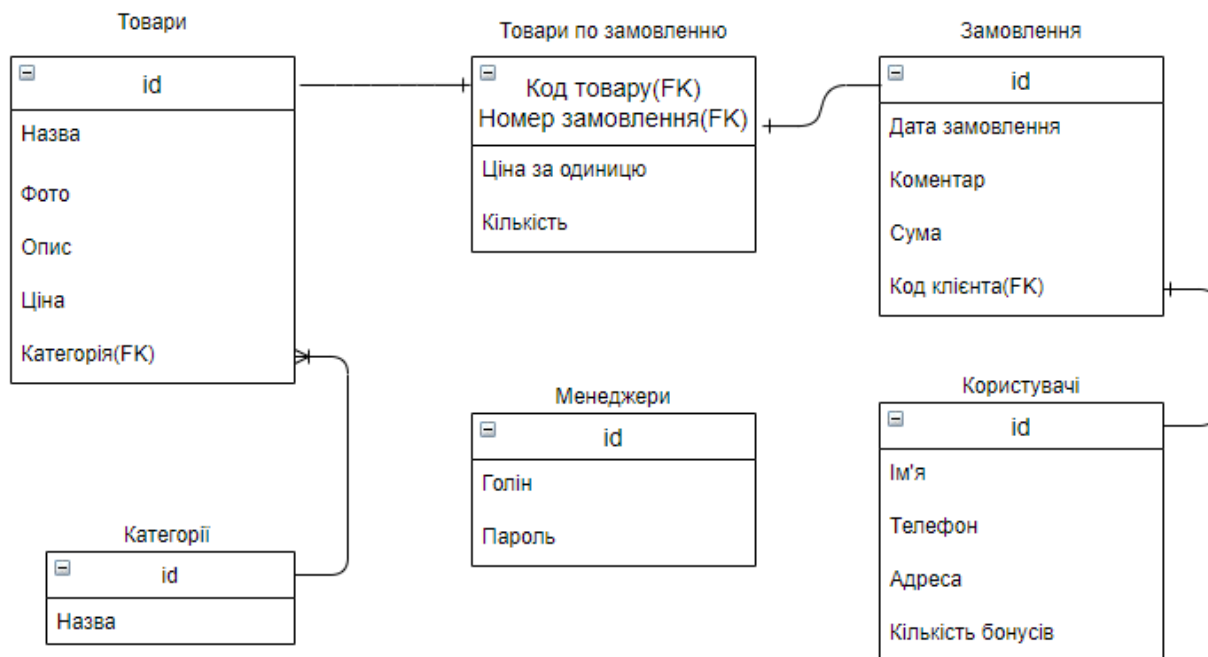


Рисунок 3.6 - Модель бази даних

Було змодельовано базу даних для подальшої розробки рис. 3.6 , де зображено зв'язки та ключі до кожної таблиці.

Данна модель бази даних повністю задовольняє потреби розроблюваного додатка та надає можливість розробити швидку та надійну базу для зберігання та викоцристання необхідних для коректної роботи додатку даних.

РОЗДІЛ 4 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Ініціалізація чат-бота за допомогою BotFather

BothFather – це спеціалізований сервіс реалізований за допомогою чат-бота для реєстрації, налаштування та управління, чат-бота користувачів.

Взаємодія з BothFather відбувається шляхом вводу в текстове поле, спеціально зарезервованих команд, зображених на рис. 4.1 Команди мають свою назву та короткий опис до них. Відкрити вікно для перегляду команд, можна ввівши в текстове символ «/».

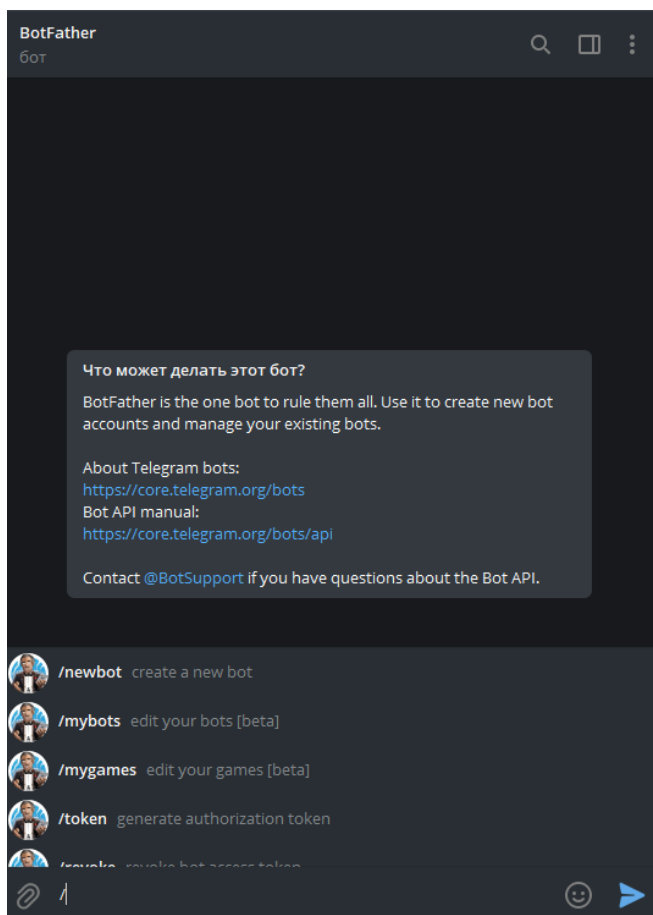


Рисунок 4.1 – Інтерфейс BotFather з частиною зарезервованих команд

За допомогою команди */newbot*, можна створити нового чат-бота, першим кроком створення, є написання унікального ім'я яке буде

відображатися користувачам при роботі з ботом та не було використано іншими користувачами, а також створення додаткового ім'я, яке буде має обов'язково містити слово bot наприкінці додаткового ім'я рис 4.2.

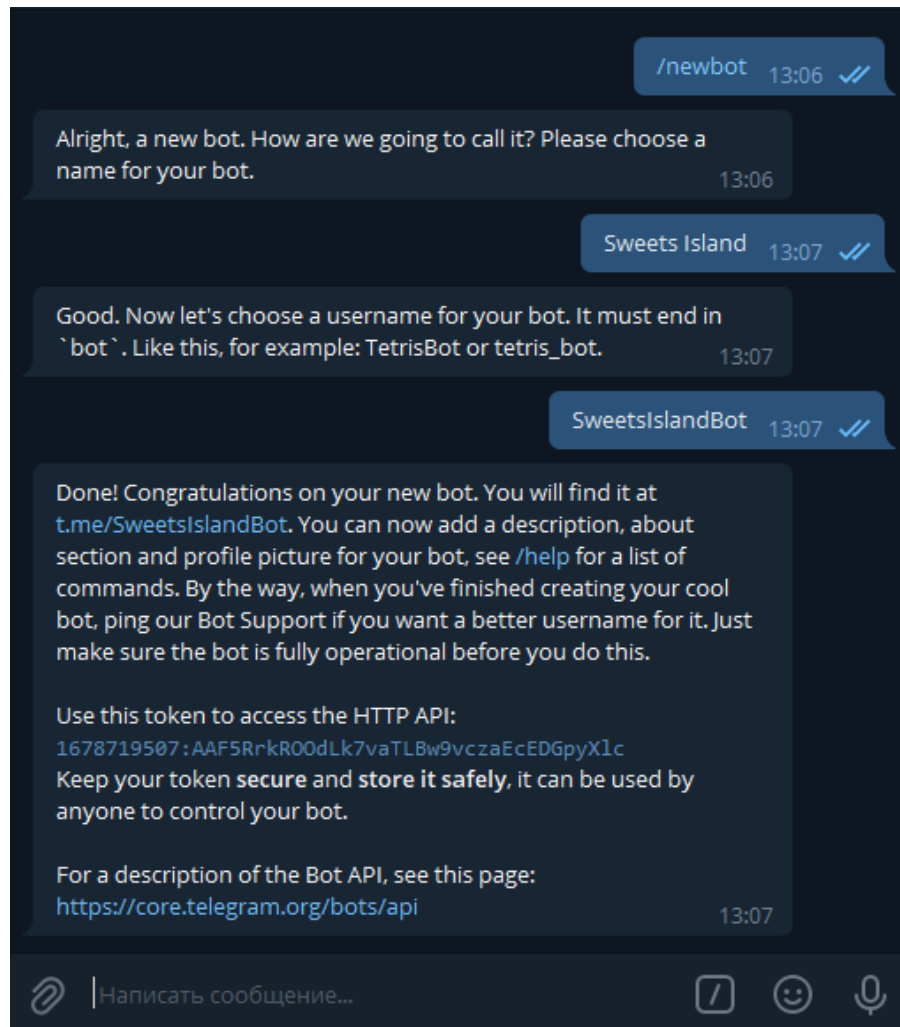


Рисунок 4.2 – Процес створення тестового чат-боту

Після успішного створення чат-бота, йому призначається унікальний токен, що використовується для зв'язку написаного коду та Telegram Bot API та який використовується для отримання доступу до функцій бота і подальшого його програмування з додаванням нового функціоналу та редагуванням стандартних налаштувань бота. Також створюється сам інтерфейс чат-бота для взаємодії з його функціоналом та отримання більш розгорнутої інформації про його можливості.

Переглянути всіх створених вами ботів, можна за допомогою команди */mybots* введеної в BotFather, після цього відобразиться список доступних вам ботів.

Обравши один з них, ви можете переглянути токен вашого боту, подивитися налаштування створеного боту, редагувати його опис, додавати нові команди, змінювати зображення бота, ім'я та видалити обраного чат-бота.

Рис 4.3

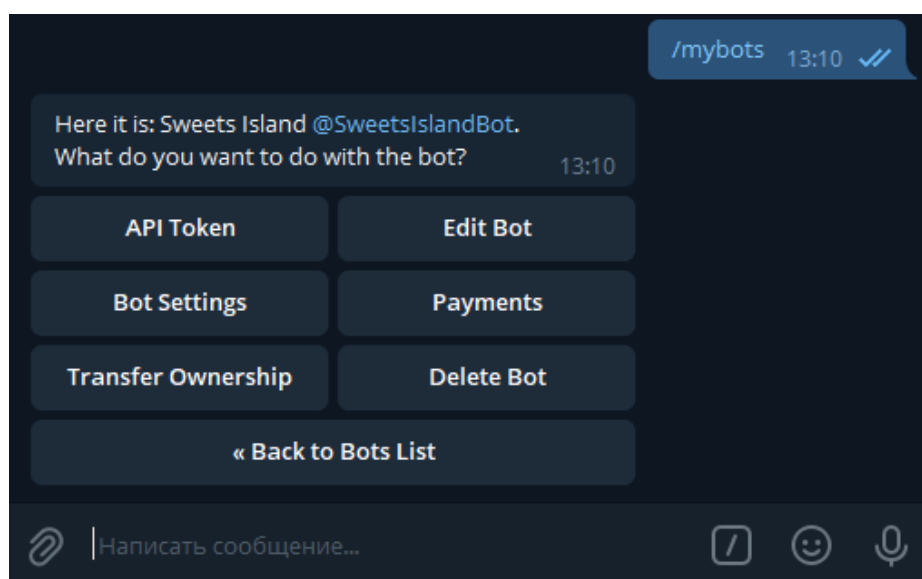


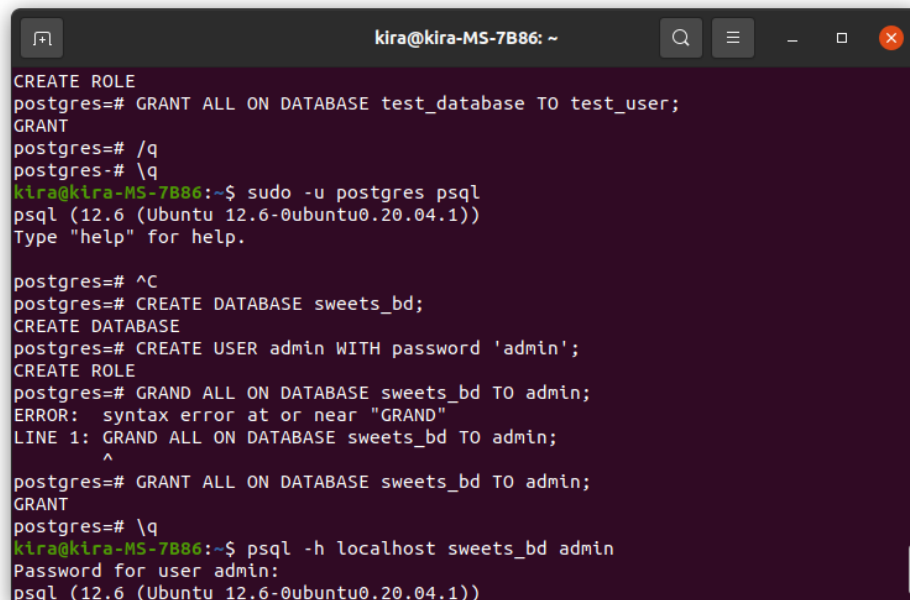
Рисунок 4.3 – Інтерфейс додаткових функцій для управління ботом

BotFather є найкращим рішенням для створення чат-ботів та редагування його функціоналу. Використавши цей сервіс, ми за декілька хвилин створили основу чат-бота для подальшого програмування та розробки яка має зручний функціонал, однаково відображається на різних електронних пристроях, та маємо змогу приступити до розробки самого додатку.

4.2 Використання бази даних Postgresql

На сьогоднішній день розробка програмного забезпечення для будь-якої галузі неможлива без використання баз даних, які необхідні для збереження та структурування великої кількості інформації що є в майже в усіх проектах.

Шлях розробки бази даних починається з завантаження та інсталяції необхідного програмного забезпечення. Наступним кроком є відкриття відповідного меню для подальшої розробки бази даних, для чого необхідно відкрити термінал та в цільовій директорії, прописати команду *sudo -u postgres psql* про успішне виконання даної команди свідчить текст з якого починається введення команд *postgres=#*.



```

kira@kira-MS-7B86: ~
CREATE ROLE
postgres=# GRANT ALL ON DATABASE test_database TO test_user;
GRANT
postgres=# /q
postgres=# \q
kira@kira-MS-7B86:~$ sudo -u postgres psql
psql (12.6 (Ubuntu 12.6-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# ^C
postgres=# CREATE DATABASE sweets_bd;
CREATE DATABASE
postgres=# CREATE USER admin WITH password 'admin';
CREATE ROLE
postgres=# GRAND ALL ON DATABASE sweets_bd TO admin;
ERROR: syntax error at or near "GRAND"
LINE 1: GRAND ALL ON DATABASE sweets_bd TO admin;
      ^
postgres=# GRANT ALL ON DATABASE sweets_bd TO admin;
GRANT
postgres=# \q
kira@kira-MS-7B86:~$ psql -h localhost sweets_bd admin
Password for user admin:
psql (12.6 (Ubuntu 12.6-0ubuntu0.20.04.1))

```

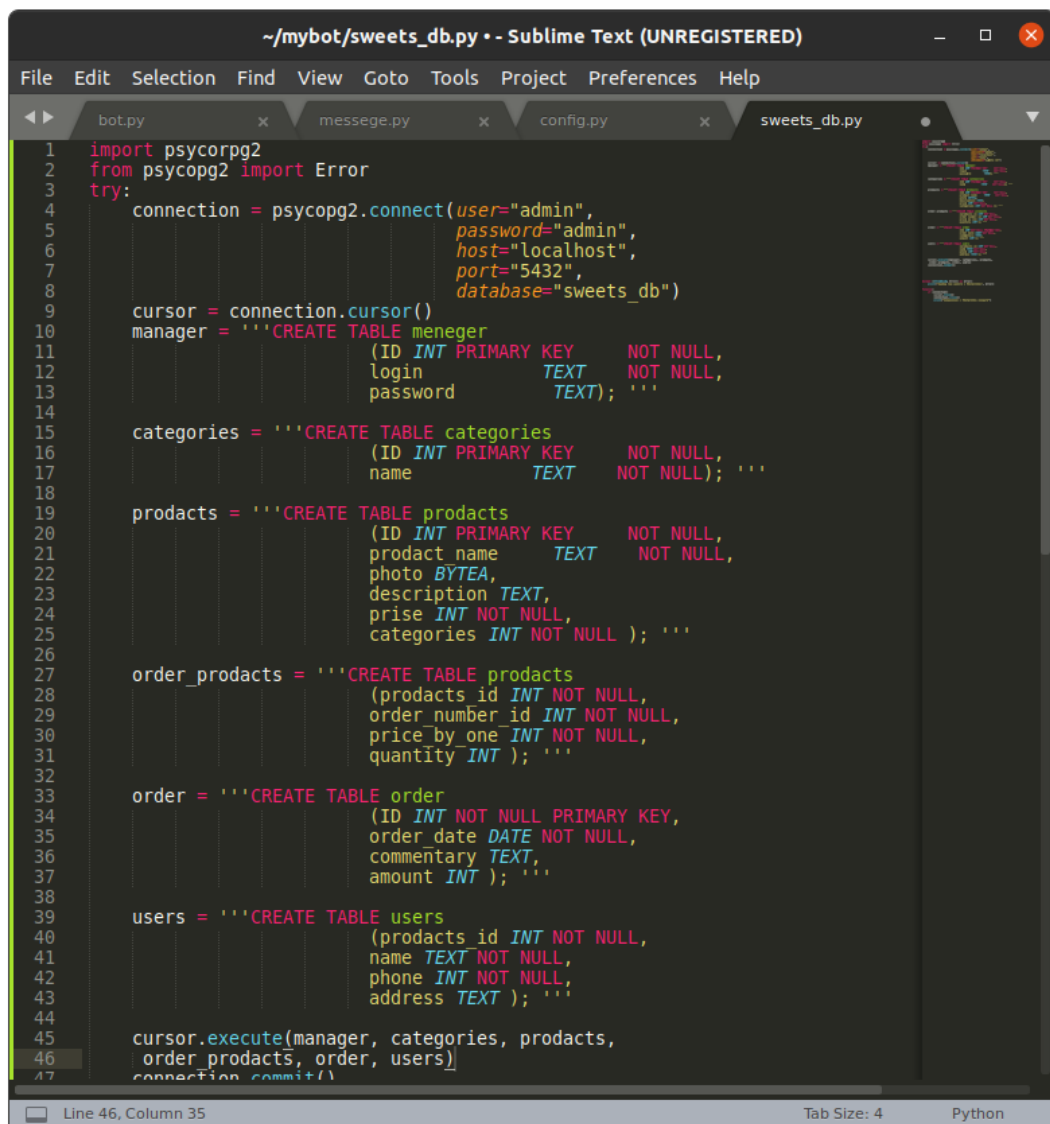
Рисунок 4.4 Початок роботи з Postgresql

Після виконання даних кроків можна починати створення бази даних для чого використовуємо команду *CREATE DATABASE* та вводимо назву бази даних *sweets_bd*; Далі необхідно створити користувача, що буде мати доступ до створення та редагування бази даних за допомогою команди *CREATE USER admin WITH password 'admin'*; де вказується логін та пароль користувача. Останній крок який потрібно виконати це надати користувачеві доступ до

відповідної бази за допомогою команди *GANT ALL ON DATABASE sweets_bd TO admin*.

Це першочергові кроки які потрібно виконати для подальшої розробки бази даних які можна побачити на рис 4.4.

Для подальшої розробки з використанням мови програмування Python необхідно обрати та під'єднати спеціальний модуль для підключення до PostgreSQL та наступного виконання SQL-запитів та інших операцій з базою даних.



```

~/mybot/sweets_db.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
bot.py x messege.py x config.py x sweets_db.py
1 import psycopg2
2 from psycopg2 import Error
3 try:
4     connection = psycopg2.connect(user="admin",
5                                   password="admin",
6                                   host="localhost",
7                                   port="5432",
8                                   database="sweets_db")
9     cursor = connection.cursor()
10    manager = '''CREATE TABLE manager
11                (ID INT PRIMARY KEY     NOT NULL,
12                 login      TEXT      NOT NULL,
13                 password   TEXT); '''
14
15    categories = '''CREATE TABLE categories
16                (ID INT PRIMARY KEY     NOT NULL,
17                 name       TEXT      NOT NULL); '''
18
19    products = '''CREATE TABLE products
20                (ID INT PRIMARY KEY     NOT NULL,
21                 product name TEXT      NOT NULL,
22                 photo BYTEA,
23                 description TEXT,
24                 prise INT NOT NULL,
25                 categories INT NOT NULL ); '''
26
27    order_products = '''CREATE TABLE products
28                (products id INT NOT NULL,
29                 order_number id INT NOT NULL,
30                 price_by_one INT NOT NULL,
31                 quantity INT ); '''
32
33    order = '''CREATE TABLE order
34                (ID INT NOT NULL PRIMARY KEY,
35                 order date DATE NOT NULL,
36                 commentary TEXT,
37                 amount INT ); '''
38
39    users = '''CREATE TABLE users
40                (products id INT NOT NULL,
41                 name TEXT NOT NULL,
42                 phone INT NOT NULL,
43                 address TEXT ); '''
44
45    cursor.execute(manager, categories, products,
46                  order_products, order, users)
47    connection.commit()
Line 46, Column 35                               Tab Size: 4   Python

```

Рисунок 4.5 - Код створення бази даних з використанням psycopg2

Існує багато модулів для роботи з базою даних, в даному проекті перевага було надана Psycopg2 бо це один з найстабільніших та популярніших модулів для Python для роботи з SQL. Основні переваги:

- Використовується в багатьох фреймворках Python та Postgres;
- Підтримується та працює з усіма версіями Python
- Потоко-безпечний та спроектований для роботи в багатопотокових застосунках.

Саме останній пункт є головною перевагою над іншими модулями, так як додаток активно використовує багатопоточність та в специфіці проекту вона є необхідною для швидкої роботи.

4.3 Тестування додатку

Додаток повинен працювати чітко, швидко, без збоїв забезпечувати якість продукту. Якість товарів і послуг може бути визначена як: загальна складовою продукту і експлуатаційні характеристики маркетингу, інжинірингу, виробництва та обслуговування, продуктів і послуг у використанні що буде відповідати очікуванням замовника.

Якість ПЗ характеризується трьома головними аспектами:

- якість програмного продукту;
- якість процесів ЖЦ;
- якість супроводу або впровадження.

Для тестування додатку було створено набір методів, та правил які використовуються тестувальниками програмного забезпечення. Результати тестування оформлено в тест-комплекс.

Тест-кейс це професійний документ тестувальника який має набір кроків для перевірки функціоналу, який описує всі кроки які потрібно виконати для отримання очікуваного результату.

Таблиця 4.1 - Тест-кейс для тестування функціоналу авторизації

Функціонал авторизації менеджера		
Передумова: Ввести команду для авторизації		
Крок тесту	Дані тесту	Очікуваний результат
Ввід даних в текстове поле та почергове відправлення	<ul style="list-style-type: none"> - Ввести команду /login - Ввести логін - Ввести пароль - Натиснути кнопку надіслати повідомлення 	<ul style="list-style-type: none"> - Менеджер зайшов у систему - Відкривається адміністративне меню додатку

Таблиця 4.2 - Тест-кейс функціоналу додавая товару до бд

Функціонал сторінки менеджера		
Передумова: Щоб перейти до адміністративного меню необхідно відкрити чат з ботом, ввести команду /login та пройти процедуру авторизації.		
Крок тесту	Дані тесту	Очікуваний результат
Додавання товарів до бд через меню менеджера	<ul style="list-style-type: none"> - Натиснути на кнопку Додати товар - Ввести та відправити назву товару - Ввести та відправити опис товару - Ввести та відправити опис фото товару - Ввести та відправити ціну товару 	<ul style="list-style-type: none"> - Почергове заповнення полів даних товару - При відправлені даних система без помилок переходить до наступного поля - Зберегти дані стосовного нового товару в БД та додати до меню

Було проведено тестування головного функціоналу адміністративного меню застосунку та перевірено роботоздатність реалізованого функціоналу на основі типових помилок у програмах та підтверджено якість розробленого функціоналу.

Звіт щодо виконання повтавленого завдання було оформлено в тест-кейси та представлено таблицях 4.1, 4.2 з використанням норм та документації тестувальників програмного забезпечення.

Таблиця 4.3 - Тест-кейс функціоналу роботи додання товару до кошику

Функціонал додання та видалення товару в кошику		
Передумова: користувач відкриває діалог з чат-ботом та вводить команду /start		
Крок тесту	Дані тесту	Очікуваний результат
Додання та видалення товарів до кошику	<ul style="list-style-type: none"> - Перейти до пункту Меню - Обрати категорію товару - Переглянути список товарів - Обрати декілька товарів - Натиснути кнопку Додати в кошик - Натиснути кнопку Кошик 	<ul style="list-style-type: none"> - Коректне відображення категорій меню - Коректне відображення товарів - Додавання відповідних товарів до кошику - Коректний підрахунок ціни на основі доданого товару - Перехід в кошик для подальшого редагування

Таблиця 4.4 – Тест-кейс функціоналу оформлення замовлення

Функціонал оформлення замовлення		
Передумова: користувач відкриває діалог з чат-ботом та вводить команду /start, додає декілька товарів до кошику та натискає кнопку Оформити замовлення		
Крок тесту	Дані тесту	Очікуваний результат
Оформлення замовлення користувачем	<ul style="list-style-type: none"> - Ввести та надіслати Ім'я - Ввести та надіслати номер телефону - Натиснути кнопку Продовжити - Обрати спосіб доставки кур'єром 	<ul style="list-style-type: none"> - Коректне відображення тексту - Користувач заповнює та надсилає контактні данні - Система групує та виводить користувачеві
	<ul style="list-style-type: none"> - Ввести на надіслати час доставки - Натиснути кнопку завершити оформлення замовлення 	<ul style="list-style-type: none"> - Після підтвердження коректності введення даних замовлення надсилається менеджеру

В таблицях 4.3, 4.4 представлено тест-кейси тестування тестування головного функціоналу користувачів та перевірено функціонал оформлення замовлення.

4.4 Інструкція користувача

Для того щоб розпочати роботу необхідно встановити Telegram та ввести тег бота @Sweet_Island_bot в поле для пошуку це необхідно для початку роботи.

Уся взаємодія з ботом відбувається з використанням клавіатури бота, рис 4.6 текстових команд, введенням та відправкою тексту використовуючи можливості месенджера.

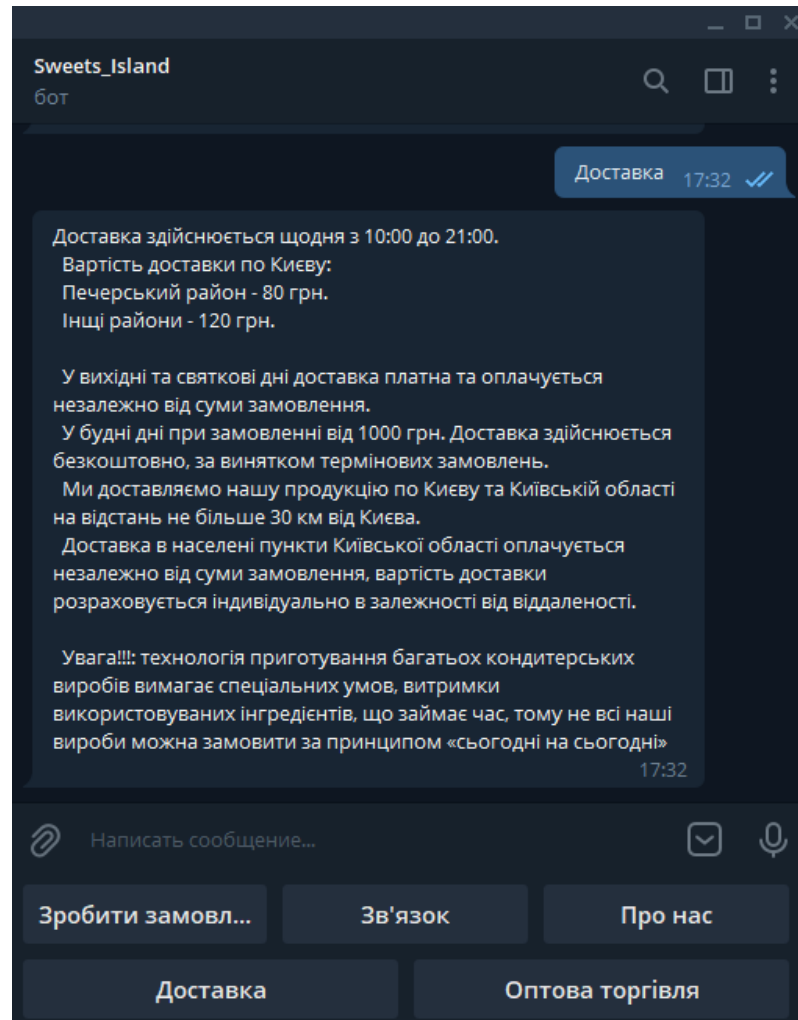


Рисунок 4.6 Клавіатура чат-боту

На головній сторінці користувач взаємодіє з інтерактивною клавіатурою бота, для того щоб оформити замовлення необхідно натиснути *Зробити замовлення* усі інші зображені кнопки носять інформаційних характер надають користувачеві можливість отримати корисну, та важливу інформацію стосовно закладу, та можуть відповісти на основні питання, що можуть виникнути.

Далі користувачеві буде надіслано повідомлення під яким буде знаходиться клавіатура з розділами меню, натиснув на будь-який з розділів користувач отримає меню та можливість додати до кошика.

Після того як користувач закінчить обирати товари необхідно натиснути *Оформити замовлення* де користувачеві буде показано сі додані товари та цін, після чого необхідно буде ввести Ім'я, телефон та обрати спосіб доставки, на вибір буде *Само вивіз* або *Кур'єр*. В першому випадку користувач має можливість подивитися на адресу закладу та вибрати на котру годину він бажає прийти та забрати замовлення. Якщо було вибрано другий спосіб то також необхідно ввести адресу доставки та годину, після чого натиснути *Далі* та перейти до наступного пункту.

Далі бот виведе усе замовлення користувачеві та запропонує перевірити сформоване замовлення, якщо все влаштовує то необхідно натиснути *Надіслати замовлення*, у разі якщо десь була допущена помилка то користувач повинен натиснути *Повернутися на крок* та виправити, після чого продовжити, бот відправить повідомлення, що заказ було успішно надіслано.

Це останній крок в ході оформлення замовлення, далі користувачеві необхідно трохи дочекатися телефонного дзвінка від менеджера та підтвердити замовлення за потреби можна виправити помилку, або додати щось до замовлення.

Висновок

У бакалаврській роботі проведено дослідження та розробка програмного забезпечення для реалізації

Основними результатами роботи є:

- досліджено варіанти реалізації проектів для реалізації продажу товарів через інтернет;
- досліджено вплив факторів на популярність месенджерів;
- проаналізовано існуючі застосунки системи замовлень;
- описано та змодульовано діаграму прецедентів;
- описано та змодульовано модель предметної галузі;
- розроблено та реалізовано програмне забезпечення;

Упровадження додатку допоможе користувачам зручно та комфортно замовляти товари з використанням мережі інтернет. Замовникам надає можливість виходу на перспективний поки ще не зайнятий ринок збуту товарів, спросити та пришвидшити методи обробки замовлень, отримати просту в редагуванні систему.

Розроблено додаток для продажу товарів в месенджері Telegram де реалізація збуту виконується послідовними повідомленнями в чаті з ботом з використанням звичайних та програмованих клавіатур для більш простої взаємодії з чат-ботом.

В майбутньому планується розширити функціонал адміністративного меню, розширити меню збуту товарів, реалізувати можливість перегляду минулих замовлень для користувачів, розробка системи бонусів. Та в подальшому буде перероблено та удосконалено структуру додатку з використанням отриманих знань та навичок з виправленням помилок та недоліків.

ПЕРЕЛІК ПОСИЛАНЬ

1. Python documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/>.
2. Telegram Bot API [Електронний ресурс] – Режим доступу до ресурсу: <https://core.telegram.org/bots/api>.
3. Aiogram documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.aiogram.dev/en/latest/>.
4. Most popular global messenger [Електронний ресурс] – Режим доступу до ресурсу: <https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/>.
5. UML diagram [Електронний ресурс] – Режим доступу до ресурсу: <https://www.uml-diagrams.org/>.
6. A Byte of Python [Електронний ресурс] – Режим доступу до ресурсу: <https://python.swaroopch.com/>
7. PostgreSQL documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postgresql.org/docs/>.
8. Information and general information about the Visual Studio [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/uk-ua/visualstudio/get-started/visual-studio-ide?view=vs-2019>.
9. Visual Studio Code documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com/docs>.
10. Development Environment [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techopedia.com/definition/16376/development-environment#:~:text=A%20development%20environment%20is%20a,referr ed%20to%20as%20the%20DSP>.
11. GitHub documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.github.com>

12. Edition Regina O. Obe., Leo S. Hsu PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database 3rd Edition / – 2017 - 198-201 с.
13. Russ M., Kim H. Learning UML 2.0: A Pragmatic Introduction to UML 1st Edition / - O'Really Media, Inc. 2006 - 50-55 с.
14. George Brand E-Commerce Business Model 2020: This Book Includes: Online Marketing Strategies, Dropshipping, Amazon FBA - Step-by-Step Guide with Latest Techniques to Make Money Online and Reach Financial Freedom / - 2020 – 80-84 с.
15. Grady Booch, James Rumbaugh, Ivar Jacobson The Unified Modeling Language User Guide second edition/ - Addison Wesley New York 2007 - 30-38 с.
16. Топ месенджерів в Україні та світі 2020 [Електронний ресурс] – Режим доступу до ресурсу: <https://marketer.ua/top-messengers-in-ukraine-and-the-world/>.
17. Telegram [Електронний ресурс] – Режим доступу до ресурсу: <https://telegram.org/?setln=en>.
18. Development Survey Result [Електронний ресурс] – Режим доступу до ресурсу: https://insights.stackoverflow.com/survey/2019#technology-_-most-popular-development-environments.

Додаток А
Демонстраційні матеріали