

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ТЕЛЕГРАМ-БОТА ДЛЯ КРАМНИЦІ.
TELEGRAM BOT DEVELOPMENT FOR A STORE.

Виконав: студент(ка) 4 курсу, групи КНТз-117

Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Абдуллаєв М.С.

(прізвище та ініціали)

Керівник Федорончак Т.В.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ПРЕ, ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 20 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Абдуллаєва Мамеда Самандар огли
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка Телеграм-бота для крамниці. Telegram Bot Development for a Store.

керівник проекту (роботи) Федорончак Тетяна Василівна., к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року № 103

2. Строк подання студентом проекту (роботи) 25 травня 2021 року

3. Вихідні дані до проекту (роботи) технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Опис програми. 3. Розробка програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Федорончак Т.В., доцент		
Нормоконтроль	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання “ 28 ” лютого 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз предметної області	1 тиждень	
2	Розробка та затвердження технічного завдання	2 тиждень	Технічне завдання
3	Проектування структури системи	3 тиждень	Структурна схема
4	Розробка схеми функціонування системи та модулів програми	3-4 тиждень	Функціональна схема. Модулі
5	Створення програмного коду системи	5-6 тиждень	Текст програми
6	Тестування та відлагодження телеграм-бота	7 тиждень	Працездатна система
7	Розробка розділів пояснювальної записки	7 тиждень	ПЗ
8	Розробка слайдів презентації	7-8 тиждень	Слайди презентації
9	Захист роботи	8 тиждень	

Студент(ка)

(підпис)

Абдуллаєв М.С.
(прізвище та ініціали)

Керівник проекту (роботи)

(підпис)

Федорончак Т.В.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: с.73, рис.24,табл.3, дод.2, джерел 23.

РОЗРОБКА ДОДАТКІВ, ТЕЛЕГРАМ-БОТ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, HTTP, API, PYTHON, АРХІТЕКТУРА ДОДАТКУ.

Об'єкт роботи – програмне забезпечення для магазинів.

Предмет роботи – телеграм-бот магазин, бот реалізований на Python.

Мета роботи - розробити телеграм-бот магазин, який використовує HTTP протокол.

Програмне забезпечення розроблено з використанням мови програмування Python, середовище розробки – PyCharm, особливістю розробки є її крос-платформна реалізація.

Матеріали, методи та технічні засоби: мова програмування Python, середовище розробки PyCharm, HTTP-запити, Json, Telegram API персональний комп'ютер під управлінням операційної системи Microsoft Windows 10.

Результати. Створено телеграм-бот магазин з можливістю додавання товару, та створення списку.

Висновки. Розроблено телеграм-бот магазин з можливістю додавання товару, та створення списку.

ABSTRACT

Explanatory note to the final qualifying work of the bachelor: 73 pages, 24 figures, 3 tables, 2 appendixes, 23 sources.

APPLICATION DEVELOPMENT, TELEGRAM-BOT, SOFTWARE, HTTP, API, PYTHON, APPLICATION ARCHITECTURE.

The object of work - software for shops.

The subject of the work is a telegram-bot shop, a bot implemented in Python.

The purpose of the work is to develop a telegram-bot store that uses the HTTP protocol.

The software is developed using the Python programming language, the development environment is PyCharm. a feature of the development is its cross-platform implementation.

Materials, methods and technical means: Python programming language, PyCharm IDE, HTTP-requests, Json, Telegram API, personal computer running under Microsoft Windows 10 operating system.

Results. Created a telegram-bot shop with the ability to add goods and create a list.

Conclusions. Developed telegram-bot shop with the ability to add goods and create a list.

ЗМІСТ

Перелік скорочень та умовних познач	8
Вступ	9
1 Аналіз предметної області	11
1.1 Концепція ботів	11
1.2 Боти для месенджерів	19
1.3 Переваги телеграм ботів	22
1.4 Постановка завдання роботи	25
1.5 Висновки до розділу	25
2 Опис програми	26
2.1 Дослідження серверних мов	26
2.2 Середовище розробки	34
2.3 Аналіз необхідних функцій	36
2.4 Алгоритм функціонування	36
2.5 Telegram-Bot API	38
2.7 Формат JSON	40
2.8 HTTP-запити	45
2.9 REST API	46
2.7 Висновки по розділу	47
3 Розробка програми	48
3.1 Створення функціоналу програми	48
3.2 Модулі в Python	49
3.2 Опис розроблених команд	50
4 Експлуатація, тестування та експериментальне дослідження програми	51

4.1 Опис застосування.....	51
4.2 Інструкція по експлуатації програми	51
Висновки.....	54
Перелік джерел посилання	55
Додаток А Текст програми	62
Додаток Б Слайди презентації.....	66

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ACM	– adaptive case management;
API	– Application Programming Interface;
CAPTCHA	– Completely Automated Public Turing test to tell Computers and Humans Apart;
DDoS	– Distributed Denial of Service;
DDatаka	– General Purpose Input Output;
HTML	– HyperText Markup Language;
IRC	– Internet Relay Chat;
IT	– Information technology;
JS	– JavaScript;
JSON	– JavaScript Object Notation;
KLM	– Royal Dutch Airlines;
NASA	– National Aeronautics and Space Administration;
REST	– Representational State Transfer;
SOAP	– Simple Object Access Protocol;
URL	– Uniform Resource Locator;
XML	– eXtensible Markup Language.
ЖКТ	– Житлово-комунальне господарство;
OS	– operating system;
ПК	– персональний комп'ютер;
ПЗ	– програмне забезпечення.

ВСТУП

Останні роки чат-боти в месенджерах і соціальних мережах все активніше впроваджуються компаніями з самих різних галузей. Актуальність вирішення підтверджується тим, що в соцмережах сконцентрована дуже велика і активна аудиторія, тобто потенційні споживачі. Про це свідчить велика кількість користувачів, наприклад в Telegram [1].

Також було зафіксовано, що в українців на смартфонах в середньому встановлено по чотири месенджера. Все це - вагомі причини для бізнесу, щоб активно просувати свій бренд в соцмережах і месенджерах. Чат-бот в цьому випадку - один з варіантів взаємодії з потенційною аудиторією. З його допомогою можна автоматизувати різні бізнес-процеси, підвищити їх ефективність і заощадити на додатковому персоналі.

Метою даної роботи є створення телеграм-бота для магазину, який використовує HTTP-запити та головні розроблені функції програми.

Актуальність роботи полягає у зростаючому рівні конкуренції у даній сфері послуг, що потребує автоматизації та оптимізації процесу торгівлі. Даний ПП дозволяє зменшити зусилля, необхідні для введення роздрібною торгівлі в Інтернет.

В якості інструментальної мови програмування для виконання роботи буде використовуватись Python та середовище розробки PyCharm IDE від компанії JetBrains. Для розробки телеграм-бота буде використовуватися HTTP запити. Орієнтація мови програмування Python на багатоплатформову розробку дозволяє використовувати даний додаток на більшості операційних систем, таких як Windows, Linux, Mac OS X, Android, IOS тощо, за умови що для платформи існує будь-який сучасний браузер.

Для досягнення поставленої мети роботи необхідно розв'язати такі завдання:

- аналіз предметної області;
- розробка технічного завдання;

- вибір середовища розробки;
- розробка схеми взаємодії модулів програми;
- конструювання програмного забезпечення для телеграм-бота магазину;
- тестування розробленого програмного забезпечення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

У ході аналізу необхідно розглянути концепцію інтернет ботів, обґрунтувати вибір платформи для його розміщення, обрати мову програмування і середовище розробки.

1.1 Концепція ботів

Інтернет бот, веб-робот, робот або просто бот, це програмний додаток, який виконає автоматизовані завдання (сценарії) через Інтернет.

Зазвичай боти працюють по мережі. Боти, які можуть спілкуватися один з одним, використовують для цього інтернет-сервіси: обмін миттєвими повідомленнями, (Twitterbots) або через Internet Relay Chat (IRC). В цілому, більше половини інтернет-трафіку складають боти, які взаємодіють з веб-сторінками, розмовляють з користувачами, сканують контент і виконують інші завдання.

Боти складаються з наборів алгоритмів, які допомагають їм у вирішенні поставлених перед ними завдань. Завдання, які зазвичай можуть виконувати роботи, включають бесіду з людиною (боти, які намагаються імітувати людську поведінку) або збір контенту з інших веб-сайтів. Існує багато типів ботів, розроблених для виконання найрізноманітніших завдань.

Зазвичай боти виконують прості і повторювані завдання набагато швидше, ніж людина. Найчастіше боти використовуються для сканування веб-сторінок, коли автоматизований сценарій витягує, аналізує і зберігає інформацію з веб-серверів.

Зусилля вебсерверів по обмеженню ботів розрізняються. На деяких серверах є robots.txt файлу, що містить правила поведінки ботів на цьому сервері. Будь-якому боту, що не відповідає правилам, теоретично може бути відмовлено в доступі або він може бути видалений з веб-сайту. Якщо опублікований текстовий файл не має пов'язаної програми / програмного

забезпечення / додатки, то дотримання правил є повністю добровільним. Неможливо забезпечити дотримання правил або гарантувати, що творець або розробник бота прочитає або підтвердить файл robots.txt.

Існує багато видів ботів серед них виділяють такі:

- соціальні боти;
- шкідливі боти;
- корисні боти;
- комерційні боти (чат-боти).

Соціальні боти - це алгоритми, які виконують повторювані набори інструкцій, щоб надати послугу або встановити з'єднання між користувачами соціальних мереж [2]. Такі алгоритми, призначені для імітації людської поведінки і створені щоб розмовляти по шаблонах, аналогічних тим, що використовує людина-користувач.

Виділяють п'ять основних категорій застосування для соціальних ботів [3].

Технічні боти - в їх структурі практично завжди є спеціально прописані програми. Їх основне завдання - накопичення лайків і коментарів під потрібними записами, створення великої кількості друзів для збільшення довіри до боту і поширення публікацій. Це найбільш популярний вид ботів, який використовується в будь-яких соціальних мережах: Facebook або Instagram тощо.

Бойові боти - створені для зниження репутації або Блокування певної сторінки в соціальній мережі, шляхом відправлення великої кількості скарг і негативних коментарів.

Зливні боти - це боти для поширення певної інформації. Спочатку використовують спеціальних ботів, які діють як реальні користувачі, але в певний момент починають поширювати недостовірну інформацію. Таким чином створюється фальшиве джерело інформації.

Гіперболізовані боти - призначені для здобуття довіри клієнтів конкурента з подальшим створенням антиреклами. З першого погляду

фальшивий користувач повністю підтримує ідеологію і погляди опонента. Але в певний момент починає поширювати гіперболізовану (перебільшену) інформацію.

Інтелектуальні бойові боти або просто «Тролі» - головне завдання такого бота спілкуватися на вузькоспеціалізовану тематику в певній темі. Крім цього, такий бот часто може переходити на образи чи провокаційні висловлювання відносно інших користувачів, відволікаючи увагу від основної теми бесіди [3],[5].

Одним із прикладів шкідливого використання ботів є координація та виконання автоматичної атаки на мережеві комп'ютери, (атака типу «відмова в обслуговуванні» з боку ботнету). Інтернет-боти або веб-боти також можуть використовуватися для здійснення шахрайства з кліками, і останнім часом вони також дуже поширились в комп'ютерних та мобільних іграх.

Існують шкідливі боти наступних типів:

- спам-боти;
- парсери веб-сайтів;
- реєстраційні боти;
- віруси та хробаки;
- DDoS- атаки;
- ботнети, зомбі-комп'ютери і т. д.

Спам-боти - це боти які збирають адреси електронної пошти зі сторінок контактів або гостьових книг. Завантажені програми, які віднімають пропускну здатність, блокуючи цілі веб-сайти. Спам-боти, які намагаються спрямувати людей на шкідливий веб-сайт, іноді зустрічаються в розділах коментарів або на форумах різних веб-сайтів. В'ю-боти створюють фальшиві перегляди [5].

Парсери - це боти які захоплюють контент веб-сайтів і повторно використовують його без дозволу на автоматично створених інформаційних порталах.

Реєстраційні боти - це боти які реєструють певну адресу електронної пошти в численних службах, щоб повідомлення з підтвердженням заповнювали поштову скриньку і відволікали від важливих повідомлень [4].

Вірус - це програма, яка має здатність до прихованого самовідтворення. Одночасно зі створенням власних копій віруси можуть завдавати шкоди: знищувати, пошкоджувати, викрадати дані, знижувати або й зовсім унеможлиблювати подальшу працездатність операційної системи комп'ютера.

Комп'ютерні віруси мають таку назву через їх схожість на біологічні віруси. Також як біологічні віруси проникають в тіло і інфікують клітини, комп'ютерні віруси потрапляють в комп'ютери і заражають файли. Обидва типи вірусів можуть репродукувати себе і поширюватися шляхом передачі інфекції від однієї зараженої системи до іншої. Як і біологічний вірус є мікроорганізмом, так і комп'ютерний вірус є мікропрограмою.

DoS – це зловмисна атака на обчислювальну систему з метою довести її до відмови, тобто створення таких умов, при яких сумлінні користувачі системи не зможуть отримати доступ до системних ресурсів (серверів), або цей доступ буде ускладнений. Відмова атакованою системи може бути і кроком до оволодіння певною інформацією (якщо в нештатній ситуації ПО видає будь-яку критичну інформацію - наприклад, версію, частину програмного коду і т. д.). Але частіше це міра економічного тиску: вимушена бездіяльність служби, що приносить дохід, витрати на підтримання і заходи по відновленню після атаки завдають економічної шкоди [5].

Якщо атака виконується одночасно з великої кількості комп'ютерів, говорять про DDoS-атаку [4] (від англ. Distributed Denial of Service, розподілена атака типу «відмова в обслуговуванні»). Така атака проводиться в тому випадку, якщо її цілю є відмова в обслуговуванні добре захищеної великої компанії або урядової організації.

Спершу зловмисник сканує велику мережу за допомогою спеціально підготовлених сценаріїв, які виявляють потенційно слабкі вузли. Обрані вузли піддаються нападу, і зловмисник отримує на них права адміністратора. На

захоплені вузли встановлюються троянські програми, які працюють у фоновому режимі [4]. Тепер ці комп'ютери називаються комп'ютерами-зомбі, їх користувачі навіть не підозрюють, що є потенційними учасниками DDoS-атаки. Далі зловмисник відправляє певні команди захопленим комп'ютерів і ті, в свою чергу здійснюють колективну DoS-атаку на цільовий комп'ютер. Існує також програми для добровільної участі в DDoS-атаках.

У деяких випадках до фактичної DDoS-атаки призводить ненавмисна дія, наприклад, розміщення на популярному інтернет-ресурсі посилання на сайт, розміщений на не дуже продуктивному сервері. Великий наплив користувачів призводить до перевищення допустимого навантаження на сервер і, отже, відмови в обслуговуванні частини з них.

В даний час DoS і DDoS-атаки дуже популярні, так як дозволяють довести до відмови практично будь-яку погано захищену систему, не залишаючи юридично вагомих доказів.

Для захисту від мережевих атак застосовується ряд фільтрів, підключених до інтернет-каналу з великою пропускною здатністю. Фільтри діють таким чином, що послідовно аналізують трафік, виявляючи нестандартну мережеву активність і помилки. У число аналізованих шаблонів нестандартного трафіку входять всі відомі на сьогоднішній день методи атак, в тому числі реалізовані і за допомогою розподілених бот-мереж. Фільтри можуть реалізовуватися як на рівні маршрутизаторів, так і спеціалізованими апаратними засобами.

Зловмисні боти можуть також використовуватися для скупки місць на концерти, які користуються підвищеним попитом, особливо квитковими брокерами, які перепродають квитки [5]. Ці боти проходять процес покупки квитків на розважальні заходи і отримують кращі місця, купуючи якомога більше місць. Боти часто використовуються в багатокористувацьких мережевих рольових іграх для збору ресурсів, отримання яких в іншому випадку вимагало б значного часу або зусиль; це проблема для більшості ігрових онлайн-економік. Боти також використовуються для збільшення

переглядів відео на YouTube. Боти використовуються для збільшення кількості трафіку в аналітичних звітах з метою вилучення грошей від рекламодавців.

Ботів можна використовувати на інтернет-форумах для автоматичної публікації підбурливих або безглузвих повідомлень, щоб порушити роботу форуму і розлютити користувачів.

Найбільш широко використовуваний метод захисту від ботів - це використання CAPTCHA[7], який представляє собою форму тесту Тьюринга, використовуваного для розрізнення людини-користувача і менш складного бота на базі штучного інтелекту за допомогою графічно закодованого тексту, що розпізнається людиною. Приклади постачальників включають Recaptcha і комерційні компанії, такі як Minteye, Solve Media і NuCaptcha. Однак такі перевірки не є надійним засобом захисту від ботів, оскільки їх часто можна обійти за допомогою комп'ютерного розпізнавання символів, дірок в безпеці і т.д.

Комерційний бот - це бот який спрямований на збільшення обсягу продажів. Пряма взаємодія з замовником є основою продажів і підвищення задоволеності покупців в умовах роздрібного ринку з дуже жорсткою конкуренцією.

Основні вимоги до цих ботів: продаж, підтримка користувачів, зворотний зв'язок.

Ще одним із головних завдань комерційного-чат-бота є максимальний обсяг продажів. Обрається найбільш популярний месенджер серед цільової аудиторії і відкривається новий торговий канал. Вказується месенджер, в якому знаходиться ваш бот і доступні методи оплати. Останнім кроком створення комерційного чат бота є сповіщення потенційних покупців про його наявність.

Після впровадження чат-бота навантаження на call-центри знижується до 50 відсотків. Це зумовлено тим, що користувачі вважають за краще писати

про свою проблему, а не дзвонити в службу підтримки. Комерційний чат бот є найбільш популярним сценарієм використання чат-бота в бізнесі.

Комерційний чат-бот, також може приймати зворотній зв'язок від клієнтів і керувати рекомендаціями відповідно до придбаних клієнтами товарів. Такий чат-бот представляє чек за виконану покупку, а також дає можливість оцінити сервіс. Близько 15% учасників програми лояльності використовують чат-бот, щоб залишити зворотний зв'язок або отримати інформацію про найсвіжіші пропозиції і знижки. Близько 30% користувачів вважають за краще залишати зворотний зв'язок у формі, запропонованій чат-ботом або у формі опитувань.

Корисні боти - це боти які взаємодіють між людьми і соціальними роботами і є дуже поширеними в повсякденному житті [8]. Вони спрямовані на те, щоб полегшувати повторюванні дії наприклад: боти перекладачі, словникові боти, боти сповіщувачі, ЖКХ боти і т.д.

Використання ботів дає кілька переваг. Боти завжди доступні, що забезпечує швидке і зручне обслуговування клієнтів. Ботів можна використовувати для різних цілей, наприклад для спілкування в чаті або відповідей для надання відповідей на запитання і реклами. Боти також скорочують витрати на робочу силу, що, в свою чергу, приносить компанії більше прибутку.

Компанії та клієнти можуть отримати вигоду з інтернет-ботів. Інтернет-боти дозволяють клієнтам зв'язуватися з компаніями без необхідності спілкуватися з людьми. KLM Royal Dutch Airlines розробила чат-бота, який дозволяє клієнтам отримувати посадочні талони, нагадування про реєстрацію та іншу інформацію, необхідну для польоту [9].

Було багато суперечок з приводу використання ботів в функції автоматичної торгівлі, тобто використання комерційних ботів(чат-ботів). Веб-сайт аукціонів eBay подав до суду, намагаючись заборонити сторонній компанії використовувати ботів для обходу свого сайту в пошуках вигідних покупок; цей підхід мав неприємні наслідки для eBay і привернув увагу інших

ботів. Біржа ставок Betfair в Сполученому Королівстві побачила такий великий обсяг трафіку, що виходить від ботів, що запустила API WebService, призначений для програмістів-ботів, за допомогою якого вона може активно керувати взаємодіями з ботами. Відомо, що ферми ботів використовуються в онлайн-магазинах додатків, таких як Apple App Store і Google Play, для маніпулювання позиціями [6] або підвищення позитивних оцінок / відгуків.

Чат бот - це поширена і нешкідлива форма інтернет-ботів. Компанії використовують інтернет-ботів для збільшення онлайн-взаємодії і оптимізації комунікації. Компанії часто використовують ботів, щоб скоротити витрати, замість того, щоб наймати людей для спілкування зі споживачами. Такі чат-боти використовуються для відповіді на питання клієнтів. Наприклад, Domino's розробила чат-бота, який може приймати замовлення через Facebook Messenger. Чат-боти дозволяють компаніям приділяти час своїх співробітників більш важливих справ [6].

Є багато переваг, які дає використання ботів, а також недоліки, такі як ризики, які можуть запропонувати інші боти. Деякі потенційні переваги ботів включають в себе:

- боти швидше ніж люди в повторюваних завданнях;
- економія часу для клієнтів;
- доступні 24/7;
- організації можуть зв'язатися з великою кількістю людей через додатки для обміну повідомленнями;
- боти налаштовуються;
- покращений інтерфейс користувача.

До деяких недоліків можна віднести:

- ботів можна налаштувати лише для виконання певних завдань, і вони ризикують неправильно зрозуміти користувачів;
- користувачі можуть зробити ботів шкідливими;
- ботів можна використовувати для розсилки спаму.

1.2 Боти для месенджерів

За статистикою в месенджерах спілкуються більше 75% відсотків користувачів інтернету. Це зумовлене їх швидкістю, простотою використання, і тим фактом, що вони безкоштовні.

Чат-бот для месенджерів - це зручний інструмент для тих користувачів, які хочуть швидко отримати відповідь на конкретне запитання. Шукати інформацію самостійно не потрібно, достатньо лише сформулювати і відправити запит.

Чат-ботів месенджерів використовують для таких завдань:

- проведення опитування відвідувачів та клієнтів;
- надання консультацій по продукції та послугам;
- прийняття заявок;
- збір відгуків і побажань клієнтів;
- відповіді на питання про майбутні заходи (конференції, семінари тощо);
- запис клієнта на прийом (до лікаря, юриста, в банк та ін.);
- рекомендація альтернативних або комплектних товарів;
- для після продажного спілкування[56].

Існують дві групи чат-ботів[6].

1. Стандартні. До цієї групи належать боти, які відповідають відвідувачеві на підставі ключових слів з його запиту. В даному випадку для кожної введеної фрази потрібно використати відповідну команду. Однак, якщо запит введено невірно, чат-бот буде не в змозі відповісти. Для уникнення непорозумінь можуть бути використані підказки і варіанти діалогу.

2. Самонавчальні. Дана група побудована за іншою схемою. Такі чат-боти ближче до штучного інтелекту, вони навчаються кращому розумінню мови у кожному діалозі. Цей помічник може дати відповідь, навіть якщо запит введено нестандартно.

З точки зору користувача, у чат-ботів є й інші переваги[6]:

- не потрібна установка нових додатків - це зберігає пам'ять пристрою;
- комунікація відбувається в знайомому інтерфейсі звичного месенджера;
- додатки для спілкування доступні на більшості пристроїв;
- розробники самі дбають про їх доступність.

Для спрощення класифікації види було згруповано за завданням, які такі помічники виконують для бізнесу:

- продавець;
- маркетинг;
- інтерфейс;
- інформатор;
- психолог.

Через торговий-чат-бот користувач може швидко отримувати інформацію про продукт і відповіді на свої запитання. Крім того, такий помічник може швидко підібрати аналоги і оформити замовлення.

Головна задача чат-бота продавця - збільшення числа замовлень. Схема роботи виглядає наступним чином:

- покупцеві пропонують купити товар або скористатися послугою;
- чат-бот може запропонувати додаткові опції до замовлення;
- якщо налаштувати інтеграцію з платіжною системою, то помічник відразу виставить покупцеві рахунок або переведе на сторінку онлайн оплати;
- завершальний етап діалогу - підписка на повідомлення про статус замовлення.

Такий чат-бот ефективно справляється із завданням збільшення продажів.

Головне завдання маркетингу - збір глядачів (аудиторії людей, які в тому чи іншому вигляді з вами вже взаємодіяли) і конвертація їх в покупців. Найчастіше такий чат-бот бере на себе додатково функцію інформування

клієнтів про продукцію. Зібрані таким ботом дані використовуються компаніями для розробки рекламних кампаній.

Наприклад, користувачеві потрібна юридична консультація. Чат-бот задає питання, далі, слідуючи алгоритму, дає стандартні відповіді і пропонує записатися на щось, наприклад, безкоштовне.

Чат-бот-інтерфейс - надає користувачам доступ до різних сервісів бізнесу. Це можуть бути веб-віджети або популярні месенджери: Viber, WhatsApp, Skype, чат в Facebook [11]. Так чи інакше роль «Інтерфейсу» задіяна в усіх видах чат-ботів. Віртуальний помічник обробляє запит користувача і пересилає його через API в інші сервіси [13]. Отримана відповідь чат-бот дає в тому ж каналі, де було поставлено питання.

Інформатор - це найпоширеніша модель віртуального помічника. Такий чат-бот займається поширенням інформації про продукцію та послуги компанії. Головне його завдання - швидко і ясно відповісти на запит клієнта.

Найчастіше такі віртуальні помічники працюють в сфері туризму і транспорту, коли потрібно підказати клієнтові варіанти перельотів, готелів та інших сервісів.

Чат-бот психолог це - цифровий терапевт, який лежить в основі платформи для терапевтичних відкриттів. Один з перших ботів, в якому вивчаються методи лікування розладів, пов'язаних з вживанням психо-активних речовин, за допомогою мобільних додатків.

При стресі або депресії людям буває складно поговорити з кимось про свої проблеми. Багатьом в складних життєвих обставинах міг би допомогти якраз чат-бот. Така програма запитує у користувачів про настрої, самопочутті, планах на день і інших аспектах психологічного здоров'я. Один з найуспішніших проектів в цій сфері - віртуальний помічник Woebot, який був розроблений вченими з Стенфорда [16].

Правильний вибір ролі для чат-бота - це ключовий момент при розробці віртуального помічника.

Чат-бот може бути налаштований практично в будь-яких популярних месенджерах. Найкращий для бізнесу варіант, коли такий помічник є на всіх популярних платформах. Крім того, чат-ботів часто розміщують на веб-сайтах. Нижче наведено список найпопулярніших систем, де можна впровадити віртуального помічника [11].

Telegram - на сьогоднішній день ця платформа є найпопулярнішою для розміщення чат-ботів.

Створити помічника в Telegram просто і зручно. Платформа підтримує, функціонал для додавання кнопок в діалог. Співрозмовника-робота легко додати до групи або у бесіду, скориставшись відповідною функцією. Боти, як і інші канали, мають ім'я в формі @name, що полегшує їх пошук.

Facebook-месенджер має багато плюсів і корисних функцій: додавання кнопки в діалог, створення повноцінного меню, шаблони для товарів і послуг. Цей канал зв'язку найкраще підходить для інфопродуктів, рекламних заходів, сервісів для бізнесу. Серед мінусів виділяють такі: непрозора політика зберігання даних, обмеження інтерфейсу [11].

Найбільш популярний Viber в Білорусі і на Україні. Як і в інших месенджерах, ви зможете додати діалогові кнопки, а також оформити підписку бота на теми. Недоліки: цим месенджером користуються в основному люди у віці від 25 до 44 років [11].

1.3 Переваги телеграм ботів

Переваги телеграм бота [13]:

- відправка файлів різних форматів, розміром до 1,5 Гб;
- можливість створення каналів;
- усі файли зберігаються в хмарі і доступні з усіх пристроїв;
- швидкість доставки повідомлень;
- безпека;

– можливість створення групових чатів з великою кількістю учасників.

За декілька останніх десятирічь дуже активний розвиток отримала технологія Internet, інтернет-сервіси та мобільні додатки. Вони дозволяють задовольнити потреби практично всіх користувачів, маючи певну функціональність, яка полегшує вирішення певних низки завдань, дозволяє знаходити практично будь-яку інформацію. Більшість сучасних компаній, фірм та установ мають свої власні веб-портали на яких розміщена певна інформація. Однак існує проблема неоднорідності і дуже низької якості даних, малої швидкості відповіді від сервісу та високої собівартості.

У більшості випадків дані погано підготовлені, додатки і сервіси мають надлишковий інтерфейс, що погіршує сприйняття інформації. Витрачається велика кількість часу на те, щоб перемикатися між сервісами або додатками для виконання певних завдань, що не є оптимальним варіантом. Так само через високу собівартість веб-сервісів і додатків в розробці і обслуговуванні страждає функціональність, в більшості випадків відсутня машинне навчання, алгоритми використовуються при розробці не є оптимальними, через що, в свою чергу, страждає продуктивність і якість програми.

Виходячи з перерахованих вище проблем все частіше в життя суспільства впроваджуються чат-боти. Чат-бот створюється і навчається для певного спектру завдань.

Поширення і використання чат-ботів призвело до UX-парадигмі взаємодії «messaging-as-an-interface» [11]. Саме в месенджерах, аудиторія яких зараз в світі тільки зростає, чат-боти знайшли благодатне середовище для розвитку.

Слід зазначити, що завдяки тому, що аудиторія користувачів месенджерів постійно зростає, пропорційно зростає і попит на користування чат-ботами, замість окремих додатків або веб-сервісів. Використовувати чат-бот безпосередньо вбудований в месенджер набагато зручніше, оскільки потрібно менше зусиль і ресурсів для установки, взаємодія з чат-ботом

простіше ніж використовувалася програма, так як спілкування наближене до людського.

Зараз всі великі корпорації такі як WhatsApp, Google, Facebook, Skype роблять великі капіталовкладення в розвиток чат-ботів. Найбільш популярними можна назвати такі чат-боти таких корпорацій як Slack, Skype, Viber, Telegram. Всі вони інтегруються в месенджери і мають відкрите API і готові платформи для розробки, мають можливість вибору мови програмування для створення чат-бота, що збільшує аудиторію зацікавлених в розробці користувачів і не вимагає вкладення ресурсів для вивчення нової технології [11].

Нижче наведено основні спільні загальні риси між Slack, Skype, Viber і Telegram. Для всіх перерахованих вище платформ існує можливість інтеграції нейронних мереж для навчання ботів, щоб зробити їх не механічними, які відповідають на чітко поставлені команди, а зробити їх учнями і більш інтелектуальними, які вміють аналізувати отримані дані і на підставі вибірки виконувати найбільш сприятливу дію [11]. Також існує можливість як інтеграції чат-бота в бесіди, так і взаємодію з ним в якості співрозмовника.

Порівнюючи платформу Telegram зі Slack, Skype, Viber, можна виділити такі відмінні риси на користь Telegram, як хмарне зберігання всіх даних листувань, яке реалізовано тільки в Telegram, що вбереже користувачів від втрати важливих даних, наприклад, якщо у користувача Telegram був бот через який здійснювалась оплата, дані про транзакції не будуть загублені при переході на інший пристрій. У інших платформах що розглядаються, хмарне збереження даних відсутнє. Так само в Telegram існує двофакторна автентифікація користувачів, що робить використання цієї платформи більш захищеним.

У Telegram є власне зашифроване хмарне сховище, розподілене по різним юрисдикціям, і воно захищене набагато краще, ніж сховища Google і Apple [11]. Це робить можливим використання ботів в корпоративних цілях, гарантуючи конфіденційність даних. Slack месенджер, який використовується

в основному для корпоративного спілкування, в якому вся історія листування може бути збережена і передана третім особам. За аналогічним принципом, з точки зору конфіденційності, працює Skype.

Telegram дозволяє користувачам мати доступ до чатах відразу з декількох пристроїв одночасно, завдяки хмарній синхронізації. Таким чином, користувачі Mac, ПК, iPad і навіть сервери на Linux отримують стабільний досвід взаємодії з месенджером, винятками є лише секретні чати, для яких не застосовується збереження даних в хмарні сховища [13].

1.4 Постановка завдання роботи

Передбачається, що на перших етапах життєвого циклу проекту будуть вибрані і сформовані основні архітектурні структури відповідно до умов завдання.

Метою дипломної роботи є розробка програмно-продукту телеграм-бота крамниця.

Для досягнення мети роботи необхідно вирішити такі завдання:

- розробити архітектуру програми;
- обрати мову програмування;
- середу розробки;
- виконати програмну реалізацію;
- провести тестування.

1.5 Висновки до розділу

У ході аналізу були розглянуті концепції ботів, основні види та платформи для їх розміщення. Враховуючи переваги платформи Telegram, було прийнято рішення розробляти телеграм-бот (магазин).

2 ОПИС ПРОГРАМИ

2.1 Дослідження серверних мов

PHP - це поширена мова сценаріїв загального призначення з відкритим вихідним кодом.

PHP була спеціально розроблена для написання web-додатків (сценаріїв), що виконуються на Web-серверах.

Абревіатура PHP означає "Hypertext Preprocessor (Препроцесор Гіпертексту)". Синтаксис мови бере початок з C, Java і Perl. PHP досить простий для вивчення. Перевагою PHP є надання web-розробникам можливості швидкого створення web-сторінок, що динамічно генеруються.

Важливою перевагою мови PHP перед такими мовами, як Perl і C, є можливість створення HTML документів з впровадженими командами PHP.

Значною відмінністю PHP від будь-якого коду, що виконується на стороні клієнта, наприклад, JavaScript, є те, що PHP-сценарії виконуються на стороні сервера. Можливо також налаштувати свій сервер таким чином, щоб HTML-файли оброблялися процесором PHP. При такому підході клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання сценарію.

PHP дозволяє створювати якісні Web-додатки за дуже короткі терміни, отримуючи продукти, що легко модифікуються і підтримуються в майбутньому [14].

PHP простий для вивчання, і водночас здатний задовольнити запити професійних програмістів.

В області веб-програмування, зокрема серверної частини, PHP - це одна з найпопулярніших мов (разом з JSP, Perl і мовами, які використовуються в ASP.NET).

Має великий набір вбудованих засобів і додаткових модулів для розробки веб-додатків [14]. Основні з них:

- автоматичний витяг POST- і GET-параметрів, а також змінних оточення веб-сервера в зумовлені масиви;
- взаємодія з великою кількістю різних систем –управління базами даних через додаткові модулі (MySQL, MySQLi, SQLite, PostgreSQL, Oracle Database (OCI8), Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape і Apache Derby, Informix, Ovrimos SQL, Lotus Notes, DB ++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird і InterBase, Paradox File Access, MaxDB, інтерфейс PDO, Redis);
- автоматизована відправка HTTP-заголовків;
- робота з HTTP-авторизацією;
- робота з cookies і сесіями;
- робота з локальними і віддаленими файлами, сокетам;
- обробка файлів, що завантажуються на сервер;
- робота з XForms.

Входить в LAMP - поширений набір програмного забезпечення для створення та хостингу веб-сайтів (Linux, Apache, MySQL, PHP).

Серед сайтів, які використовують PHP - Facebook, Wikipedia, Yahoo! та Baidu.

JavaScript ("JS") - це повноцінна динамічна мова програмування, яка застосовується до HTML документу, і може забезпечити динамічну інтерактивність на веб-сайтах. Його розробив Brendan Eich, співзасновник проекту Mozilla, Mozilla Foundation і Mozilla Corporation [15].

JS це універсальна і проста у вивченні мова. JS досить компактна, але дуже гнучка. Існує велика кількість додаткових інструментів для мови JavaScript. До них відносяться:

- програмні інтерфейси додатка (API), вбудовані в браузер, що забезпечують різні функціональні можливості, такі як динамічне створення HTML і установку CSS стилів, захоплення і маніпуляція відео потоку, робота з веб-камерою користувача або генерація 3D графіки і аудіо семплів;

- сторонні API дозволяють розробникам впроваджувати функціональність в свої сайти від інших розробників, наприклад Twitter або Facebook;

- також можна застосувати сторонні HTML фреймворки і бібліотеки, що дозволяє прискорити створення сайтів і додатків.

JS є об'єктно-орієнтованою мовою, але процес створення прототипів що використовується в мові обумовлює відмінності в роботі з об'єктами в порівнянні з традиційними клас-орієнтованими мовами. Крім того, JS має ряд властивостей, притаманних функціональним мовам, - функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання - що додає мові додаткову гнучкість.

Незважаючи на схожий з C синтаксис, JS в порівнянні з мовою C має корінні відмінності:

- об'єкти з можливістю інтроспекції;
- функції як об'єкти першого класу;
- автоматичне приведення типів;
- автоматичне прибирання сміття;
- анонімні функції.

У мові відсутні:

- стандартна бібліотека: зокрема, відсутній інтерфейс програмування додатків по роботі з файловою системою, управління потоками введення-виведення, базових типів для бінарних даних;

- стандартні інтерфейси до веб-серверів і баз даних;
- система управління пакетами, яка б відстежувала залежності і автоматично встановлювала їх.

Ruby - інтерпретована, повністю об'єктно-орієнтована мова програмування з чіткою динамічною типізацією. Вона поєднує в собі Perl-подібний синтаксис з об'єктно-орієнтованим підходом. Також деякі риси запозичені з мов програмування Python, Lisp, Dylan і CLU [18]. Крос-платформна реалізація інтерпретатора мови Ruby поширюється на умовах

відкритого програмного забезпечення [16]. Код, написаний на Ruby, може бути зрозумілий навіть людині, яка не розбирається в програмуванні. На Ruby on Rails (RoR) були створені такі проекти, як Redmine, Twitter, Shopify, Basecamp, GitHub, Kickstarter, Airbnb і інші.

Можливості та характеристика мови Ruby [18]:

- високорівнева - зручна для розробників, з сильною абстракцією і використанням конструкцій натуральної або людської мови;
- динамічна - типи даних визначаються на етапі виконання програми, що збільшує швидкість розробки;
- інтерпретується - код на Ruby обробляється інтерпретатором в момент виконання без попередньої компіляції, це забезпечує незалежність від платформ і зменшує розмір виконуваних програм;
- мова програмування загального призначення - на відміну від предметно-орієнтованих, мови програмування загального призначення не створюються для застосування в специфічних областях;
- має лаконічний і простий синтаксис, частково розроблений під впливом Ада, Eiffel і Python;
- дозволяє обробляти виключення в стилі Java і Python;
- дозволяє перевизначити оператори, які насправді є методами;
- повністю об'єктно-орієнтована мова програмування. Всі дані в Ruby є об'єктами в розумінні Smalltalk. Наприклад, число «1» - це екземпляр класу;
- може динамічно завантажувати розширення, якщо це дозволяє операційна система;
- має незалежну від ОС підтримку багатопоточності.
- перенесений на безліч платформ. Він розроблявся на Linux, але працює на багатьох версіях Unix, DOS, Microsoft Windows (зокрема, Win32), Mac OS, BeOS, OS / 2 і т. д.

Python[19] - високорівнева мова програмування загального призначення з динамічною суворою типізацією і автоматичним управлінням

пам'яттю [19], орієнтована на підвищення продуктивності розробника, читання коду і його якості, а також на забезпечення перенесення написаних на ньому програм. Мова є повністю об'єктно-орієнтованою - все є об'єктами [19]. Незвичайною особливістю мови є виділення блоків коду пробільними відступами. Синтаксис ядра мови мінімалістичний, через це на практиці нечасто виникає необхідність звертатися до документації, сама ж мова відома як інтерпретована і використовується в тому числі для написання сценаріїв. Недоліками мови є часто більш низька швидкість роботи і більш високе споживання пам'яті написаних на ньому програм в порівнянні з аналогічним кодом, написаним на компільованих мовах, таких як C або C ++.

Python є мультипарадигмальною мовою програмування, що підтримує імперативне, процедурне, структурне, об'єктно-орієнтоване програмування, метапрограмування і функціональне програмування. Завдання узагальненого програмування вирішуються за рахунок динамічної типізації. Аспектно-орієнтоване програмування частково підтримується через декоратори, більш повноцінна підтримка забезпечується додатковими фреймворками. Такі методики як контрактне і логічне програмування можна реалізувати за допомогою бібліотек або розширень. Основні архітектурні риси - динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень з глобальною блокуванням інтерпретатора (GIL), високорівневі структури даних. Підтримується розбиття програм на модулі, які, в свою чергу, можуть об'єднуватися в пакети.

Еталонною реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ [19] і є стандартом де-факто мови [19]. Він поширюється під вільною ліцензією Python Software Foundation License, що дозволяє використовувати його без обмежень в будь-яких додатках, включаючи пропрієтарні. CPython компілює вихідні тексти в високорівневий байт-код, який виконується в стековій віртуальній машині. До інших трьом основним реалізаціям мови відносяться Jython (для JVM),

IronPython (для CLR / .NET) і PyPy [19]. PyPy написаний на підмножині мови Python (RPython) і розроблявся як альтернатива CPython з метою підвищення швидкості виконання програм, в тому числі за рахунок використання JIT-компіляції. Підтримка версії Python 2 закінчилася в 2020 році. На поточний момент активно розвивається версія мови Python 3. Розробка мови ведеться через пропозиції щодо розширення мови PEP (англ. Python Enhancement Proposal), в яких описуються нововведення, робляться коригування відповідно до зворотного зв'язку від спільноти і документуються підсумкові рішення [19].

Стандартна бібліотека включає великий набір корисних функцій, починаючи від функціоналу для роботи з текстом і закінчуючи засобами для написання мережевих застосунків. Додаткові можливості, такі як математичне моделювання, робота з обладнанням, написання веб-додатків або розробка ігор, можуть реалізовуватися за допомогою великої кількості сторонніх бібліотек, а також інтеграцією бібліотек, написаних на C або C++, при цьому і сам інтерпретатор Python може інтегруватися в проекти, написані на цих мовах [19]. Існує і спеціалізований репозиторій програмного забезпечення, написаного на Python, - PyPI. Даний репозиторій надає засоби для простої установки пакетів в операційну систему і став стандартом де-факто для Python. Станом на 2019 рік у ньому містилося понад 175 тисяч пакетів.

Python стала однією з найпопулярніших мов, вона використовується в аналізі даних, машинному навчанні, DevOps і веб-розробці, а також в інших сферах, включаючи розробку ігор. За рахунок читабельності, простого синтаксису і відсутності необхідності в компіляції мова добре підходить для навчання програмуванню, дозволяючи концентруватися на вивченні алгоритмів, концептів і парадигм. Налагодження ж і експериментування в значній мірі полегшуються тим фактом, що мова є інтерпретованою.

Мова застосовується багатьма великими компаніями, такими як Google, Facebook [19]. Станом на квітень 2021 року Python займає третє місце в рейтингу TIOBE популярності мов програмування з показником 11,03%.

«Мовою року» за версією TIOBE Python оголошувався в 2007, 2010, 2018 і 2020 році [19].

У Python багато готових бібліотек для вирішення завдань

Бібліотеками в програмуванні називають інструменти для вирішення конкретних типів завдань. Ось кілька прикладів популярних бібліотек для Python:

- Pygame. Бібліотека для створення невеликих ігор і мультимедійних додатків;
- NumPy. Бібліотека для роботи з штучним інтелектом і машинним навчанням. Використовується для складних математичних обчислень;
- Pandas. Бібліотека для роботи з великими даними;
- SQLAlchemy. Бібліотека для роботи з базами даних;
- Django, Flask. Бібліотеки для розробки серверної частини додатків.

Python використовують компанії-гіганти. Багато відомих нам компанії і організації використовують Python:

- Spotify і Amazon використовують Python для аналізу даних і створення рекомендацій;
- Walt Disney використовує Python як скриптову мову для анімації.
- YouTube і Instagram повністю написані на Python;
- NASA: їх система автоматизації процесів WAS теж створювалася засобами Python.

Python використовується у машинному навчанні і для роботи з великими даними. Ось що роблять з його допомогою:

- збирають дані купівельної активності, будують гіпотези і знаходять нові точки зростання компанії;
- розробляють алгоритми машинного навчання. Наприклад, Netflix написали свій рекомендаційний сервіс на Python;
- автоматизують рутинні завдання. Наприклад, простий скрипт на Python може зібрати всі посилання або картинки з вказаного сайту і зберегти їх в папку.

Порівняння кількості і зрозумілості рядків коду [19].

"Java" справляється в 5 рядків, використовується безліч дужок(рис.2.1):

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!"); // Display the string.  
    }  
}
```

Рисунок.2.1 Приклад використання "Java"

"C" працює схожим чином(рис.2.2):

```
#include<stdio.h>  
int main(int argc, char** argv)  
{  
    printf("Hello World");  
}
```

Рисунок.2.2 Приклад використання "C"

Python використовує один зрозумілий рядок(рис.2.3):

```
print "Hi there"
```

Рисунок.2.3 Приклад використання "Python"

Переваги Python перед іншими мовами програмування:

- низький поріг входження. Синтаксис Python більш зрозумілий для новачка;
- логічний, лаконічний і зрозумілий. У порівнянні з багатьма іншими мовами Python має простий синтаксис;
- підходить для різних платформ: і Linux, і Windows;
- є реалізація інтерпретаторів для мобільних пристроїв і непопулярних систем;

- широке застосування. Використовується для розробки веб-додатків, ігор, зручний для автоматизації, математичних обчислень, машинного навчання, в області інтернету речей. Існує реалізація під назвою Micro Python, оптимізована для запуску на мікроконтролерах (можна писати інструкції, логіку взаємодії пристроїв, організовувати зв'язок, реалізовувати розумний будинок);
- сильніше ком'юніті і багато конференцій;
- потужна підтримка компаній-гігантів ІТ-індустрії. Такі компанії, як Google, Facebook, Dropbox, Spotify, Quora, Netflix, на певних етапах розробки використовували саме Python;
- високий попит на ринку праці;
- існує багато якісних бібліотек з документацією;
- Python відрізняється суворою вимогою до написання коду (вимагає відступи), що сприяє організації і структуризації.

2.2 Середовище розробки

PyCharm - це сама інтелектуальна Python IDE з повним набором засобів для ефективної розробки на мові Python. Випускається в двох варіантах - безкоштовна версія PyCharm Community Edition і підтримує більший набір можливостей PyCharm Professional Edition. PyCharm виконує інспекцію коду на льоту, автодоповнення, в тому числі ґрунтуючись на інформації, отриманої під час виконання коду, навігацію по коду, забезпечує безліч рефакторингів [20].

Ключові можливості PyCharm:

- потужний і функціональний редактор коду з підсвічуванням синтаксису, авто-форматуванням і авто-відступами для підтримуваних мов;
- проста і потужна навігація в кодї;

- допомога при написанні коду, що включає в себе автодоповнення, авто-імпорт, шаблони коду, перевірка на сумісність версії інтерпретатора мови, і багато іншого;
- швидкий перегляд документації для будь-якого елемента прямо у вікні редактора, перегляд зовнішньої документації через браузер, підтримка docstring генерації, підсвічування, автодоповнення і багато іншого[20];
- велика кількість інспекцій коду;
- потужний рефакторинг коду, який надає широкі можливості щодо виконання швидких глобальних змін у проекті;
- повна підтримка свіжих версій Django фреймворка;
- підтримка Google App Engine;
- підтримка IronPython, Jython, Cython, PyPy wxPython, PyQt, PyGTK і ін;
- підтримка Flask фреймворка і мов Mako і Jinja2;
- редактор Javascript, Coffescript, HTML / CSS, SASS, LESS, HAML;
- інтеграція з системами контролю версій (VCS);
- UML діаграми класів, діаграми моделей Django і Google App Engine;
- інтегроване Unit тестування;
- інтерактивні консолі для Python, Django, SSH, відладчика і баз даних;
- повнофункціональний графічний відладчик (Debugger);
- підтримка схем найбільш популярних IDE / редакторів. таких як Netbeans, Eclipse, Emacs, емуляція VIM редактора;
- мови: Python (Versions: 2.x, 3.x), Jython, Cython, IronPython, PyPy, Javascript, CoffeScript, HTML / CSS, Django / Jinja2 templates, Gql, LESS / SASS / SCSS / HAML, Mako, Puppet , RegExp, Rest, SQL, XML, YAML;
- PyCharm має кілька кольірних схем, а також налаштовується підсвічування синтаксису коду;

- інтеграція з баг / issue-трекер, такими як JIRA, Youtrack, Lighthouse, Pivotal Tracker, GitHub, Redmine, Trac;
- величезна, постійно поповнюється колекція плагінів;
- багатоплатформний (Windows, Mac OS X, Linux);

Завдяки великим можливостям і перевагам, було обране середовище розробки PyCharm.

2.3 Аналіз необхідних функцій

У ході аналізу була виявлена необхідність розробки наступних функцій:

- функція додавання назви. Бот має підтримувати можливість додавання назви товару адміністратором;
- функція додавання зображення. Ця функція повинна надавати адміністратору можливість додавання зображення товару до буферу;
- функція перевірки правильності та повноти доданих даних; Функція повинна перевіряти наявність назви та зображення перед внесенням в БД;
- функція для збереження даних. Функція має зберігати назву та зображення у БД;
- функція для виведення усіх збережених даних.

2.4 Алгоритм функціонування

На рис. 2.4 приведено стандартну логіку процесу взаємодії з застосунком.

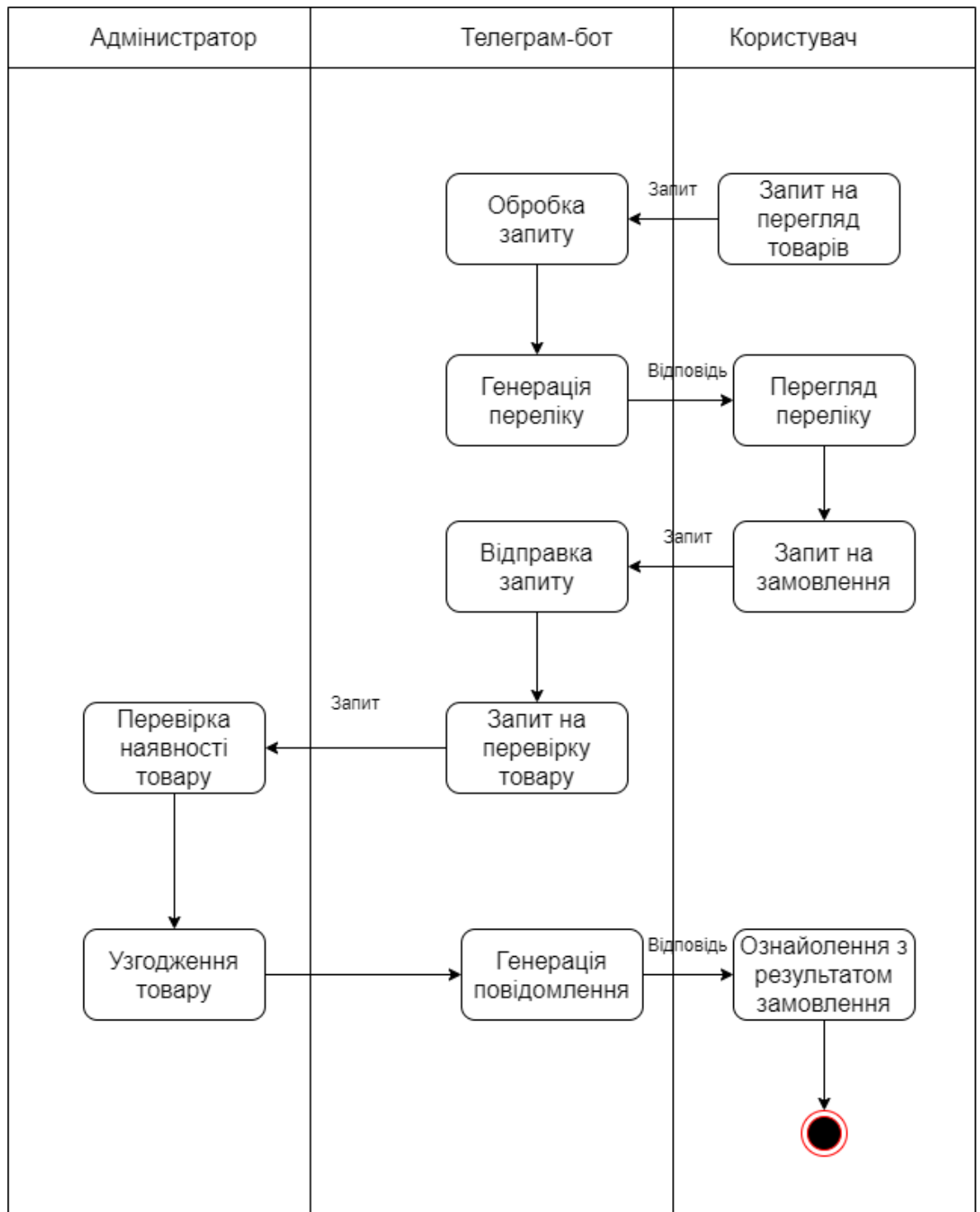


Рисунок 2.4 – Стандартний процес взаємодії з застосунком

Користувач надсилає запит на перегляд товарів. Телеграм-бот обробляє запит та генерує перелік товарів. Користувач отримає відповідь в виді переліку товарів, формує запит на замовлення товару. Запит відсилається телеграм-боту. Телеграм-бот відсилає запит на наявність товару адміністратору.

Адміністратор перевіряє наявність товару і узгоджує покупку. Бот відправляє повідомлення користувачу. Користувач ознайомлюється з замовленням.

2.5 Telegram-Bot API

Для створення застосунку для месенджера Telegram спочатку було зареєстровано застосунок через BotFather (рис. 2.5)[13].

В результаті виконання цих дій було отримано токен, який використовується для ідентифікації застосунку під час звернення до серверу Telegram:

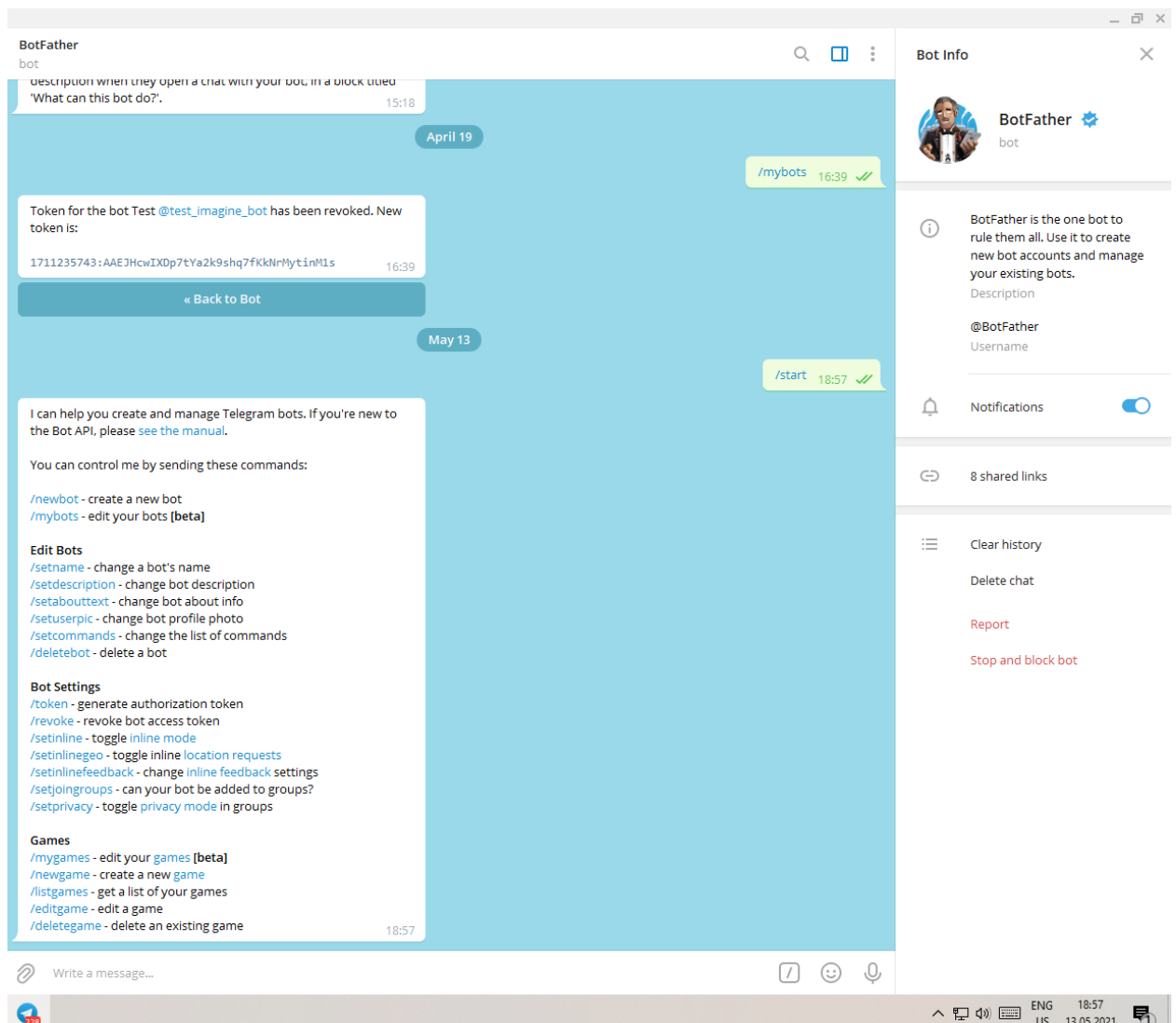


Рисунок 2.5 – Реєстрація програмного застосунку в месенджері Telegram

Для установки додаткових параметрів, таких як іконка чат-бота, вітальне повідомлення, опис чат-бота, а також видалення наявних чат-ботів, існують такі команди (таблиця 2.1).

Таблиця 2.1 – Доступні команди для зміни чат-ботів

Команда	Опис команди
/ setname	Змінює ім'я бота
/ setdescription	Змінює опис робота, що представляє собою короткий текст з описом бота.
/ setabouttext	Привласнює текст в поле «Про чат-боті»
/ setuserpic	Змінює аватар бота
/ setcommands	Дозволяє змінити список команд бота. Кожна команда складається з власне командного слова, що починається з символу косої риски («/») і короткого опису. Користувачі побачать список команд при введенні символу «/».
/ deletebot	Видаляє бота

Крім команд для зміни основних параметрів чат-бота існує ряд команд, які дозволяють виводити незмінні параметри (токен), а також привласнювати значення, які представлені в таблиці 2.2.

Таблиця 2.2 – Доступні команди для додаткової настройки чат-бота

Команда	Опис команд
/ token	Генерує токен авторизації
/ revoke	Відкликає токен доступу до боту
/ setinline	Включає вбудований режим (дозволяє звертатися безпосередньо до боту з будь-якого каналу, групи або чату, написавши його ім'я в поле відправки повідомлень)
/ setinlinegeo	Перемикає запити про розташування при використанні бота у вбудованому режимі.
/ setinlinefeedback	Змінює настройки зворотного зв'язку
/ setjoingroup	Визначає можливість додавання вашого бота в групи.
/ setprivacy	Включає режим конфіденційності. В цьому режимі бот отримує, інформацію і обробляє окремо для кожного користувача в чаті.

Після налаштувань на стороні Telegram і отриманні токена можна приступити до розробки програмної частини чат-бота.

2.7 Формат JSON

JSON (JavaScript Object Notation) – це формат для обміну даними, комфортний для зберігання, читання та написання як людиною, так і

комп'ютером. Він був заснований на підмножині відомої мови програмування JavaScript та визначений в стандарті ECMA-262 3rd Edition - December 1999 [20]. JSON – представляє собою майже звичайний текстовий формат, повністю незалежний від мови реалізації, але він використовує деякі правила, які знайомі програмістам С-подібних мов, таких як С, С ++, С # , Java, JavaScript, Perl, Python і багатьох інших. Саме ці властивості роблять JSON ідеальною мовою, яку можна використовувати для обміну даними.

JSON [21] з самого початку був заснований на двох структурах даних:

а) колекція пар ключ / значення. Ця концепція, у різних мовах може бути реалізована як об'єкт, запис, структура, словник, хеш, іменований список або асоціативний масив;

б) упорядкований список значень. Більшість мов програмування реалізують це як масив, вектор, список або послідовність. Це універсальні структури даних. Майже всі сучасні мови програмування мають змогу підтримувати їх в будь-якій формі. Досить легко припустити, що формат даних, який зберігаються в JSON, незалежний від мови програмування та повинен бути заснований на структурах вказаних нижче.

В нотації JSON це виглядає так [21]:

а) об'єкт – це неупорядкований набір пар ключ/значення. Об'єкт починається з відкритої фігурної дужки і закінчується закритою фігурною дужкою. Кожне ім'я в об'єкті супроводжується двокрапкою, а пари ключ/значення розділяються комою (рисунки 2.6 – 2.10);

Об'єкт

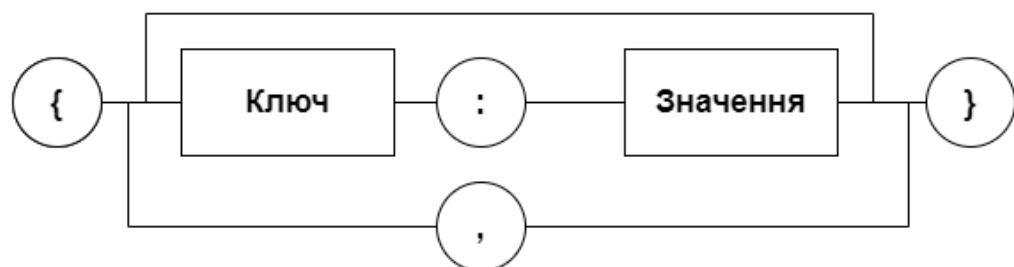


Рисунок 2.6 – Схематичне представлення об'єкта JSON

б) масив – це впорядкована колекція, яка може містити різні значення. Масив починається з квадратної дужки і закінчується цією дужкою. Значення в масиві розділяються комою (рисунок 2.7);

Масив

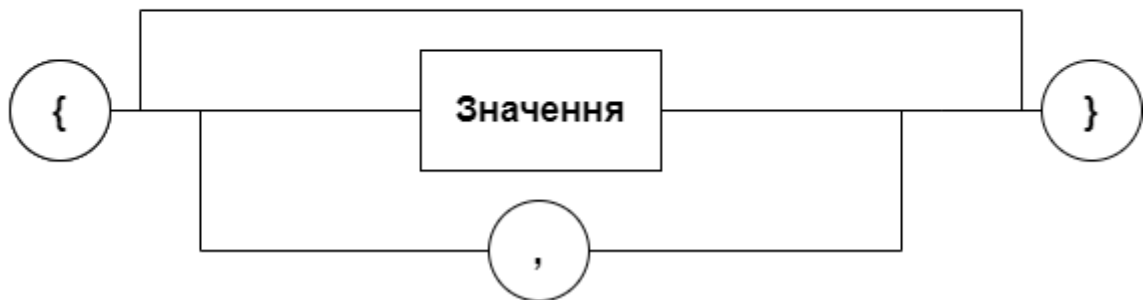


Рисунок 2.7 - Схематичне представлення масиву JSON

в) значення – може бути числом, рядком в подвійних лапках, true, false, null, об'єктом або масивом. Ці структури можуть бути вкладеними (рисунок 2.8);

Значення



Рисунок 2.8 – Схематичне представлення значення в JSON

г) рядок – це так званий набір символів в кодуванні Unicode, укладений в подвійні лапки, або символ, який використовує зворотний слеш в якості символу екранування. Символ представляється як один символьний рядок. Схожий синтаксис використовується в мовах програмування C і Java (рисунок 2.9);

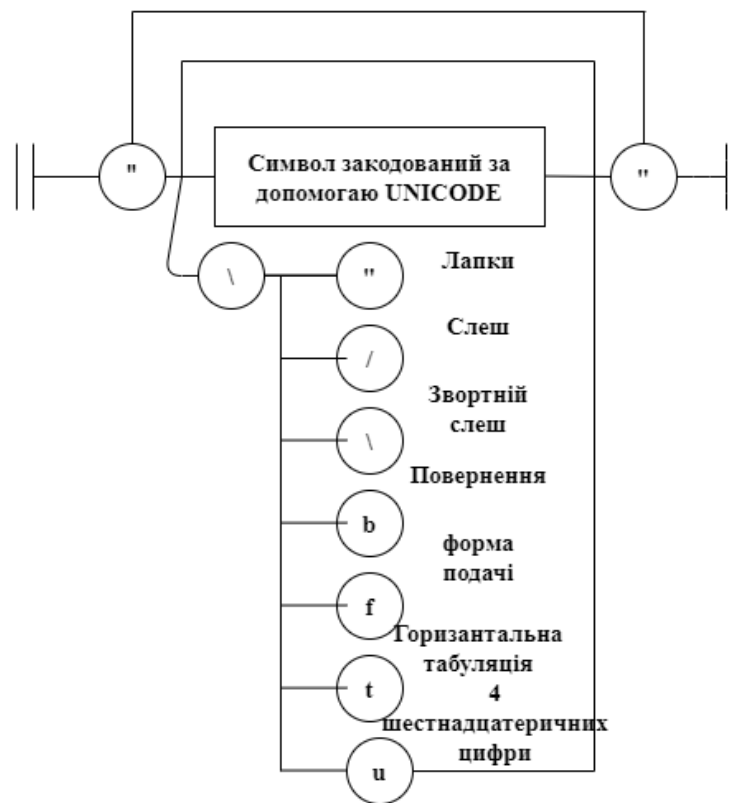
Рядок

Рисунок.2.9 - Схематичне представлення рядки в JS

г) Число в JSON передається так, як в C або Java, крім того, що використовується тільки десяткова система числення (рисунок 2.10).

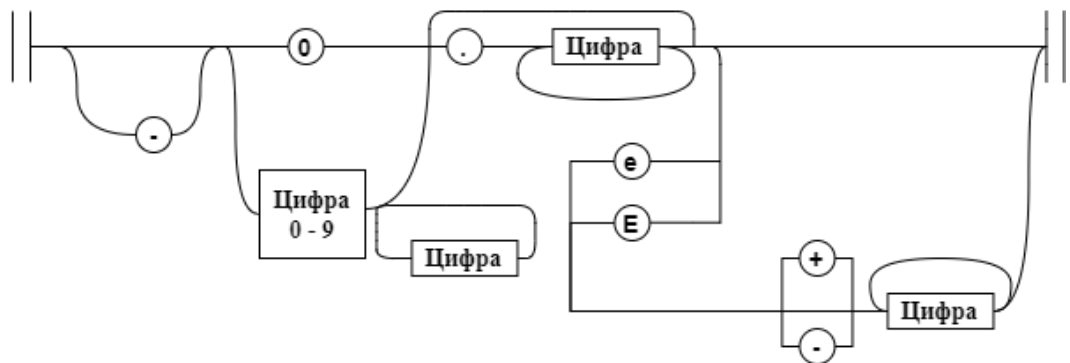
Число

Рисунок 2.10 – Схематичне представлення числа в JSON

2.8 HTTP-запити

HTTP - Hyper Text Transfer Protocol - протокол передачі гіпертексту, тобто такого тексту, коли разом з вмістом передається інформація по його відображенню. Спочатку HTTP призначався для передачі HTML-вмісту. Однак в даний час протокол HTTP використовується для передачі будь-якої інформації довільного змісту, залишаючись при цьому текстовим. Термін текстовий означає, що в ході роботи можуть передаватися тільки на екрані символи. Наприклад, в прийнятої і переданої інформації не може бути символу «0». Зрозуміло, що поняття будь-яка інформація має на увазі, що кожен байт рухаючись байт може приймати всі допустимі значення, тобто 0-255 включно. Така інформація ще називається бінарною. Передача бінарної інформації за допомогою текстового протоколу здійснюється за рахунок спеціального кодування, коли вихідна інформація на момент передачі видозмінюється таким чином, що містить тільки текстові символи. На клієнті після отримання повідомлення здійснюється зворотне перетворення. За допомогою такого механізму за допомогою HTTP можуть передаватися, наприклад, двовимірні або тривимірні зображення [22].

Крім того, HTTP широко використовується як транспортний протокол для передачі між різними програмами об'єктів (екземплярів класів). Інформація про структуру об'єкта (імена полів, їх модифікатори, значення і т. д.) Перетворюється в текстовий вигляд (наприклад, в XML або JSON). В текстовому вигляді передається по мережі і на клієнті здійснюється зворотне перетворення з тексту в програмний об'єкт. Одним з істотних переваг подібного методу передачі є та обставина, що клієнт і сервер можуть бути написані на різних мовах програмування, виконуватися на різних архітектурах і т. д. Таким чином, за допомогою протоколу HTTP зв'язуються між собою гетерогенні додатки в логічна єдине ціле (рис . 2.11) [22].

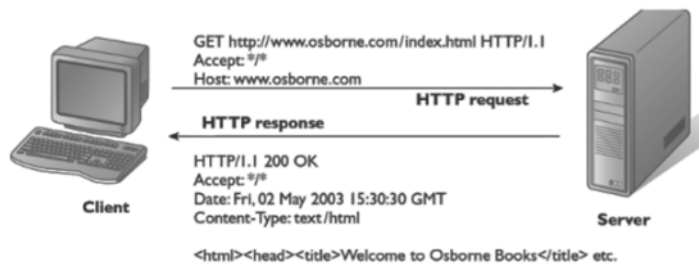


Рисунок 2.11 Приклад спілкування клієнта і сервера по HTTP-протокол

2.9 REST API

REST – це стиль архітектури програмного забезпечення, який використовується для побудови розподілених масштабованих веб-сервісів, які використовують HTTP запити. REST (Representational State Transfer) – стиль взаємодії компонентів розподіленого додатку в мережі. REST має узгоджений набір обмежень, що враховуються при проектуванні розподіленої гіпермедіа-системи. У визначених випадках (інтернет-магазини, пошукові системи) це спричинює підвищення продуктивності та змінення архітектури, а саме її спрощення. Компоненти в REST своєю взаємодією нагадують взаємодію клієнта і сервера в Інтернет. Виклик віддаленої процедури в мережі Інтернет може представляти собою звичайний HTTP-запит (зазвичай «GET» або «POST»; такий запит називають «REST-запит»), а всі необхідні для передачі дані передаються в якості параметрів запиту [23].

Для веб-служб, які були побудовані з урахуванням REST, застосовують так званий термін «RESTful». На відміну від веб-сервісів на основі SOAP, не існує єдиного офіційного стандарту для терміну RESTful веб-API. Оскільки REST є архітектурним стилем, в той час як SOAP є протоколом. REST не є стандартом сам по собі і не зважаючи на це, більшість RESTful-реалізацій використовують відомі стандарти, такі як: HTTP, URL, JSON і XML.

2.7 Висновки по розділу

У цьому розділі було проведено аналізу, мови програмування, обраного компілятора, середовища розробки визначено структуру програми та компоненти, що були використані на етапі розробки програми.

3 РОЗРОБКА ПРОГРАМИ

3.1 Створення функціоналу програми

Для роботи програми використовуються два файли `start.py`, `wizzard.py` (головний файл).

Файл `start.py` необхідний для стабільної роботи програми. Якщо в короткий проміжок часу часто підключатися до телеграм серверу то він завершує сеанс. Якщо трапляється помилка то бот її кидає файлу `start.py`. Тобто цей файл потрібен для того щоб відновлювати з'єднання телеграм-бота з серверами Telegram при необхідності.

На рисунку 3.1 зображено графічне представлення роботи файлу `start.py`.

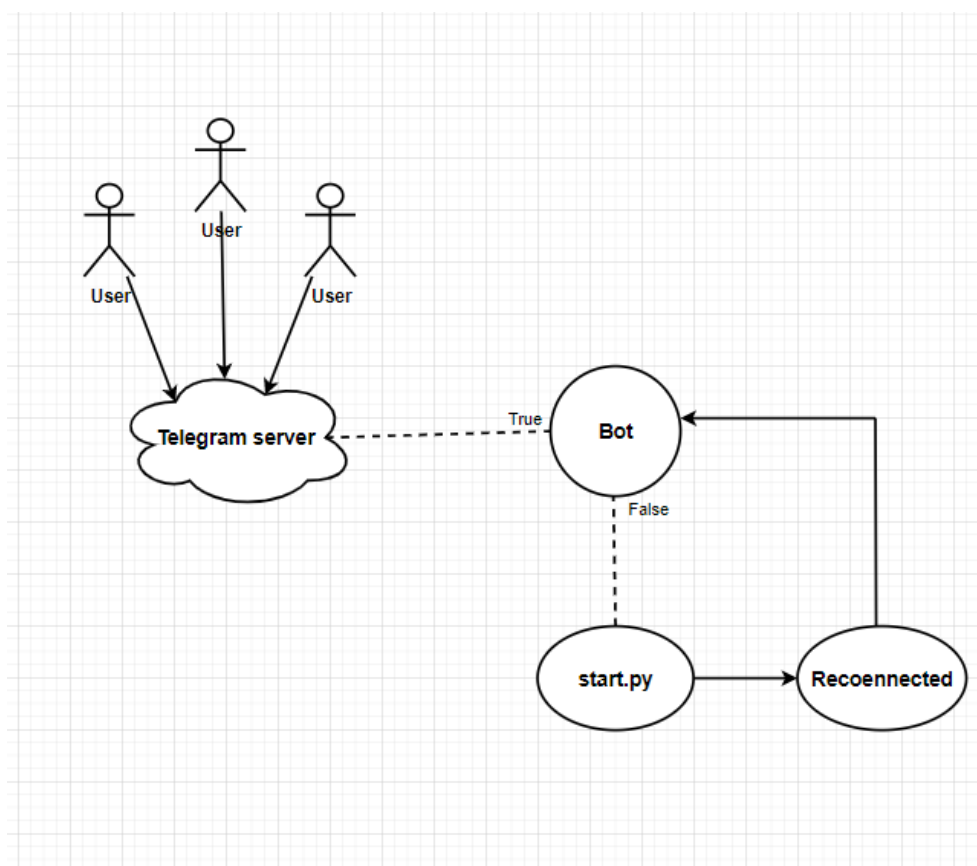


Рисунок 3.1 Робота файлу `start.py`.

Файл `wizzard.py` (головний файл). В ньому спочатку були підключені необхідні бібліотеки (Telebot, Shutil, RE). Потім присвоєний токен бота який потрібен для ідентифікації додатка при зверненні до сервера. Процес отримання токена описано в пункті 2.5.

Основними функціями програми є:

Add (message), addCard (message), check (message), save (message), all (message), phot (message). В таблиці 3.1 наведено опис головних функцій програми.

Таблиця 3.1 – Опис головних функцій програми

Назва функції	Опис
Add	Додавання картинки у телеграм-бот. Додавання картинки відбувається шляхом додавання адресу зображення до-буферу.
AddCard	Назва записується в нижньому регістрі до буферу, і виводиться повідомлення про успішний запис.
Check	Перевірка буфера
Save	Зберігає зображення і назва з буфера.
All	Видає всі збережені зображення з назвою.
Phot	Фото завантажується на сервер

3.2 Модулі в Python

В реалізованій програмі були використані такі модулі: Telebot, Shutil, Re.

Модуль TeleBot – потрібен для запита до TelegramBotAPI, використовується для спрощення і скорочення написаного коду. Всі типи визначені в `types.py`. Всі вони повністю відповідають визначенням типів API Telegram, за винятком `from` поля Message, яке перейменовано в `from_user` (оскільки `from` це зарезервований токен Python). До таких атрибутів, як `message_id` можна звертатися безпосередньо, наприклад: `message.message_id`.

Для написання програм які не сильно вибагливі до швидкості виконання варто звернути увагу, що атрибут `message.chat` може належати як певному користувачеві, так і до групового чату, це повністю окупається перевагами мови.

В `TeleBot` класі розташовані усі методи API. Щоб слідувати загальним стандартам про імена Python, вони перейменовані. Наприклад: `sendMessage` `send_message`, `editMessageText`, `edit_message_text`. Функція, прикрашена декоратором примірника `TeleBot` – обробник повідомлень. Обробники повідомлень складаються з одного або декількох фільтрів. Кожен фільтр повертає `True` або `False` для певного повідомлення і обробник отримує дозвіл на обробку повідомлення, якщо повертається `True`.

Модуль `Shutil` пропонує ряд високорівневих операцій над файлами і папками. Зокрема, передбачені функції, які підтримують копіювання і видалення файлів, переміщення, перейменування.

Модуль `Re` - регулярні вирази, також звані `regex`, використовуються практично у всіх мовах програмування. У `python` вони реалізовані в стандартному модулі `re`. Він широко використовується в природному обробці мови, веб-додатках, що вимагають перевірки введеного тексту (наприклад, адреси електронної пошти) і майже у всіх проектах в області аналізу даних, які включають в себе інтелектуальну обробку тексту.

3.2 Опис розроблених команд

Провідними командами телеграм-боту є:

- / Add-команда для додавання картинки у телеграм-бот. Додавання картинки відбувається шляхом додавання адресу зображення до-буферу;
- / Check- команда для перевірки буфером;
- / All- команда видає всі збережені зображення з назвою;
- / Save- команда зберігає зображення і назва з буфера.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Опис застосування

Розроблений додаток представляє собою телеграм бот магазин з можливістю додавання товару, збереження його та створення списку.

4.2 Інструкція по експлуатації програми

Для використання ППІ необхідно мати встановлене програмне забезпечення, таке як PyCharm, Python (версія 3.9.4). Також комп'ютер повинен мати:

- а) ОС: Windows10;
- б) ОЗУ: 4ГБ;
- в) доступ до Інтернету;
- г) смартфон (Android) або IOS.

Для програми потрібно близько 500 МБ вільного місця в оперативній пам'яті.

За допомогою команди /start телеграм-бот був запущений потім командою /add додаємо назву товару рис 4.1.

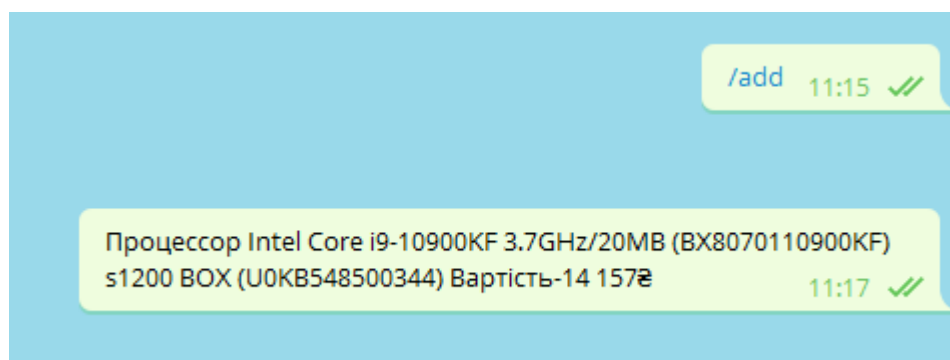


Рисунок 4.1 – Додавання назви товару

Після додавання назви треба додати картинку до товару за допомогою команди / add рис 4.2.

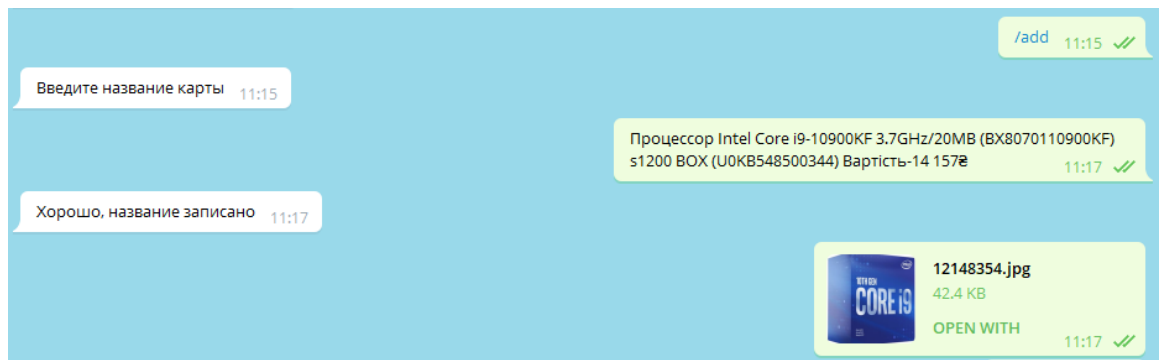


Рисунок 4.2 – Додавання картинки товару

Потім командою /check перевіряємо правильність створеної позиції рис 4.3.



Рисунок 4.3 – Перевірка створеної позиції

Після цього зберігаємо командою /save, а потім командою /all дивимося список створених позицій рис 4.4.

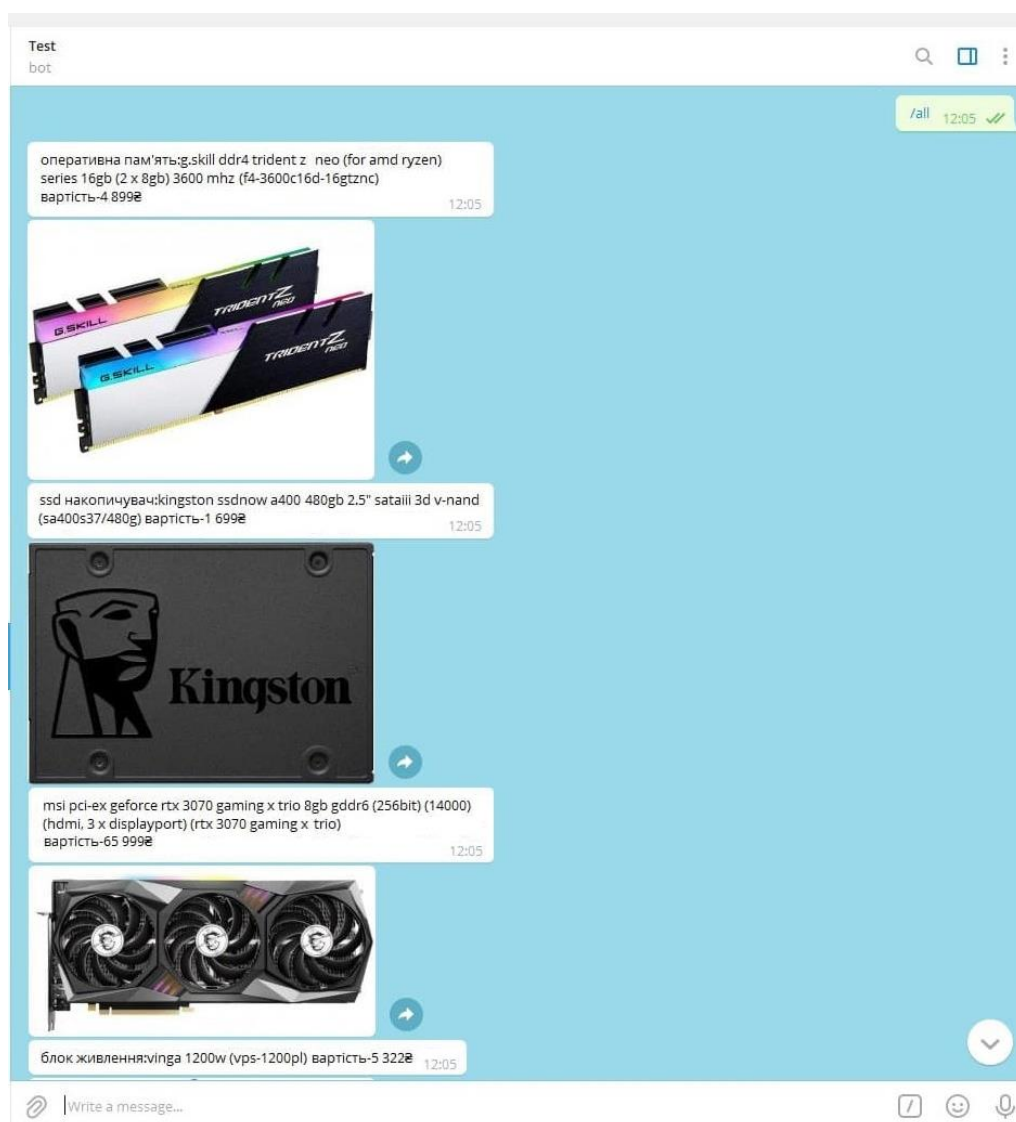


Рисунок 4.4 – Список товарів створений за допомогою команди /all

ВИСНОВКИ

В процесі виконання роботи було створено телеграм-бот крамниці. Було розв'язано такі завдання:

- а) аналіз предметної області;
- б) розробка технічного завдання;
- в) вибір середовища розробки;
- г) розробка схеми взаємодії модулів програми;
- д) конструювання телеграм-бота крамниці;
- е) тестування розробленого програмного забезпечення.

Були розроблені команди:

- / Add-команда для додавання картинки у телеграм-бот. Додавання картинки відбувається шляхом додавання адресу зображення до-буферу;
- / Check- команда для перевірки буфером;
- / All- команда видає всі збережені зображення з назвою;
- / Save- команда зберігає зображення і назва з буфера.

Серед особливостей проекту можна виділити використання сучасних бібліотек: Telebot, Shutil, Re.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Статистика кількості користувачів в Телеграмі [Electronic resource]. / – Access mode: [tg://resolve?domain=durov&post=147](https://tg.resolve?domain=durov&post=147).
2. Hosaine Y. Embedded System based Bangla Intelligent Social Virtual Robot with Sentiment Analysis [Electronic resource] / Y. Hosaine, I. Hosaine, M. Banik – Access mode: https://www.researchgate.net/publication/331132851_Embedded_System_based_Bangla_Intelligent_Social_Virtual_Robot_with_Sentiment_Analysis.
3. Grimme C. Social Bots: Human-Like by Means of Human Control? [Electronic resource] / C. Grimme, M. Preuss, I. Clever, H. Trautmann .– Access mode: https://www.researchgate.net/publication/317887526_Social_Bots_Human-Like_by_Means_of_Human_Control.
4. Tariq Banday M. Qadri J. Official Study of Botnets and their threats to Internet Security [Electronic resource] / M. Tariq Banday, J. Qadri. – Access mode: https://www.researchgate.net/publication/227859109_Study_of_Botnets_and_their_threats_to_Internet_Security.
5. Dunham K., Melnick J. Malicious bots [Electronic resource] / K. Dunham, J. Melnick. – Access mode: <http://index-of.es/Hack/Malicious%20Bots.pdf>.
6. Gorva R., Guilbeault D. Unpacking the Social Media Bot: A Typology to Guide Research and Policy: Unpacking the Social Media Bot [Electronic resource] / R. Gorva, D. Guilbeault. – Access mode: https://www.researchgate.net/publication/326973601_Unpacking_the_Social_Media_Bot_A_Typology_to_Guide_Research_and_Policy_Unpacking_the_Social_Media_Bot.
7. Hasan W. A Survey of Current Research on CAPTCHA [Electronic resource] / W. Hasan. – Access mode: https://www.researchgate.net/publication/304995447_A_Survey_of_Current_Research_on_CAPTCHA.

8. Adam M., Wessel M. AI-based chatbots in customer service and their effects on user compliance [Electronic resource] / M. Adam, M. Wessel. – Access mode: <https://link.springer.com/article/10.1007/s12525-020-00414-7>.

8. Moon M. Facebook Messenger launches its first airline bot [Electronic resource] / M. Moon. – Access mode: <https://www.engadget.com/2016-03-30-facebook-messenger-first-airline-bot.html>.

10. Sumit R. Building Chatbots with Python Using Natural Language Processing and Machine Learning [Electronic resource] / R. Sumit. – Access mode: https://www.academia.edu/40419686/Building_Chatbots_with_Python_Using_Natural_Language_Processing_and_Machine_Learning_Sumit_Raj.

11. Khan R., Das A. Build Better Chatbots: A Complete Guide to Getting Started with Chatbots [Electronic resource] / R. Khan, A. Das. – Access mode: <https://www.pdfdrive.com/build-better-chatbots-a-complete-guide-to-getting-started-with-chatbots-e158321626.html>.

12. Telegram Bot API [Electronic resource]. – Access mode: <https://core.telegram.org/bots/api>.

13. Telegram APIs [Electronic resource]. – Access mode: <https://core.telegram.org/>.

14. PHP Manual [Electronic resource]. – Access mode: <https://www.php.net/manual/en/>.

15. JavaScript (JS) [Electronic resource]. – Access mode: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.

16. Woebot health [Electronic resource]. – Access mode: <https://woebothealth.com/>.

17. Types of Chatbots [Electronic resource]. – Access mode: <https://www.artificial-solutions.com/chatbots>.

18. Ruby [Electronic resource]. – Access mode: <https://www.ruby-lang.org/en/>.

19. Python [Electronic resource]. – Access mode: <https://docs.python.org/3/>.

20. PyCharm [Electronic resource]. – Access mode: <https://www.jetbrains.com/pycharm/learn/>.
21. JSON [Electronic resource]. – Access mode: <https://www.json.org/json-en.html>.
22. HTTP запити [Electronic resource]. – Access mode: <https://httpwg.org/specs/>.
23. REST API [Electronic resource]. – Access mode: <https://docs.github.com/en/rest>.

ДОДАТОК А
Технічне завдання

A.1 Підстави для розробки

Підставою для розробки є завдання на дипломний проект на тему «Розробка Телеграм бот для крамниці», затверджене наказом по НУ «Запорізька політехніка» № 103 від 30 березня 2021 року.

A.2 Призначення розробки

Програмне забезпечення повинно дозволяти: збільшити охоплення користувачів, і відповідно кількість продажів через інтернет.

A.3 Вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення повинно забезпечувати виконання наступних функцій :

- реалізація додавання назви товару;
- реалізація додавання зображення товару;
- реалізація перевірки буфера;
- реалізація збереження внесених даних;
- створення списку зображення з назвою товару.

A.3.2 Умови експлуатації

Програма може бути записана на оптичному, жорсткому або магнітному диску.

А.3.3 Вимоги до складу і параметрів технічних засобів

Вимоги до складу і параметрів технічних засобів:

- ОС Microsoft Windows 10;
- процесор Intel Celeron G1830 2800 МГц;
- жорсткий диск місткістю не менше 250 Гб;
- мінімум 2 ГБ обсягу оперативної пам'яті;
- мати доступ к Інтернету;
- засоби керування та відображення інформації (клавіатура, миша та монітор).

А.3.4 Вимоги до маркування та упакування

Програмне забезпечення не потребує додаткових операцій щодо маркування та упакування.

А.3.5 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання програми відповідають аналогічним вимогам носія, на якому вона записана.

А.3.6 Вимоги до програмної документації

Програма повинна поставлятися разом з наступними програмними документами:

- текст програми;
- опис експлуатації та тестування програми.

А.3.7 Вимоги до маркування та упакування

Програма може бути записана на оптичному, гнучкому або жорсткому диску.

У випадку розповсюдження програмного забезпечення на відповідному носії має бути вказано маркування «Телеграм бот для крамниці».

А.3.8 Стадії етапи розробки

Стадії і етапи розробки програми:

- аналіз предметної області;
- проектування;
- реалізація;
- тестування.

А.3.9 Порядок контролю та приймання

Порядок контролю та приймання відповідає затвердженому порядку контролю та приймання дипломних кваліфікаційних робіт студентів денної та заочної форми навчання освітнього ступеня «бакалавр» НУ «Запорізька політехніка».

ДОДАТОК Б
ТЕКСТ ПРОГРАМИ

Б.1 Файл start.py:

```
import os
counter = 0
work = True
while work:
    print("\nСервер запустився")
    os.system("python wizzards.py")
    print("\nСервер лежит\n")
    counter += 1

print(counter)
```

Б.2 Файл wizzards.py:

```
import telebot
import json
import shutil
import re

TOKEN = "1711235743:AAEJHcwIXDp7tYa2k9shq7fKkNrMytinM1s"

bot = telebot.TeleBot(TOKEN)
bufCard = {}
bufferName = 'buf.jpg'
fileCards = 'cards.json'

@bot.message_handler(regexp="/add")
def add(message):
    if 1==0:
        return bot.send_message(message.from_user.id,)
    sent = bot.send_message(message.chat.id, "Введіть назву карти")
    bot.register_next_step_handler(sent, addCard)

def addCard(message):
    bufCard['name'] = message.text.lower()
    done = bot.send_message(message.chat.id, "Добре, назву записано")
@bot.message_handler(regexp="/check")
def check(message):
    if 1 == 0:
        return bot.send_message(message.from_user.id, "Вам заборонено")
    try:
        bot.send_message(message.chat.id, "Добре, назву записано: " +
str(bufCard['name']))
    except:
        bot.send_message(message.chat.id, "Введіть назву карти через функцію
/add")
    try:
        img = open(bufferName, "rb")
        bot.send_photo(message.chat.id, img)
```

```

        img.close()
    except:
        bot.send_message(message.chat.id, "")

@bot.message_handler(regexp="/save")
def save(message):
    if 1 == 0:
        return bot.send_message(message.from_user.id, " Вам заборонено")
    try:
        path = 'img/' + bufCard['name'] + '.jpg'
        shutil.move(bufferName, path)
        bufCard['path'] = path
    except:
        bot.send_message(message.chat.id, " Не пішло, краще заново ")
    a = []
    if bufCard['name'] is None:
        return bot.send_message(message.chat.id, "Введіть назву карти /add и
повторите")
    try:
        workFile = open(fileCards, mode="r")
        a = json.load(workFile)
        workFile.seek(0)
        workFile.close()
        workFile = open(fileCards, mode="w")
    except:
        workFile = open(fileCards, mode="w")
    a.append(bufCard)
    json.dump(a, workFile)
    workFile.close()
    bufCard['name'] = None
    bufCard['path'] = None
    bot.send_message(message.from_user.id, " Дані оновлені ")

@bot.message_handler(regexp="/all")
def all(message):
    try:
        workFile = open(fileCards, mode="r")
        a = json.load(workFile)
        count = 0
        for i in a:
            count += 1
        for i in range(count):
            b = a[i]
            bot.send_message(message.from_user.id, b['name'])
            img = open(b['path'], mode="rb")
            bot.send_photo(message.from_user.id, img)
        colvo = "Зареєстровано " + str(count)
        bot.send_message(message.from_user.id, colvo)
    except:
        bot.send_message(message.from_user.id, " Мабуть не занесені ")

@bot.message_handler(content_types="text")
def search(message):
    txt = message.text.lower()
    try:
        workFile = open(fileCards, mode="r")

```

```

a = json.load(workFile)
count = 0
for i in a:
    count += 1
for i in range(count):
    b = a[i]
    if re.match(str("^[\\D]*(" + txt + ")+"), b['name'].lower()):
        bot.send_message(message.from_user.id, b['name'])
        img = open(b['path'], mode="rb")
        bot.send_photo(message.from_user.id, img)
except:
    bot.send_message(message.from_user.id, "Не знайдено")

@bot.message_handler(content_types="photo")
def phot(message):
    if 1 == 0:
        return bot.send_message(message.from_user.id, " Вам заборонено ")
    img = message.photo[0].file_id
    file = bot.get_file(img)
    down = bot.download_file(file.file_path)
    with open(bufferName, 'wb') as new_file:
        new_file.write(down)
    bot.send_message(message.chat.id, "Фото завантажено")

bot.polling( none_stop = True )

```

ДОДАТОК В
СЛАЙДИ ПРЕЗЕНТАЦІЇ

Національний університет «Запорізька Політехніка»

Кафедра програмних засобів
Розробка телеграм-бота для крамниці.

Виконав
ст.гр.КНТЗ-117
М.С. Абдуллаєв

Керівник
к.т.н., доцент
Т.В. Федорончак

Рисунок В.1 – Слайд № 1

Об'єкт, предмет та мета роботи

Об'єкт роботи – програмне забезпечення для магазинів.

Предмет роботи – телеграм-бот магазин, бот реалізований на Python.

Мета-роботи – створення телеграм-бота для крамниці.

2

Рисунок В.2 – Слайд № 2

Завдання до роботи

В процесі виконання роботи були поставлені такі задачі

- аналіз предметної області;
- розробка технічного завдання;
- вибір середовища розробки;
- розробка схеми взаємодії модулів програми;
- конструювання телеграм-бота крамниці;
- тестування розробленого програмного забезпечення.

3

Рисунок В.3 – Слайд № 3

Вибір середовища розробки

PyCharm це IDE що забезпечує:

- аналіз коду в ході написання;
- навігація по коду;
- рефакторинг;
- Інтеграція з системою управління версіями;

Важливою перевагою інтегрованого середовища розробки PyCharm є аналіз коду в ході написання. Що дозволяє писати менше коду руками але затарчує більше ресурсів комп'ютера.

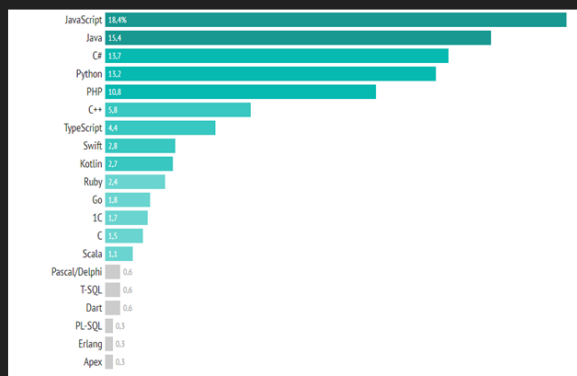


4

Рисунок В.4 – Слайд № 4

Вибір мови програмування

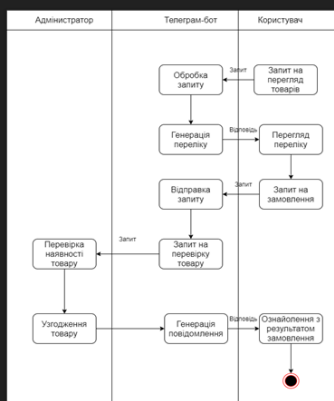
Найпопулярнішими мовами для розробки телеграм-бота є.



5

Рисунок В.5 – Слайд № 5

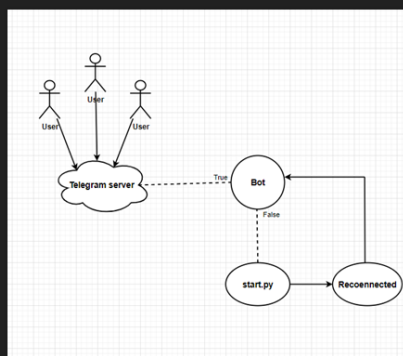
Алгоритм функціонування телеграм-бота



6

Рисунок В.6 – Слайд № 6

Робота файлу start.py



7

Рисунок В.7 – Слайд № 7

Модулі використані в програмі

Назва	Функціональні можливості
<u>Telebot</u>	Потрібен для запитів до <u>TelegramBotAPI</u> .
<u>Shutil</u>	Пропонує ряд <u>високорівневих</u> операцій над файлами і папками.
<u>Re</u>	Регулярні вирази, також звані <u>regex</u> , використовуються практично у всіх мовах програмування.

8

Рисунок В.8 – Слайд № 8

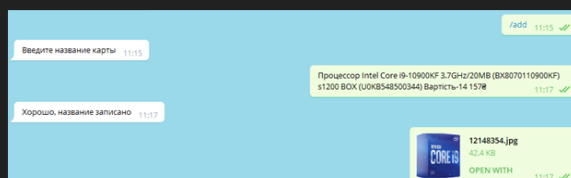
Головні функції програми

Назва функції	Опис
<u>Add</u>	Додавання картинки у телеграм-бот. Додавання картинки відбувається шляхом додавання адресу зображення до-буферу.
<u>AddCard</u>	Назва записується в нижньому регістрі до буферу, і виводиться повідомлення про успішний запис.
<u>Check</u>	Перевірка буфера
<u>Save</u>	Зберігає зображення і назва з буфера.
<u>All</u>	Вивдає всі збережені зображення з назвою.
<u>Phot</u>	Фото завантажується на сервер

9

Рисунок В.9 – Слайд № 9

Реалізація "Команди/add"



10

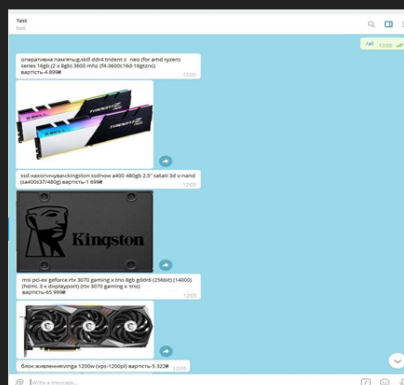
Рисунок В.10 – Слайд № 10

Реалізація "Команди/check"



Рисунок В.11 – Слайд № 11

Реалізація "Команди/all"



Висновки

В процесі виконання роботи було створено телеграм-бот крамниці. Було розв'язано такі завдання:

- а) аналіз предметної області;
- б) розробка технічного завдання;
- в) вибір середовища розробки;
- г) розробка схеми взаємодії модулів програми;
- д) конструювання телеграм-бота для крамниці;
- е) тестування розробленого програмного забезпечення;

Були розроблені команди:

/ Add-команда для додавання картинки у телеграм-бот. Додавання картинки відбувається шляхом додавання адресу зображення до-буферу;

/ Check- команда для перевірки буфером;

/ All- команда видає всі збережені зображення з назвою;

/ Save- команда зберігає зображення і назва з буфера.

Серед особливостей проекту можна виділити використання сучасних бібліотек: Telebot, Shutil, Re.

Рисунок В.13 – Слайд № 13