

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
Факультет кібербезпеки, комп'ютерної та програмної інженерії  
Кафедра комп'ютеризованих систем управління

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

Литвиненко О.Є.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2022 р.

# ДИПЛОМНИЙ ПРОЄКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ

“БАКАЛАВР”

**Тема: «Система автоматизованого обліку та фільтрації  
інформації про клієнтів у закладах харчування»**

Виконавець: \_\_\_\_\_ Ільчишин А.В.

Керівник: \_\_\_\_\_ д.т.н., Мислович М.В.

Нормоконтролер: \_\_\_\_\_ Тупота Є.В.

(підпис)

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютеризованих систем управління

Освітній ступінь: "Бакалавр"

Галузь знань, спеціальність, освітньо-професійна програма:

123 "Комп'ютерна інженерія", "Системне програмування"

ЗАТВЕРДЖУЮ

Завідувач кафедри

Литвиненко О. Є.

« \_\_\_\_\_ » \_\_\_\_\_ 2022 р.

**ЗАВДАННЯ**

**на виконання допломного проекту студента**

Гльчишина Андрія Вадимовича

(прізвище, ім'я, по батькові випускника в родовому відмінку)

**1. Тема роботи:** «Система автоматизованого обліку та фільтрації інформації про клієнтів у закладах харчування»

затверджена наказом ректора від "4" лютого 2022 року №135/ст.

**2. Термін виконання роботи:** з 17.05.2022 до 20.06.2022

**3. Вихідні дані до роботи:** системи управління проектами, технології проектування, мова програмування високого рівня *Python, PyCharm, Telegram, TelegramBotAPI, Microsoft Word*

**4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):**

- 1) сервіси надання послуг у сфері харчування;
- 2) технічні засоби для реалізації системи обліку;
- 3) програмна реалізація системи автоматизованого обліку та фільтрації інформації про клієнтів у закладах харчування.

**5. Перелік обов'язкового графічного матеріалу:**

- 1) функціональна схема боту barkrimberli;
- 2) робота боту barkimberli на стороні клієнтів(схема алгоритму);
- 3) робота боту barkimberli на стороні менеджерів(схема алгоритму);
- 4) приклад роботи боту(видалення користувача з чорного списку).

## 6. Календарний план

№ п/п	Етапи виконання дипломного проєкту	Термін виконання етапів	Примітка
1	Ознайомлення з постановкою задачі на дипломний проєкт	17.05.2022 – 18.05.2022	
2	Вивчення спеціальної літератури і технічної документації	18.05.2022 – 20.05.2022	
3	Аналіз основних завдань та цілей проєктування	20.05.2022 – 21.05.2022	
5	Аналіз сучасних методів і алгоритмів реалізації програмних засобів	21.05.2022 – 24.05.2022	
7	Аналіз методів створення систем автоматизованого обліку	24.05.2022 – 29.05.2022	
7	Розробка і опис боту	29.05.2022	
8	Оформлення пояснювальної записки	29.05.2022 – 04.06.2022	
9	Підготувати графічний демонстраційний матеріал	04.06.2022 – 06.06.2022	

Дата видачі завдання: 17.05.2022 р.

Керівник дипломного проєкту \_\_\_\_\_ Мислович М.В.  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_ Ільчишин А.В.

(підпис випускника) (П.І.Б.)

## РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Система автоматизованого обліку та фільтрації інформації про клієнтів у закладах харчування»: 54 с., 44 рис., 10 літературних джерел.

ПРОГРАМНИЙ ПРОДУКТ, СИСТЕМА АВТОМАТИЗОВАНОГО ОБЛІКУ ТА ФІЛЬТРАЦІЇ, БОТ, БОТ В ТЕЛЕГРАМ, ЧАТ-БОТ, РОЗРОБКА БОТУ.

Об'єктом дослідження даного дипломного проекту є процеси автоматизації підтримки клієнтів у закладах харчування.

Предметом дослідження є технології створення телеграм-ботів на мові програмування високого рівня.

Метою даного дипломного проекту є розробка системи автоматизованого обліку та фільтрації інформації про клієнтів у закладах харчування для швидкої обробки запитів користувачів.

Методи дослідження – технології систем управління проектів, порівняльний аналіз, методи об'єктно-орієнтованого програмування, аналіз основних алгоритмів і методів створення ботів та їх цілей, розробка коду.

Здійснено огляд теорії структури роботи бота; здійснено порівняльний аналіз функціональних можливостей вже створених додатків; проаналізовано структуру створення Telegram-ботів; реалізовано інтерактивний додаток за допомогою інтерфейсу програмування застосунків TelegramBotAPI та утиліти Telegram BotFather.

Результатом проекту є розроблений програмний продукт «Система автоматизованого обліку та фільтрації інформації про клієнтів у закладах харчування» для швидкої обробки запитів користувачів в застосунку Telegram.

Прогнозні припущення про розвиток об'єкту та предмету дослідження – застосування в якості підтримки клієнтів компанії(закладу харчування) в мобільному додатку.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ.....	7
ВСТУП .....	8
РОЗДІЛ 1. СЕРВІСИ НАДАННЯ ПОСЛУГ У СФЕРІ ХАРЧУВАННЯ .....	11
1.1. Призначення та класифікація Telegram-ботів .....	11
1.2. Аналіз застосування існуючих Telegram-ботів у закладах харчування ...	21
1.3. Висновки до розділу .....	25
РОЗДІЛ 2. ТЕХНІЧНІ ЗАСОБИ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ ОБЛІКУ .....	26
2.1. Вибір мови програмування для Telegram-бота .....	26
2.2. Середовище розробки PyCharm.....	28
2.3. BotFather та Telegram BOT Api.....	30
2.4. Висновки до розділу .....	31
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ОБЛІКУ ТА ФІЛЬТРАЦІЇ ІНФОРМАЦІЇ ПРО КЛІЄНТІВ У ЗАКЛАДАХ ХАРЧУВАННЯ....	34
3.1. Реєстрація Telegram-бота .....	34
3.2. Розробка коду і приклад роботи Telegram-бота.....	36
3.3. Висновки до розділу .....	49
ВИСНОВКИ.....	51
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

*HTML – Hyper Text Markup Language*

*API – Application Programming Interface*

*OS – Operating System*

*IOS – iPhone Operating System*

*MacOS – Macintosh Operating System*

*HTTP – HyperText Transfer Protocol*

## ВСТУП

Актуальність теми. Сьогодні, в роки розвитку всіх можливих інтернет-технологій, а також розвитку мобільних програм з різними можливостями, постає проблема з необхідністю завантажити на свій смартфон десятки додатків, які допомагають нам у різних сферах життя. Вони значно полегшують вирішення різноманітних задач користувачам, але оскільки додаток має лише одну функцію і вирішує конкретну проблему в одній області, користувачам доводиться завантажувати велику кількість багатоцільових програм.

Це може призвести до проблем із заповненням пам'яті пристрою, а також до витрат часу, залишеного на пошук потрібної програми та постійного перемикання з одного застосунку на інший .

Тому багатофункціональні програми стають все більш популярними, але в той же час можуть забезпечувати багатофункціональність, що виражається в широкому спектрі послуг, функцій, а також відповідає різноманітним потребам користувачів. Такі програми можуть включати, наприклад, функціональність десяти або більше незалежних програм.

По-перше, одна з цих багатофункціональних програм займає в кілька разів менше пам'яті, ніж загальна пам'ять, яку займають ці десять окремих незалежних програм.

Крім того, слід зазначити, що часу, необхідного для виконання того ж набору вимог у багатофункціональному додатку, витрачається значно менше, ніж при використанні окремих програм. Це пов'язано з втратою часу на перемикання між додатками, а також можливим переходом постійної авторизації між програмами.

Ця дипломна робота присвячена розробці надбудови у вигляді боту для багатофункціонального додатка Telegram. Такі програми, як боти, надають цій програмі універсальність, оскільки тепер можна створювати різноманітні додатки для будь-якої мети.

Метою виготовленого в даному випадку боту є застосунок, котрий допоможе клієнтам закладу харчування швидко вирішити свої питання і котрий також допоможе менеджерам цього закладу швидше опрацьовувати вхідні запити. Бот



буде приймати повідомлення від користувача і пересилати в чат менеджерів, потім вільний менеджер отримає запитання і зможе швидко відповісти на нього.

Відповідно, раціональний підхід до обраної теми має сприяти підвищенню інтересу до розробки більш різнофункціональних ботів, що розширять спектр послуг, які вже доступні в додатку Telegram.

Telegram є безкоштовною й доступною для всіх користувачів програмою. Крім того, він дозволяє створювати різноманітні боти та канали без додаткових витрат. Ще однією перевагою є наявність інтерфейсів програмування, які називаються API. Ця функція значно спрощує розробку програм, збір даних та/або автоматизацію.

Telegram Bot - це багатофункціональний застосунок для вирішення проблем і надання послуг користувачам. Крім того, цей додаток безкоштовний. Бот — це такий тип програми, створений у сучасній програмі для обміну повідомленнями і файлами Telegram з використанням технології штучного інтелекту, і бот може виконувати багато різних функцій, таких як: надання даних про погоду; надання новин; планування; допомога в завантаженні медіафайлів з різних соціальних мереж; покупка товарів; замовлення їжі.

Дипломна робота спрямована на створення функціонального продукту «Система автоматизованого обліку та фільтрації інформації про клієнтів у закладах харчування» на основі програми Telegram-бот.

Об'єктом дослідження даного дипломного проекту є процеси автоматизації підтримки клієнтів у закладах харчування.

Предметом дослідження є технології створення телеграм-ботів на мові програмування високого рівня.

Метою даного дипломного проекту є розробка системи автоматизованого обліку та фільтрації інформації про клієнтів у закладах харчування для швидкої обробки запитів користувачів.

Методи дослідження – технології систем управління проектів, порівняльний аналіз, методи об'єктно-орієнтованого програмування, аналіз основних алгоритмів і методів створення ботів та їх цілей, розробка коду.

Технології управління проектами застосовуються для контролю над процесами, оптимізації, успішного досягнення цілей та завершення всієї етапів проектів. Порівняльний аналіз використовується для зіставлення двох і більше об'єктів дослідження Об'єктно-орієнтоване програмування розглядає програму як множину «об'єктів», що взаємодіють між собою і застосовується для повторного використання вже написаного одного разу коду. Аналіз основних алгоритмів і методів створення ботів та їх цілей потрібно для виявлення оптимального шлягу створення найбільш відповідного застосунку для вирішення завдання проекту. Розробка коду — це процес, спрямований на створення програмного забезпечення, використовуючи технології інформатики.

**Новизна дипломного проекту** полягає у застосуванні інтерфейсу програмування застосунків TelegramBotAPI та утиліти Telegram BotFather для створення вланого додаку Telegram-бот для автоматизованого обліку та фільтрації інформації про клієнтів у закладах харчування.

**Практичне значення** Бот призначений для того, щоб клієнти закладів харчування не витрачали зайвий час на пошук номеру телефона закладу харчування, на перехід на сайт закладу, а відразу в застосунку задали запитання, на яке потребують відповіді. Так клієнт оперативно може дізнатись адресу закладу, час роботи, забронювати столик, дізнатись які страви є в меню та ціни, адже завдяки боту до клієнта завжди оперативно під'єднається менеджер і допоможе в вирішенні різноманітних запитань.

**Особистий внесок випускника.** Всі результати, представлені у дипломній роботі, отримані випускником особисто. Особливої уваги заслуговує проектування логіки роботи додатку Telegram-бот.

**Практичні значення отриманих результатів** в дипломному проекті дають змогу використовувати отримані дані при створенні нових кращих додатків для месенджерів, можливість отримати знання структури сучасних застосунків для створення своїх та використання у відповідних сферах

**Прогнозні припущення** про розвиток об'єкту та предмету дослідження — застосування в якості платформи для навчання та еспериментів нових методів та

розвиток проекту до повноцінного для закриття маркетингових задач для закладів харчування.

# РОЗДІЛ 1

## СЕРВІСИ НАДАННЯ ПОСЛУГ У СФЕРІ ХАРЧУВАННЯ

### 1.1. Призначення та класифікація Telegram-ботів

#### 1.1.1. Сфери використання телеграм-ботів

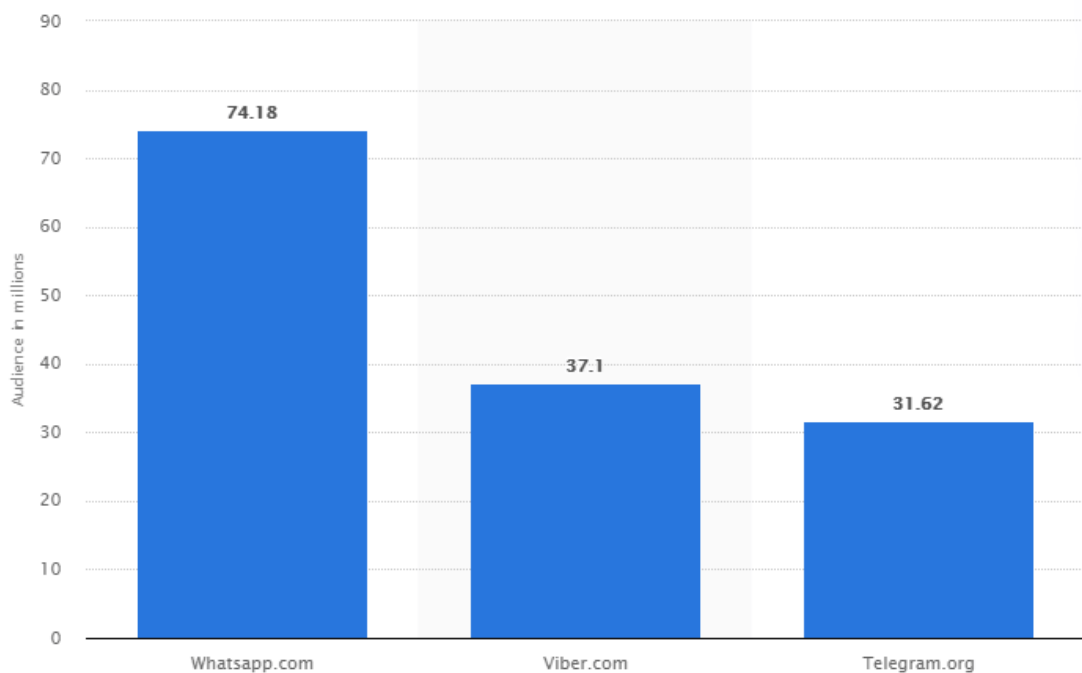
Telegram – це безкоштовний месенджер для спілкування та обміну інформацією між користувачами. Він дозволяє вести приватну чи групову бесіду, підписуватись на канали, створювати конференції, відправляти аудіо- та відеоповідомлення, та використовувати/створювати ботів. Також він підтримує обмін файлами, мультимедійними матеріалами різного формату.

12 січня 2021 року засновник програми Павло Дуров у своєму акаунті повідомив, що аудиторія активних користувачів Telegrama пододала позначку в 500 млн осіб:

Приблизно 1,5 млн осіб реєструються в Telegram щодня.

Telegram затребуваний у всьому світі, в тому числі і в Україні. За статистичними даними TGstat, на червень 2021 року кількість зареєстрованих українців склала 35 млн користувачів. За популярністю месенджер займає третє місце, поступаючись лише Whatsapp і Viber.

Кафедра КСУ				НАУ 22 18 81 000 ПЗ			
Виконав	Ільчишин А.В.			СЕРВІСИ НАДАННЯ ПОСЛУГ У СФЕРІ ХАРЧУВАННЯ	Літера	Аркуш	Аркушів
Керівник	Мислович М.В.					11	14
Консульт.					СП-435 123		
Нормокон.	Тупота С. В.						
Зав. каф.	Литвиненко О.С.						



*Рис. 1.1. Статистика месенджерів в Україні на січень 2022 року*

За допомогою Telegram можна:

- Спілкуватися онлайн з користувачами по всьому світу, також підтримуються аудіодзвінки з високою якістю звуку.
- Пересилати та отримувати файли різних форматів (фото, аудіо, відео та інші) об'ємом не більше 1,5 ГБ.
- Створювати групові чати із кількістю учасників до 200, у тому числі засекречені. Останні можливі завдяки шифруванню. Також можна додати таймер самознищення. Після його активації, через певний час, повідомлення будуть видалені з чату та пам'яті телефону та сервера. Причому, якщо буде зроблено фото екрана з повідомленням, відправник отримає повідомлення про це.
- Організувати спілкування у супер групах – до 200 тисяч людей. Завжди є, як мінімум, один адміністратор, який може модерувати повідомлення та застосовувати обмеження до учасників.
- Використовувати юзернейми – функція дозволяє визначити людину, не знаючи її номери телефону.
- Знаходити групи за інтересами та брати участь у їх обговореннях.

- Записувати та надсилати звукове повідомлення.
- Створювати канали для перегляду.
- Слухати музику – безкоштовно та навіть без інтернету. Для цього розроблений медіаплеєр зі зрозумілим інтерфейсом та великим набором функцій.
- Використовувати хмарне сховище даних від сервісу. Будь-які файли (обсяг та кількість не обмежені) за допомогою вкладки «Збережене» та пересилання самому собі потрапляють у віддалений сервер.
- Пересилати, самостійно створювати, безкоштовно (на відміну від інших месенджерів) завантажувати стікери.
- Надсилати знайдені у пошукових системах фото без збереження в пам'яті пристрою.
- Використовувати автоматизованих помічників – ботів.

#### Переваги Telegram:

- Висока швидкість роботи – незважаючи на розташування серверів у різних частинах планети, доставка повідомлення займе частки секунди.
- Зрозумілий класичний інтерфейс програми, який можна оформити у різних стилях: вибрати із готових варіантів або застосувати власний абсолютно безкоштовно.
- Автоматичне додавання списку користувачів з телефонної книги.
- Простий початок роботи. Для реєстрації потрібний лише номер телефону.
- Всі функції та можливості безкоштовні.
- Інноваційність – сервіс постійно оновлюється. Його можливості розширюються як з урахуванням пропозицій конкурентів, і власними розробками, які не мають аналогів.
- Високий рівень безпеки та захисту від злому облікового запису та листування.
- Сумісність з усіма основними операційними системами: IOS, Android OS, Windows OS, Linux OS, MacOS.

#### Недоліки:

- Комунікація зі службою технічної підтримки ведеться лише англійською мовою.
- Винятковий захист листування та відмова надавати владі та правоохоронним органам історію діалогів, спричиняє те, що через цей комунікатор відбувається спілкування терористичних угруповань, збут наркотиків та інші дії, заборонені законодавчо. Це породжує невдоволення деяких людей та звинувачення на адресу компанії у співучасті з криміналом.
- Неможливість проконтролювати та зупинити розсилку незаконної інформації призводить до блокування роботи програми в деяких країнах. Так, вводилися обмеження в Китаї на час урядових страйків та мітингів, та в Ірані за відмову у сприянні стеженню.

### **1.1.2. Функції Telegram-бота**

Telegram – це не просто засіб спілкування з іншими користувачами, обміну інформацією і файлами, але і чудовий інструмент, що дозволяє виконувати багато корисних завдань.

Також через месенджер можна дізнаватися новини, бути в курсі валют, перекладати тексти, грати в ігри, і ще багато всього. А виконувати всі ці дії допомагають спеціальні роботи – Telegram-боти.

Назва «бот» походить від скороченого слова «робот», чим він і є. Telegram-бот – це спеціальний аккаунт, створений в автоматичному режимі, який дозволяє користувачам здійснювати різні дії через сам месенджер.

Роботи виконують дії з текстовим командам користувача, за принципом «питання-відповідь» після натиснення кнопки «Старт».

Робота легко визначити, в його назві завжди є слово «bot», а ще він не може сам перший почати бесіду і не має статусу «онлайн» або «не в мережі» – ви завжди будете бачити підпис «bot».

Сьогодні роботи стали дуже популярні, вони допомагають користувачам виконувати типові рутинні дії в автоматичному режимі, значно полегшуючи їм життя. Для власників самих каналів боти стали незамінними помічниками у роботі.

Telegram-боти мають безліч очевидних плюсів:

- Доступні 24/7;
- Моментальна відповідь користувачу;
- Зручність у користуванні, спілкування за принципом «питання-відповідь» і текстові завдання під силу давати навіть зовсім недосвідченому користувачеві месенджера;
- Не вимагають установки додаткових програм, програм і т.п. Все спілкування з ботом ведеться безпосередньо через месенджер;
- Безпека особистих даних – боти працюють виключно за заданими командами.
- Необмежені можливості – віртуального помічника можна запрограмувати на відправку новин, розповідь анекдотів, нагадування важливої інформації, пошук закладів, замовлення квитків і т.д.

Боти виконують безліч функцій, і ось лише невелика їх перелік:

- Розваги. Боти можуть надсилати смішні меми, картинки, анекдоти, допомагають вибрати фільм, знайти пісню за голосовим повідомленням і не тільки.
- Пошук і обмін файлів. Бот допомагає надсилати та зберігати файли з різних джерел, знаходити Торренти, електронні книги тощо.
- Новини, важлива інформація. Бот освітить новини, погоду, курси валют і т.д.
- Інструменти. Робот допомагає перекладати тексти, нагадувати про важливі події і т.д.



- Інтеграція з іншими сервісами. Робот може відправляти повідомлення, управляти розумним будинком і т.д.
- Пошук місць. Telegram-бот допомагає шукати готелі, кінотеатри, ресторани та інші заклади.
- Транзакції. Робот дозволяє бронювати квитки, робити замовлення, викликати таксі та ін.

Майже кожен власник бізнесу – від невеликої піцерії до великої ІТ-компанії – хотів би мати власний Telegram-бот, який допоможе виконувати однотипні дії в автоматичному режимі.

Бізнес у Telegram отримав додаткові можливості для спілкування зі споживачами.

Для просування застосовуються такі методи:

- Боти для підтримки клієнтів з питань, що виникають.
- Канали для бізнесу – інформування про новинки, акції, ролики про застосування товарів. Можна стати ближче до клієнтів, розповівши про себе, свою роботу та проекти, канал цілком підійде для цих цілей. Цікавий зміст збільшує популярність, спонукає людей рекомендувати його перегляду іншим.
- Створення та розповсюдження стікерів для просування бренду.
- Спілкування всередині компанії – постановка завдань, дискусії, звіти, плани. Застосовується як на рівні всієї фірми, так і по відділах. Бонус у вигляді секретності та неможливості перехопити важливу інформацію конкурентами.
- Документообіг – зберігання та доступ до потрібних документів. Його можна швидко та зручно організувати за допомогою хмарного зберігання.

- Онлайн-каталог - за допомогою бота можна вибрати товар прямо в додатку.
- Прийом платежів.

**Telegram-боти для інтернет-магазинів.** За допомогою ботів користувачам легко робити замовлення, оплачувати їх, дізнаватися статус свого замовлення, відстежувати доставку і т.д. Роботи допоможуть інформувати покупців про знижки, про спеціальні бонусні програми і навіть проводити консультації.

**Telegram-боти для оффлайн-магазинів і служб доставки.** Багато служби доставки, супермаркети і офлайнові магазини вже використовують в роботі Telegram-бота. Так, наприклад, Вкусвилл з допомогою бота допомагає клієнтам відслідковувати залишок товарів, дивитися чеки, міняти улюблений продукт і т. д.

**Telegram-боти для заходів.** Telegram-боти стали все частіше використовуватися на заходах. Бот допомагає слухачам отримувати інформацію про виступи і спікерів, передавати питання виступаючим, сповіщати про різних заходах, робити рекламу та ін.

**Telegram-боти для служб таксі та інших послуг.** Бот таксі в телеграм підійде клієнтам, які не люблять завантажувати програми або погано володіють сучасними технологіями. Користувачі можуть замовити таксі, написавши пару повідомлень у популярному месенджері.

Бот як диспетчер таксі не потребує перерв на обід та відпочинок, може приймати необмежену кількість замовлень за хвилину, йому не потрібно платити зарплату.

**Telegram-боти для служб таксі і beauty-сфери.** Салони краси, приватні майстри, перукарні значно спростять життя з допомогою Telegram-бота. Він допоможе записатися на послугу, дізнатися ціни, проінформувати клієнта про акції, нагадати йому про запис, отримати відгук і т. д.

**Telegram-боти для фінансових організацій.** Відмінним віртуальним помічником стане Telegram-бот для банків і інших фінансових організацій. З допомогою нього користувачі зможуть шукати адреси офісів, залишати заявки на послуги(кредити тощо), дізнаватися курси валют, активувати карти і т.д.

**Telegram-боти для HoReCa.** Готелі, ресторани, кафе та інші організації зі сфери готельно-ресторанного бізнесу також спростять роботу з допомогою Telegram-бота. Бот допоможе забронювати номери і столики, вибрати певні послуги, збирати відгуки, нагадувати про важливі події.

### 1.1.3. Застосування Telegram-ботів у різних сферах

Instasavebot — цей Telegram-бот зберігає фотографії і відео з соціальної мережі Instagram за посиланням(рис. 1.2.).



**InstaSave**

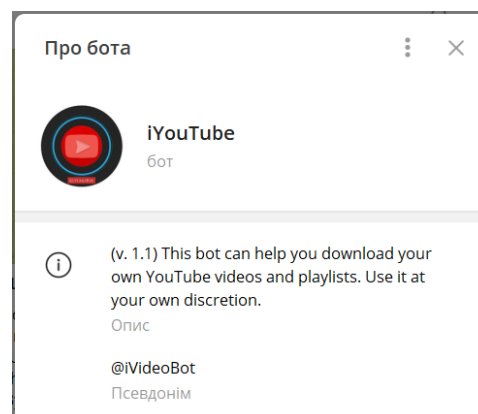
@Instasave\_bot

This bot helps you to download images/videos from Instagram.

SEND MESSAGE

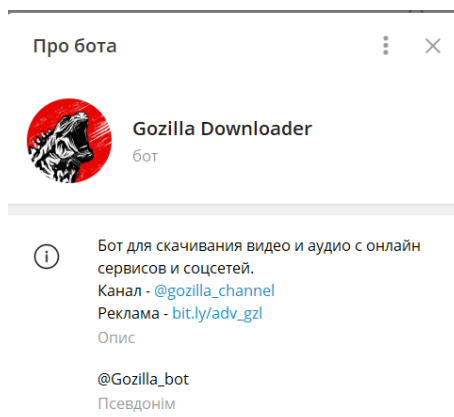
*Рис. 1.2. Вигляд Instasavebot*

Ivideobot — бот, який допомагає зберігати відеоролики з Youtube — після запиту він відправить вам посилання на скачування безпосередньо в чат Telegram(рис. 1.3.).



*Рис. 1.3. Вигляд Ivideobot*

Gozilla bot — ще один бот-помічник для завантаження відео, аудіо та фото з соціальних мереж та інших сервісів, в тому числі Вконтакте і Youtube(рис. 1.4).



*Рис. 1.4. Вигляд Gozilla bot*

Tap2bot — бот, який стане в нагоді кожному(рис. 1.5.). Допомогає користувачам шукати одяг, готелі, квитки, книги, фільми, різні файли.



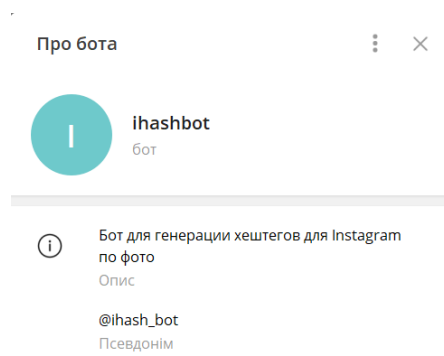
**Tap2Bot книги торренты  
фильмы**  
@Tap2Bot

Лучший бот для поиска книг, фильмов,  
билетов и многого другого!

SEND MESSAGE

*Рис. 1.5. Вигляд Tap2bot*

ihash\_bot — бот, який підбирає хештеги для соцмережі Instagram за картинкою(рис. 1.6.).



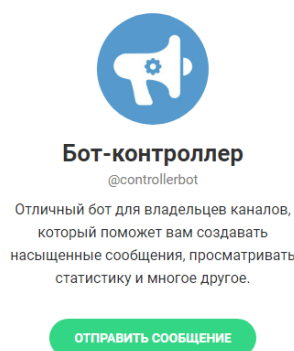
*Рис. 1.6. Вигляд ihash\_bot*

izibot — бот для скачування відео з соціальних мереж Instagram, Facebook, YouTube(рис. 1.7.).



*Рис. 1.7. Вигляд izibot*

controllerbot — бот для роботи в Telegram, який використовується для публічних каналів(рис. 1.8.). Допомагає створювати повідомлення і залишати пости з відкладеним публікацією, дивитися статистику.



*Рис. 1.8. Вигляд controllerbot*

## 1.2. Аналіз застосування існуючих Telegram-ботів у закладах харчування

Проаналізуємо застосування Telegram-ботів у закладах харчування міста Києва, зробимо висновки, які плюси та недоліки вони мають.

To-be – бот ресторану “То-Ве”(рис.1.9.). Його функціонал: кнопки клавіатури, або replyKeyboardMarkup-кнопки, після натискання бот виводить інформацію.

На мою думку, такий бот не є практичним для закладу, адже він виконує суто інформативну функцію і не може ніяк отримати запит від клієнта, або забронювати столик. Інформативну потребу можна було закрити звичайним закріпленим повідомленням в Telegram-каналі цього закладу, де було б відображено всю інформацію про соц мережі, адресу, графік роботи і т.д.

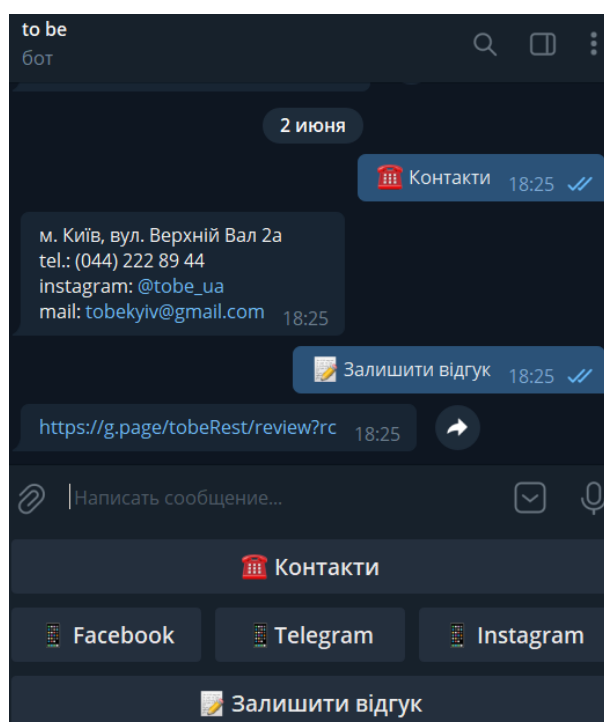


Рис. 1.9. Вигляд бота ресторану “То-ве”

skydon\_bot – бот, який відображає наявні акції в закладах харчування міста Києва(рис.1.10.). Його функціонал: строчні кнопки, або InlineKeyboardMarkup (кнопки прив’язані до повідомлень бота), після натискання, кнопки змінюються на інші, бот запропонує вибрати заклад та час для відвідування.

Цей бот добре вирішує свою задачу: пропонує клієнту заклади поруч біля його місцезоташування і відразу пропонує забронювати столик. Але не можна задати своє питання, отримати відповідь на питання, котрого немає в списку цих кнопок.

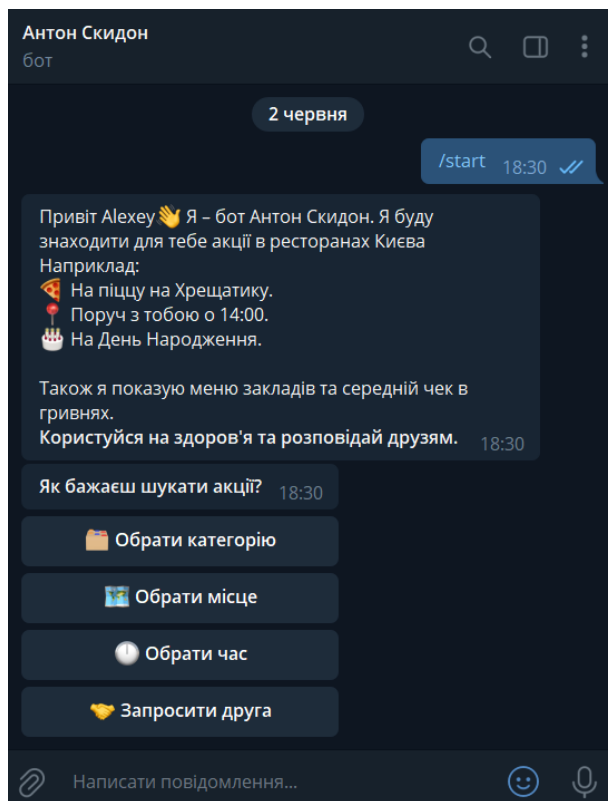


Рис. 1.10. Вигляд бота skydon\_bot

kyivbeer\_bot – бот закладу харчування «КПБ» міста Києва(рис.1.11.).

Його функціонал: бот під'єднує в чат менеджера, і при написанні повідомлення, тобі відповідає людина. Бот добре вирішує задачу, необмежений в функціоналі відносно запитів, тобто клієнт точно отримає відповідь на своє запитання, забронює столик.

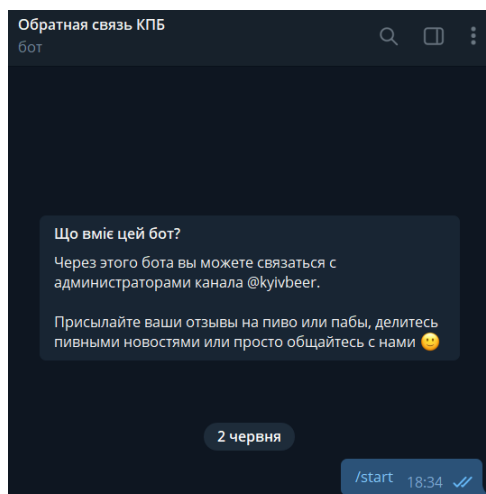


Рис. 1.11. Вигляд бота kyivbeer\_bot

Punkraft Beer Chanel Bot – бот бару “Punkraft”(рис.1.12.). Його функціонал: кнопки клавіатури, або replyKeyboardMarkup-кнопки, після натискання бот виводить інформацію.

Цей бот вже краще від бота ресторану “То-бе” тим, що може не тільки вивести інформацію після натискання кнопок, та й дає можливість забронювати столик шляхом вибору дати та часу кнопками. Але бот не може отримати зворотній зв’язок від клієнта, або відповісти на запитання, якого в нього нема в наявності.

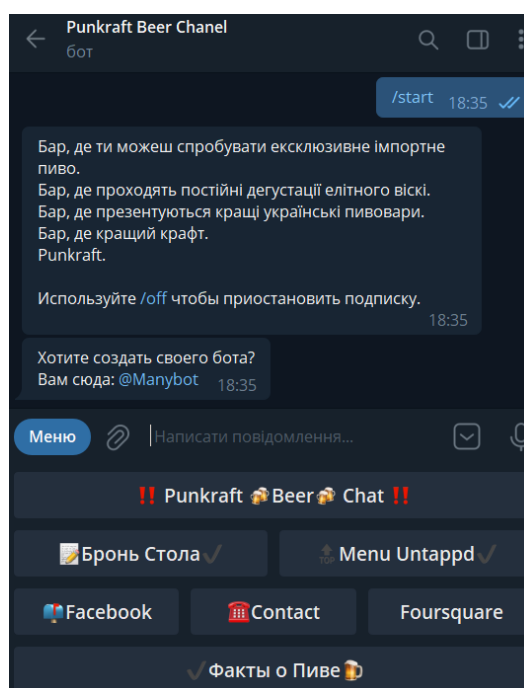


Рис. 1.12. Вигляд бота kyivbeer\_bot



Отже, оптимальним варіантом бота для закладу харчування є бот на прикладі kyivbeer\_bot. При написанні йому повідомлення, в чат під'єднується вільний менеджер і допомагає вирішити питання.

### **1.3. Висновки до розділу**

Отже, Telegram — один із найпопулярніших месенджерів для обміну текстовими, відео- та аудіоповідомленнями, та файлами необмеженої кількості вагою до 1.5 ГБ. Він є повністю безкоштовним, доступним на всіх більшвикористовуваних операційних системах, має розширені функції конфіденційності та шифрування, пропонує наскрізне шифрування за допомогою функції «секретні чати».

Також Telegram має можливість створення так званих «ботів», спеціальних аккаунтів, створених в автоматичному режимі, які дозволяють користувачам здійснювати різні дії через сам месенджер. Вони мають безліч плюсів у використанні:

- доступні 24/7;
- моментально відповідають користувачу;
- спілкування за принципом «питання-відповідь»;
- не вимагають установки додаткових програм, програм і т.п.;
- безпека особистих даних – боти працюють виключно за заданими командами.

Всі ці переваги зробили ботів затребуваними в різних сферах життя: бізнес, розваги, навчання, інформаційна сфера, як інструмент для завантажування медіафайлів з різних соціальних мереж, як інструмент для перекодування, або зміни формату файлів.

Було вибрано оптимальний варіант бота для закладу харчування, а саме бот, котрий приєднує менеджера в чат для відповіді на запитання клієнта.

## РОЗДІЛ 2

### ТЕХНІЧНІ ЗАСОБИ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ ОБЛІКУ

#### 2.1. Вибір мови програмування для Telegram-бота

Основна мова для ботів — Python. Але писати бота можна будь-якою мовою програмування. Python — основна мова програмування тому, що у його екосистемі багато відкритих бібліотек для ботів, причому бібліотек на основі штучного інтелекту. По-друге, Python — це відносна легкість у вивченні.

Ієрархія мов, якими пишуть ботів, виглядає так:

- Python;
- PHP;
- Node.js;
- Go;
- C#;
- Ruby;
- Java;
- Rust;
- C++;
- JavaScript та ін.

Кафедра КСУ				НАУ 22 18 81 000 ПЗ			
Виконав	Ільчишин А.В.			ТЕХНІЧНІ ЗАСОБИ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ ОБЛІКУ	Літера	Аркуш	Аркушів
Керівник	Мислович М.В.					25	8
Консульт.					СП-435 123		
Нормокон.	Тупота С. В.						
Зав. каф.	Литвиненко О.С.						

В інших мовах можна зустріти: Crystal, Delphi, Kotlin, Lua, Scala, Perl, TCL, Elixir, Haskell, Red, Swift, Microsoft Bot Framework та ін. Отже, бота можна написати будь-якою з популярних мов сучасності.

Сфера застосування робота. Якщо чат в месенджері, тоді Python або Java. Якщо для браузерної гри, то краще JavaScript. Якщо для комп'ютерної гри, тоді краще C# або C++.

Python – це високорівнева об'єктно-орієнтована та структурна мова програмування загального призначення, яка відноситься до категорії інтерпретованих мов і не потребує компіляції. Це є скриптова мова і відрізняється високим ступенем універсальності. Завдяки цьому, оптимально підходить для безлічі платформ та завдань, від серверних ОС до мобільних додатків під iOS/Android.

Переваги та недоліки:

- Python — мова програмування, що інтерпретується. З одного боку, це дозволяє значно спростити налагодження програм, з іншого — зумовлює порівняно низьку швидкість виконання.
- Динамічна типізація. У Python не треба заздалегідь оголошувати тип змінної, що дуже зручно під час розробки.
- Гарна підтримка модульності. Можна легко написати свій модуль та використовувати його в інших програмах.
- Вбудована підтримка Unicode у рядках. У Python необов'язково писати все англійською мовою, у програмах цілком може використовуватися інша мова.
- Підтримка об'єктно-орієнтованого програмування. При цьому його реалізація в python є однією з найзрозуміліших.
- Відсутність витоків пам'яті.
- Інтеграція з C/C++, якщо можливостей Python недостатньо.

- Зрозумілий та лаконічний синтаксис, що сприяє чіткому відображенню коду. Зручна система функцій дозволяє при грамотному підході створювати код, у якому легко розібратися іншій людині у разі потреби.
- Величезна кількість модулів, які входять до стандартного постачання Python 3, так і сторонніх. У деяких випадках для написання програми достатньо лише знайти відповідні модулі та правильно їх скомбінувати, що підвищує продуктивність програміста до десяти разів.
- Кросплатформність. Програма, написана на Python, функціонуватиме однаково незалежно від того, в якій операційній системі вона запущена. Відмінності виникають лише в окремих випадках, і їх легко заздалегідь передбачити завдяки наявності детальної документації.

Отже, для програмування Telegram-бота обрано мову програмування Python.

## 2.2. Середовище розробки PyCharm

Середовище розробки PyCharm(рис.2.1.) потрібно для створення якісних програм на мові програмування Python, для редагування скриптів на Javascript та Coffeescript, а також файлів розмітки та стилів web-документів.



*Рис. 2.1. Середовище розробки PyCharm*

PyCharm має пошук та навігацію за системою класів та методів мови. Це прискорює написання коду. Наявне автодоповнення та шаблони програмування,

можна імпортувати код з інших джерел. Також здійснюються перевірки сумісності версій інтерпретатора. Автоматичне форматування та підсвічування коду допомагають уявити код структурніше та логічніше, це полегшує сприйняття коду, його аналіз та виявлення синтаксичних помилок.

Для налагодження та тестування використовується інтегрований графічний налагодчик. Також в середовищі є зручний командний рядок. Профайлер Python допоможе визначити неоптимальні фрагменти коду, які працюють необґрунтовано довго, та оптимізувати їх. Надається можливість використання віддалених інтерпретаторів. Інтегровані засоби для створення та конфігурування віртуального середовища розробки Vagrant, а також розробки в середовищах із підтримкою контейнеризації Docker.

PyCharm підтримує unit-тестування, можна перевірити правильність роботи окремих ділянок коду, модулів та процедур. Це прискорює часто довгий і складний процес локалізації помилок програмування в тексті коду програми, адже він може складатися з багатьох сотень і навіть тисяч рядків.

При формуванні звернень до баз даних PyCharm пропонує зручні інструменти для формування SQL-запитів до Oracle, SQL Server, PostgreSQL, MySQL та інших СУБД. Можна писати і редагувати SQL-код і переглядати дані, що запитуються з його допомогою.

Важливо, що PyCharm підтримує багато популярних фреймворків. Так, для програмування на Python дуже корисна інтеграція з фреймворками Django і Flask.

Рефакторинг з PyCharm дозволяє швидко та безпечно оновлювати код та виконувати зміни у рамках всього проекту.

PyCharm працює на всіх популярних операційних системах Windows, MacOS та Linux.

Середовище розробки PyCharm підтримує кастомізацію: можна змінити інтерфейс користувача, використовувати приємні для сприйняття колірні схеми і підсвічування синтаксису, налаштувати сполучення клавіш для активування команд. Розроблено плагіни для підтримки зовнішніх систем управління версіями, баг-трекерами та оновленнями, інтеграції з фреймворками та різноманітними

інструментами. Бібліотека плагінів постійно поповнюється та суттєво розширює можливості PyCharm.

PyCharm розробляється та підтримується компанією JetBrains.

### 2.3. BotFather та Telegram BOT Api

У месенджері Telegram є бот BotFather(рис.2.2). - помічник, який допомагає користувачам створити свого власного бота та управляти ботами.

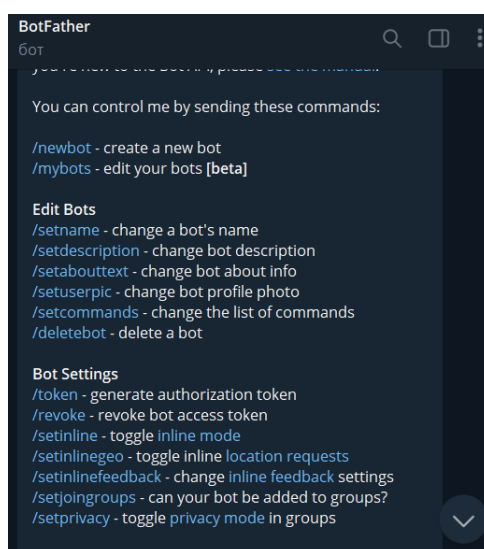


Рис. 2.2. Інтерфейс бота BotFather

У BotFather є 2 основні команди:

- */newbot* дозволяє створити новий бот.
- */mybots* запускає редактор налаштувань ваших роботів.

6 команд для редагування:

- */setname* для зміни назви бота.
- */setdescription* для зміни опису бота. Короткий опис бота. Дозволяється не більше 512 символів. Цей текст буде видно користувачеві під час ініціалізації діалогу.
- */setabouttext* — запускає редактор інформації "про бот". Інформація про робота відображається в його профілі, максимум 120 символів.

- */setuserpic* — відповідає за запуск редактора фотографії профілю бота. Розмір фотографії профілю повинен бути не меншим ніж 200×200 пікселів. Також бажано завантажувати квадратні зображення.
- */setcommands* – запускає редактор списку команд бота. Команди потрібні для управління та взаємодії з ботом. Кожна команда починається з косої межі "/", можна використовувати латинські літери, цифри, а також підкреслення. Але не більше ніж 32 символи. Також необхідно вказати опис команди. Цей список команд можна побачити, написавши косу межу «/» у діалозі з ботом.
- */deletebot* - команда для видалення бота

7 команд для налаштування бота:

- */token* — команда для генерації токена.
- */revoke* — команда зміни токена.
- */setinline* — команда увімкнення inline mode.
- */setinlinegeo* — увімкнення розташування.
- */setinlinefeedback* — відкриває налаштування зворотного зв'язку.
- */setjoingroups* — дозволяє увімкнути та вимкнути можливість додавання ботів у групи.
- */setprivacy* — команда для активації режиму конфіденційності у групах.

Можна вказати, які повідомлення бот може отримувати в групах.

Для створення бота потрібно ввести у чат із BotFather команду */newbot*. Далі ввести назву. Підтримується кирилиця та латиниця, наприклад: Студент bot. Ім'я відображатиметься в заголовку та в інформації про бота.

Після потрібно буде задати адресу бота на латиниці, підтримується до 32 символів, причому вона повинна закінчуватися на слово bot. Наприклад, *naubot* або *studentbot*.

Якщо все виконано вірно, то бот буде зареєстрований в Telegram і BotFather видасть токен.

Цей токен можна використовувати для авторизації бота та надсилання запитів до Bot API.

Якщо токен було втрачено, можна згенерувати його знову. Для цього потрібно написати боту BotFather команду /token, вибираємо потрібний бот, отримуємо новий токен.

За допомогою токена є можливість підключити Telegram бота з його програмною частиною у коді програми.

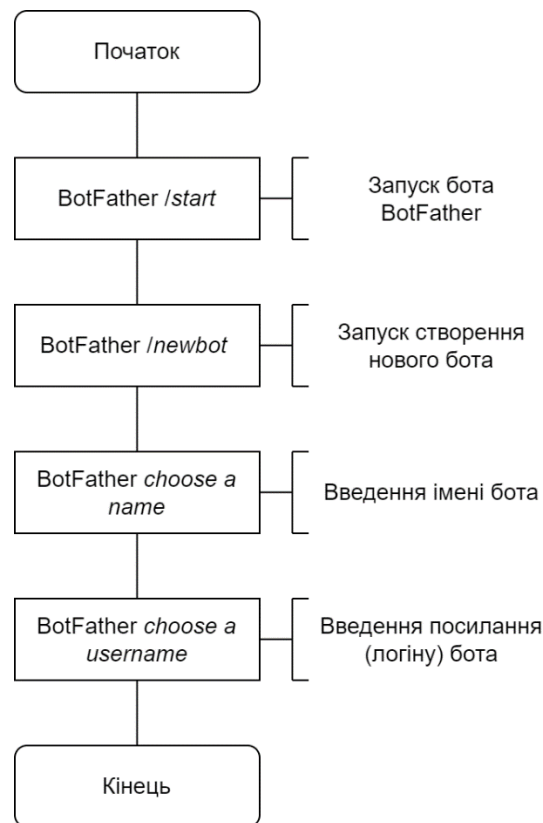


Рис. 2.3. Схема алгоритму створення бота за допомогою додатку BotFather

## 2.4. Висновки до розділу

Отже, в даному розділі було описано програмне забезпечення, що використовується для реалізації проекту.

Python — це високорівнева мова програмування, що інтерпретується. У його екосистемі багато відкритих бібліотек для ботів, причому бібліотек на основі штучного інтелекту, зрозумілий та лаконічний синтаксис, що сприяє чіткому відображенню коду. Python має багато модулів, що значно полегшує написання



програм. Python — кросплатформна мова, тому програми, функціонуватимуть однаково незалежно від того, в якій операційній системі вони запуснені.

PyCharm — це інтелектуальна Python IDE з повним набором засобів для ефективної розробки мовою Python. Вона підтримує кастомізацію і дозволяє налаштувати середовище для зручного користування. Підтримує багато фреємворків, плагінів, працює зі всіма бібліотеками.

BotFather — помічник, який допомагає користувачам створити свого власного бота та управляти ботами. Робота з ним проста і не вимагає специфічних навичок. За допомогою BotFather можна зареєструвати необмежену кількість ботів.

## РОЗДІЛ 3

### ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ОБЛІКУ ТА ФІЛЬТРАЦІЇ ІНФОРМАЦІЇ ПРО КЛІЄНТІВ У ЗАКЛАДАХ ХАРЧУВАННЯ

#### 3.1. Реєстрація Telegram-бота

Проходимо реєстрацію за допомогою бота BotFather(рис. 3.1.)



*Рис. 3.1. Вигляд BotFather*

Далі потрібно придумати ім'я та тег-посилання для бота закладу харчування. В мене назва BarKimberliBot, посилання @barkimberlibot. Також зробив опис бота і його фото-мініатюру.

Після перевірки доступності цього імені і посилання, бот було створено і я отримав хеш-токен для доступу до HTTP API(рис. 3.2.).

Кафедра КСУ				НАУ 22 18 81 000 ПЗ			
Виконав	Ільчишин А.В.			ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ОБЛІКУ ТА ФІЛЬТРАЦІЇ ІНФОРМАЦІЇ ПРО КЛІЄНТІВ У ЗАКЛАДАХ ХАРЧУВАННЯ	Літера	Аркуш	Аркушів
Керівник	Мислович М.В.					33	18
Консульт.					СП-435 123		
Нормокон.	Тупота Є. В.						
Зав. каф.	Литвиненко О.Є.						

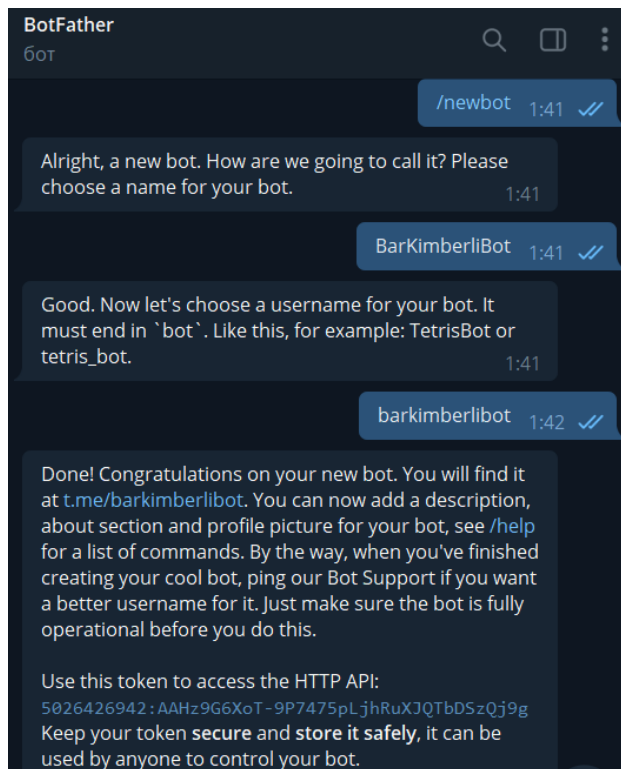


Рис. 3.2. Процес створення бота за допомогою BotFather

Після всіх виконаних кроків отримав такого бота(рис. 3.3.).

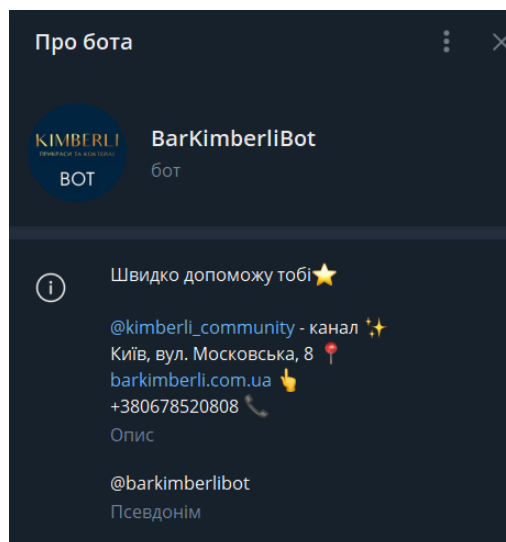


Рис. 3.3. Вигляд BarKimberliBot

Але він ще не має ніякого функціоналу, бот на даному етапі ніяк не оброблює вхідні повідомлення. Щоб бот працював і виконував свої функції, потрібно запрограмувати його.

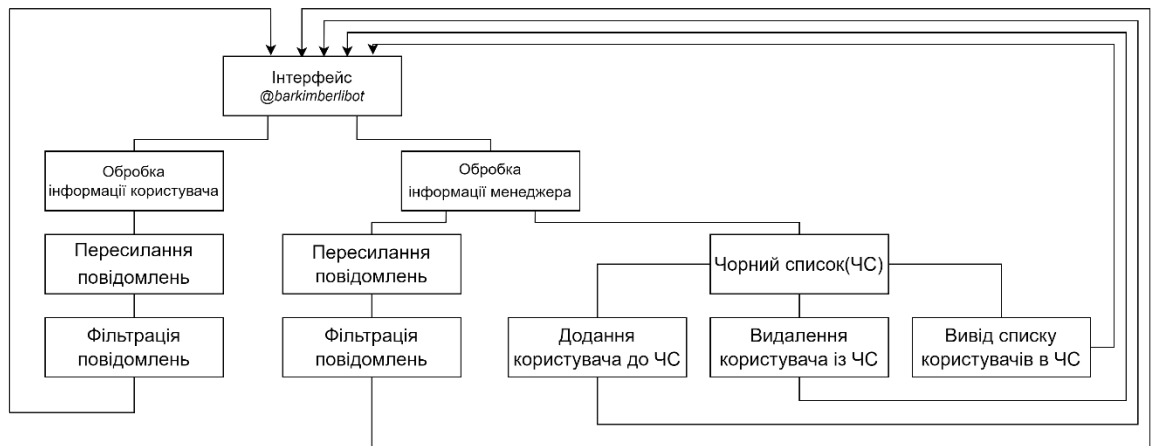


Рис. 3.4. Функціональна схема боту bakrimberli

### 3.2. Розробка коду і приклад роботи Telegram-бота

Функціонал бота:

- Привітальне повідомлення користувачу(клієнту), яке буде видно 1 раз після запуску, що описує функціонал бота.
- Кнопка «/help», яку в любий момент може натиснути користувач і ще раз ознайомитись з функціоналом бота.
- Пересилання повідомлення від клієнта в чат менеджерів.
- Фільтрація невідримуваних для пересилання типів повідомлень, а саме: опитування, стікери, геомітки і т.д.
- Пересилання повідомлення від менеджерів в чат клієнта шляхом відповіді на повідомлення від бота.
- Можливість менеджера подивитись ім'я, id, посилання на користувача, котрий звернувся.
- Можливість менеджера додати користувача в чорний список із сповіщенням для клієнта про це. Тобто, повідомлення від клієнта, доданого в цей список, не будуть пересилатись в чат менеджерів і клієнт буде отримувати сповіщення, що він не може писати в чат.
- Можливість менеджера додати користувача приховано в чорний список без сповіщення клієнта про це. Тобто, повідомлення від клієнта,

доданого в цей список, не будуть пересилатись в чат менеджерів і клієнт НЕ буде отримувати сповіщення, що він не може писати в чат.

- Перегляд користувачів в чорному спуску.
- Можливість видалення користувачів із чорного списку.

Спочатку імпортуємо бібліотеки(рис. 3.5.), необхідні для розробки. Основана бібліотека для розробки бота — aiogram, яка виділяється своєю швидкістю реакції на повідомлення, і також вона підтримує асинхронність, яка потрібна нам для роботи кількох менеджерів з кількома користувачами.

Бібліотека asyncio — для асинхронної роботи бота, тобто можна прийняти кілька повідомлень від кількох користувачів і опрацювати її одночасно. Без цієї бібліотеки ботом міг би користуватись тільки 1 менеджер, у нас передбачений функціонал додання бота в чат з необмеженою кількістю(до 200 000) менеджерів.

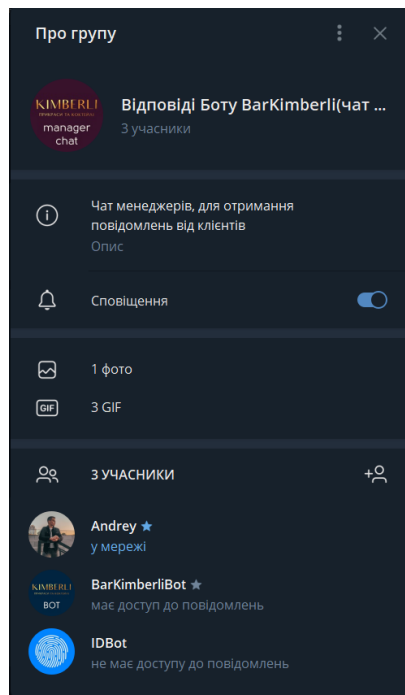
Бібліотека aiohttp надає асинхронний HTTP-клієнт, побудований поверх asyncio. Ця бібліотека дозволяє надсилати запити послідовно, але не чекаючи першої відповіді, перш ніж надсилати нову. На відміну від конвеєрної передачі HTTP, aiohttp відправляє запити щодо кількох з'єднань паралельно, уникаючи проблеми, описаної раніше.

```
import asyncio

from aiohttp import web
from aiogram import Bot, Dispatcher
from aiogram.bot.api import TelegramAPIServer
from aiogram.dispatcher.webhook import configure_app
```

*Рис. 3.5. Імпорт бібліотек*

Створюємо групу для менеджерів, куди додаємо бота(рис. 3.6.). Саме в цю групу будуть надсилатись повідомлення від користувачів. Також дізнаємось ID групи від спеціального бота getmyid, це потрібно, щоб тільки в цій групі бот працював.



*Рис. 3.6. Група для менеджерів для опрацювання звернень від клієнтів*

Додаємо в код програми бота токен, ід групи менеджерів(рис. 3.7.).

```
@dataclass
class Bot:
    token: str
    admin_chat_id: int
    remove_sent_confirmation: bool

def load_config() -> Config:
    return Config(
        bot=Bot(
            token='5026426942:AAHz9G6XoT-9P7475pLjhRuXJQTbDSzQj9g',
            admin_chat_id=-651976812,
            remove_sent_confirmation=getenv("REMOVE_SENT_CONFIRMATION", "yes") in ("yes", "1", "true")
        ),
    )
```

*Рис. 3.7. Додання в код бота токenu та ід групи менеджерів*

Додаємо вітальне повідомлення та кнопку /help, що надсилає справку щодо користування ботом(рис. 3.8.-3.10.)

```

async def cmd_help_user(message: types.Message):
    """
    Довідка для користувача

    :param message: повідомлення від користувача з командою /help
    """
    await message.answer(
        "З мою допомогою ти можеш зв'язатися з менеджером та отримати від "
        "нього відповідь. Просто продовжуй писати в цей діалог, але врахуй, що "
        "підтримуються не всі типи повідомлень, а лише текст, фото, відео, "
        "аудіо, файли та голосові повідомлення.")

async def cmd_start_user(message: types.Message):
    """
    Вітальне повідомлення від бота користувачеві

    :param message: повідомлення від користувача з командою /start
    """
    await message.answer(
        "Привіт 🍷\n"
        "Я бот KimberliBar 🍷\n"
        "З мою допомогою ти можеш забронювати столик, або отримати відповідь на будь-яке інше питання 🍷\n"
        "Просто напиши своє питання, а я приєднаю вільного менеджера 🍷\n\n"
        "Київ, вул. Московська, 8 🍷\n"
        "barkimberli.com.ua 🍷\n"
        "+380 67 852 08 08 🍷")

```

*Рис. 3.8. Вітальне повідомлення користувачу, та кнопка /help, що дає справку щодо користування ботом*

```

def register_usermode_handlers(dp: Dispatcher):
    dp.register_message_handler(cmd_start_user, commands="start")
    dp.register_message_handler(cmd_help_user, commands="help")
    dp.register_message_handler(text_message, content_types=ContentType.TEXT)
    dp.register_message_handler(supported_media, content_types=[
        ContentType.ANIMATION, ContentType.AUDIO, ContentType.PHOTO,
        ContentType.DOCUMENT, ContentType.VIDEO, ContentType.VOICE
    ])
    dp.register_message_handler(unsupported_types, content_types=ContentType.ANY)

```

Рис. 3.9. Реєстрація кнопок start та help в коді програми

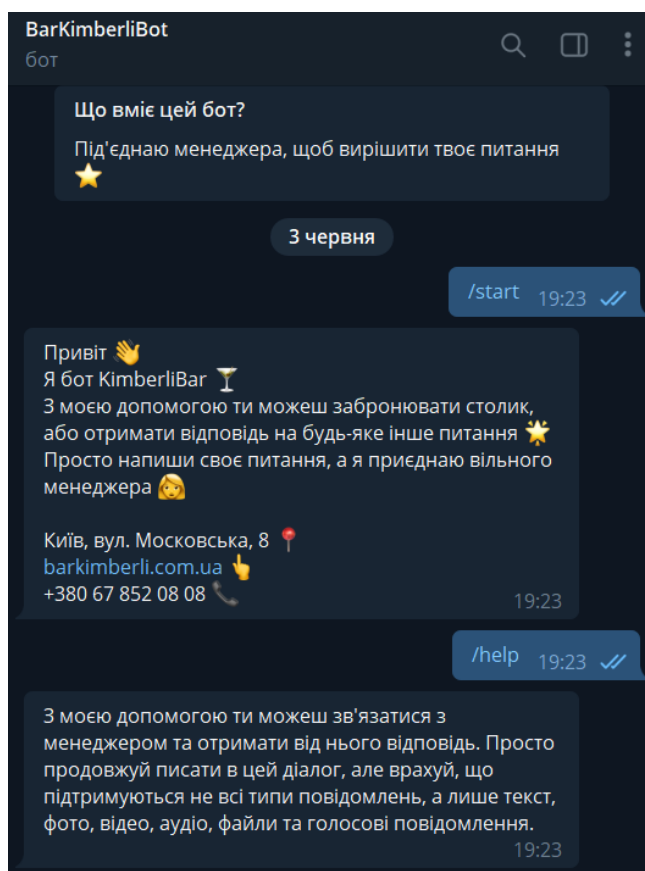


Рис. 3.10. Демонстрація функціоналу кнопок start та help

Додаємо можливість пересилання повідомлень(рис. 3.11.-3.13.). Додаємо функціонал фільтрації повідомлень за довжиною символів(до 4000, без файлу, до 1000 без медіафайлу), перевіряємо на медіафайли, котрі не можна пересилати, перевіряєм, чи заблокований користувач.



```

async def text_message(message: types.Message):
    """
    Хендлер на текстові повідомлення від користувача

    :param message: повідомлення від користувача для адміна(-ів)
    """
    if len(message.text) > 4000:
        return await message.reply("На жаль, довжина цього повідомлення перевищує допустимий розмір. "
                                     "Будь ласка, скороти свою думку і спробуй ще раз.")
    admin_chat_id = message.bot.get("admin_chat_id")

    if message.from_user.id in banned:
        await message.answer("На жаль, ти був заблокований, і твої повідомлення не будуть доставлені.")
    elif message.from_user.id in shadowbanned:
        return
    else:
        await message.bot.send_message(
            admin_chat_id, message.html_text + f"\n\n#{message.from_user.id}", parse_mode="HTML"
        )
        await create_task(_send_expiring_notification(message))

```

*Рис. 3.11. Код пересилання повідомлень користувача і перевірки його на кількість символів*

```

async def supported_media(message: types.Message):
    """
    Хендлер на медіафайли від користувача.
    Підтримуються лише типи, до яких можна додати підпис

    :param message: медіафайл від користувача
    """
    if message.caption and len(message.caption) > 1000:
        return await message.reply("На жаль, довжина підпису медіафайлу перевищує допустимий розмір."
                                     "Будь ласка, скороти свою думку і спробуй ще раз.")
    admin_chat_id = message.bot.get("admin_chat_id")
    await message.copy_to(admin_chat_id,
                          caption=((message.caption or "") + f"\n\n#{message.from_user.id}"),
                          parse_mode="HTML")
    await create_task(_send_expiring_notification(message))

async def unsupported_types(message: types.Message):
    """
    Хендлер на типи повідомлень, що не підтримуються, тобто. ті, до яких не можна додати підпис

    :param message: повідомлення від користувача
    """
    # Игнорируем служебные сообщения
    if message.content_type not in (
        ContentType.NEW_CHAT_MEMBERS, ContentType.LEFT_CHAT_MEMBER, ContentType.VOICE_CHAT_STARTED,
        ContentType.VOICE_CHAT_ENDED, ContentType.VOICE_CHAT_PARTICIPANTS_INVITED,
        ContentType.MESSAGE_AUTO_DELETE_TIMER_CHANGED, ContentType.NEW_CHAT_PHOTO, ContentType.DELETE_CHAT_PHOTO,
        ContentType.SUCCESSFUL_PAYMENT, ContentType.PROXIMITY_ALERT_TRIGGERED,
        ContentType.NEW_CHAT_TITLE, ContentType.PINNED_MESSAGE):
        await message.reply("На жаль, цей тип повідомлення не підтримується "
                             "для надсилання від користувачів. Відправте щось інше.")

```

*Рис. 3.12. Код пересилання повідомлень користувача і перевірки його на невідримувані типи*

```

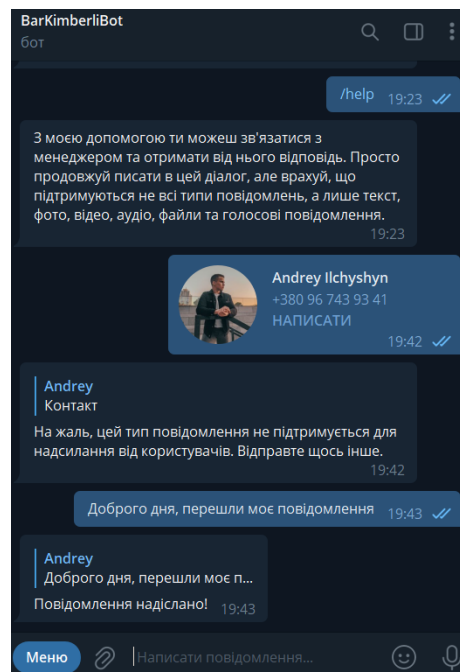
async def _send_expiring_notification(message: types.Message):
    """
    Відправляє "самознищуване" через 5 секунд повідомлення

    :param message: повідомлення, на яке бот відповідає підтвердженням надсилання
    """
    remove_sent_confirmation = message.bot.get("remove_sent_confirmation")
    msg = await message.reply("Повідомлення надіслано!")
    if remove_sent_confirmation:
        await sleep(5.0)
        await msg.delete()

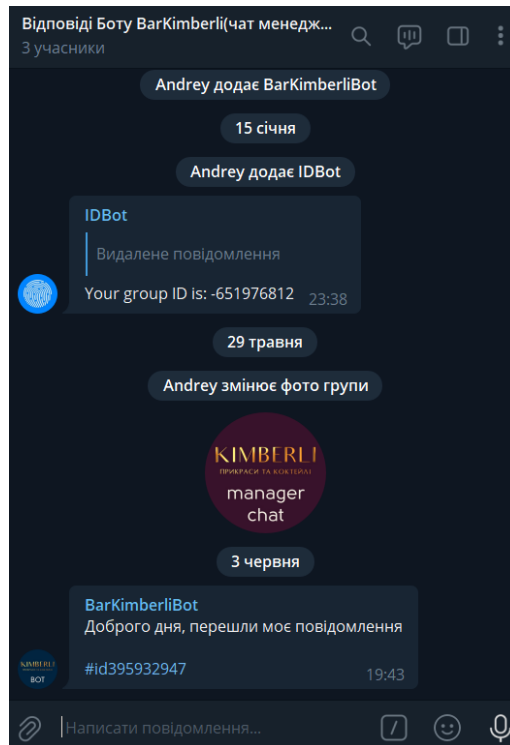
```

*Рис.3.13. Код відправки повідомлення від бота, про успішне надіслання повідомлення менеджеру*

При тестуванні функціоналу пересилання повідомлення(рис. 3.14.), повідомлення було надіслано в групу менеджерів(рис. 3.15.).



*Рис. 3.14. Демонстрація функціоналу фільтрації повідомлень за типами і пересилання*



*Рис. 3.15. Демонстрація функціоналу пересилання повідомлень від користувачів в чат менеджерів*

Додаємо можливість пересилання повідомлень від менеджерів користувачам(рис. 3.16.-3.19.). Додаємо перевірки на можливість пересилання повідомлень користувачу. Якщо користувач заблокував бота, або неможливо буде переслати йому повідомлення, то в чаті менеджерів буде повідомлення про це.

```

async def reply_to_user(message: types.Message):
    """
    Відповідь адміністратора на повідомлення користувача (надіслане ботом).
    Використовується метод copy_message, тому можна відповідати чим завгодно
    """
    .reply message: повідомлення від адміну, яке є відповіддю на інше повідомлення
    """

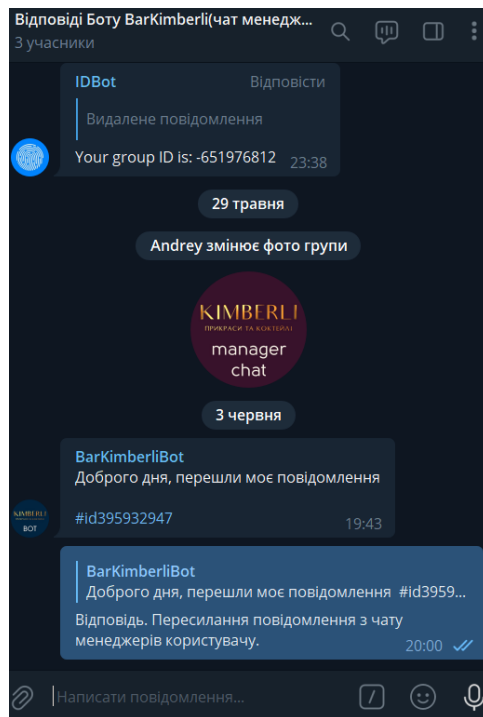
    try:
        user_id = extract_id(message)
    except ValueError as ex:
        return await message.reply(str(ex))

    # Вирізаємо ID і намагаємося надіслати копію повідомлення.

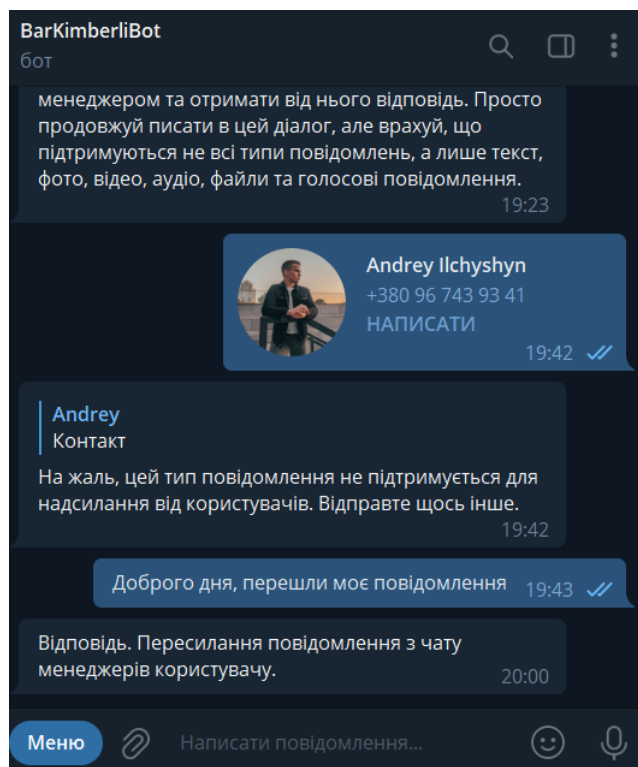
    try:
        await message.copy_to(user_id)
    except BotBlocked:
        await message.reply("Неможливо відправити повідомлення адресату, так як бот заблокований з його сторони")
    except TelegramAPIError as ex:
        await message.reply(f"Неможливо надіслати повідомлення адресату! Помилка: {ex}")

```

*Рис. 3.16. Код функціоналу пересилання повідомлень з чату менеджерів користувачам*



*Рис. 3.17. Демонстрація функціоналу пересилання повідомлень з чату менеджерів користувачам*



*Рис. 3.18. Демонстрація функціоналу отримання користувачем повідомлень з чату менеджерів*

Додаємо можливість менеджеру подивитись ім'я, id, посилання на користувача, котрий звернувся та додаємо кнопки /get, /who, при написанні яких та пересиланні повідомлення від користувача, буде надіслана інформація про користувача, та додаємо кнопку /help для справки в чаті менеджерів (рис.3.18.-3.19.).

```
async def get_user_info(message: types.Message):
    try:
        user_id = extract_id(message)
    except ValueError as ex:
        return await message.reply(str(ex))
    try:
        user = await message.bot.get_chat(user_id)
    except TelegramAPIError as ex:
        return await message.reply(f"Неможливо отримати інформацію про користувача! Помилка: {ex}")
    u = f"@{user.username}" if user.username else 'ні'
    await message.reply(f"Ім'я: {user.full_name}\n\nID: {user.id}\n\nUsername: {u}")

async def admin_help(message: types.Message):
    await message.answer("Доступні наступні команди адміністратора:\n\n"
                        "/get або /who (у відповідь на повідомлення) — запит інформації про користувача за його ID.")

def register_adminmode_handlers(dp: Dispatcher, admin_chat_id: int):
    dp.register_message_handler(get_user_info, IsReplyFilter(is_reply=True), IDFilter(chat_id=admin_chat_id),
                                commands=["get", "who"])
    dp.register_message_handler(admin_help, IDFilter(chat_id=admin_chat_id), commands="help")
    dp.register_message_handler(reply_to_user, IsReplyFilter(is_reply=True), IDFilter(chat_id=admin_chat_id),
                                content_types=types.ContentTypes.ANY)
```

Рис. 3.19. Код функціоналу отримання інформації про користувача, що пише повідомлення менеджерам, та кнопок /who, /get, /help

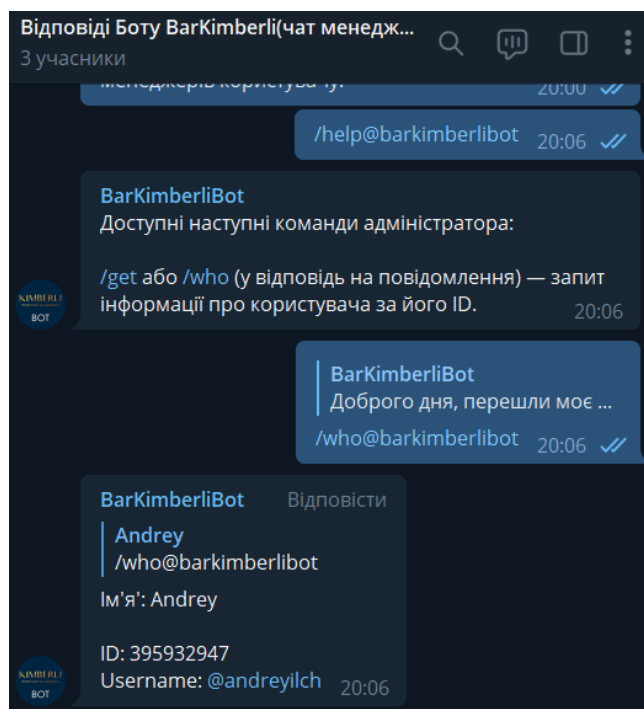


Рис. 3.20. Демонстрація функціоналу отримання інформації про користувача, та кнопок /who, /get, /help

Додаємо можливість менеджеру додати користувача в чорний список із сповіщенням або без сповіщення для клієнта про це(рис. 3.21.-3.27.). Якщо додати клієнта в цей список, то його повідомлення не будуть надсилатись в чат менеджерів.

Отримав такі кнопки:

- /ban – кнопка додає користувача в чорний список із сповіщенням клієнта про це.
- /shadowban – кнопка додає користувача в чорний список без сповіщенням клієнта про це.
- /unban – кнопка видаляє користувача в чорного списку.
- /list\_banned – кнопка виводить список користувачів доданих до чорного списку.

```
def register_bans_handlers(dp: Dispatcher, admin_chat_id: int):
    dp.register_message_handler(cmd_ban, IsReplyFilter(is_reply=True), IDFilter(chat_id=admin_chat_id),
                                commands="ban")
    dp.register_message_handler(cmd_shadowban, IsReplyFilter(is_reply=True), IDFilter(chat_id=admin_chat_id),
                                commands="shadowban")
    dp.register_message_handler(cmd_unban, IsReplyFilter(is_reply=True), IDFilter(chat_id=admin_chat_id),
                                commands="unban")
    dp.register_message_handler(cmd_list_banned, IDFilter(chat_id=admin_chat_id),
                                commands="list_banned")
```

Рис. 3.21. Код функціоналу кнопок /ban, /shadowban, /unban, /list\_banned

```
async def cmd_ban(message: types.Message):
    try:
        user_id = extract_id(message)
    except ValueError as ex:
        return await message.reply(str(ex))
    banned.add(int(user_id))
    await message.reply(
        f"ID {user_id} додано до списку заблокованих. "
        f"Якщо спробує надіслати повідомлення, користувач отримає сповіщення про блокування."
    )

async def cmd_shadowban(message: types.Message):
    try:
        user_id = extract_id(message)
    except ValueError as ex:
        return await message.reply(str(ex))
    shadowbanned.add(int(user_id))
    await message.reply(
        f"ID {user_id} додано до списку потай заблокованих. "
        f"Якщо спробує надіслати повідомлення, користувач не дізнається, що його заблоковано."
    )
```

Рис. 3.22. Код функціоналу додавання до чорного списку

```

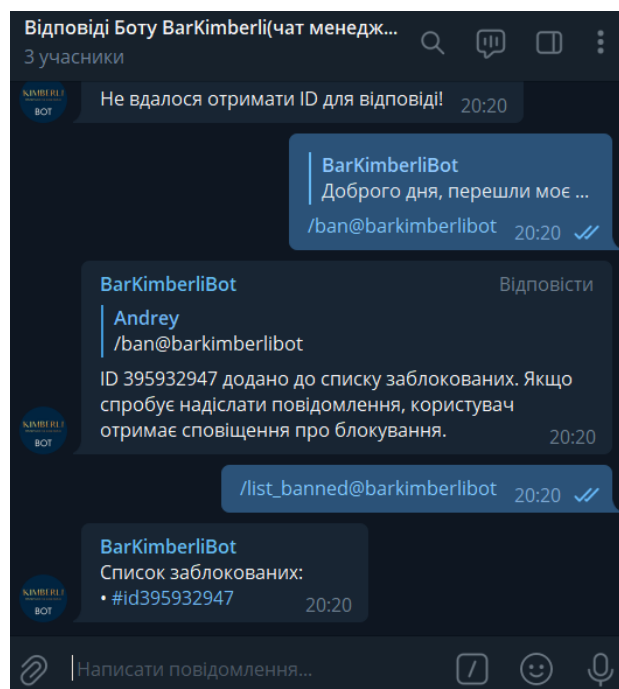
async def cmd_unban(message: types.Message):
    try:
        user_id = extract_id(message)
    except ValueError as ex:
        return await message.reply(str(ex))
    user_id = int(user_id)
    with suppress(KeyError):
        banned.remove(user_id)
    with suppress(KeyError):
        shadowbanned.remove(user_id)
    await message.reply(f"ID {user_id} розблоковано")

async def cmd_list_banned(message: types.Message):
    has_bans = len(banned) > 0 or len(shadowbanned) > 0
    if not has_bans:
        await message.answer("Немає заблокованих користувачів")
        return
    result = []
    if len(banned) > 0:
        result.append("Список заблокованих:")
        for item in banned:
            result.append(f"• #id{item}")
    if len(shadowbanned) > 0:
        result.append("\nСписок скритих заблокованих:")
        for item in shadowbanned:
            result.append(f"• #id{item}")

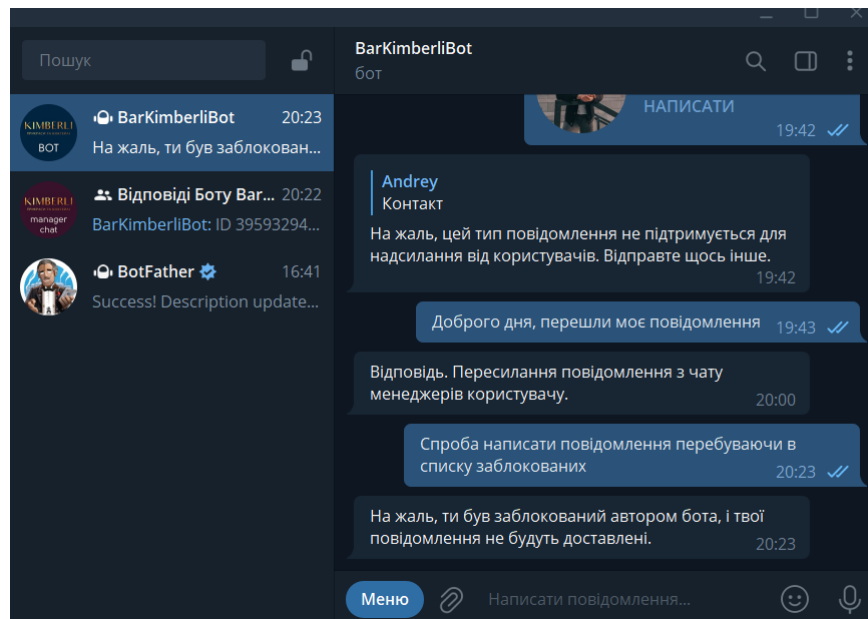
    await message.answer("\n".join(result))

```

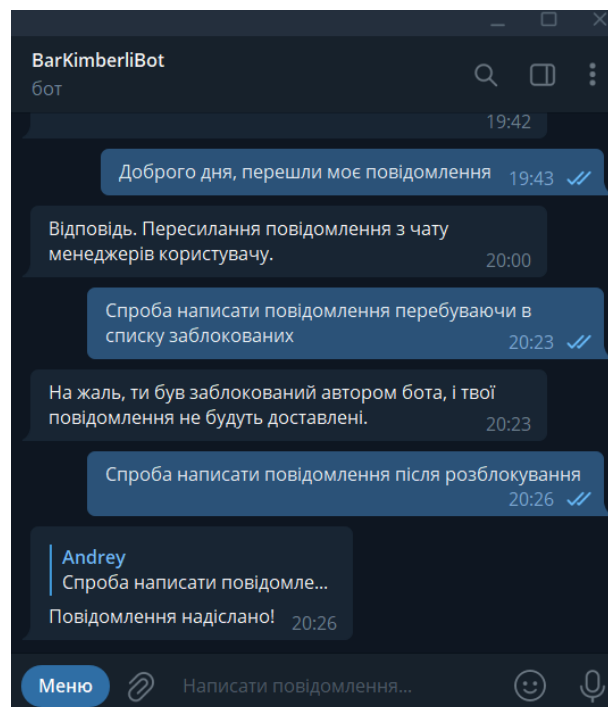
*Рис. 3.23. Код функціоналу видалення користувача із чорного списку та виведення списку цих користувачів*



*Рис. 3.24. Демонстрація функціоналу додавання користувача в список заблокованих і вивід цього списку*



*Рис. 3.25. Демонстрація функціоналу додавання користувача в список заблокованих зі сторони користувача. Повідомлення від користувача не пересилаються в чат менеджерів*



*Рис. 3.26. Відображення функціоналу видалення з чорного списку з сторони користувача*



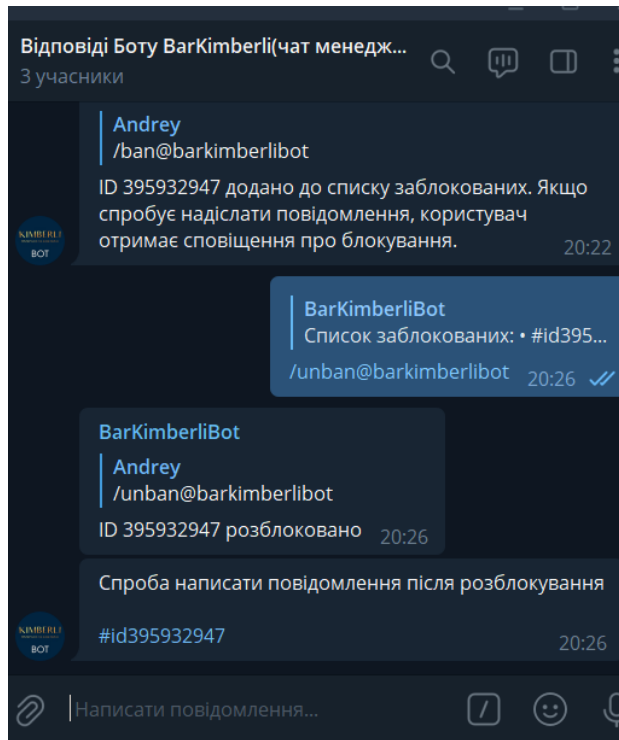


Рис. 3.27. Приклад роботи боту(видалення користувача з чорного списку)

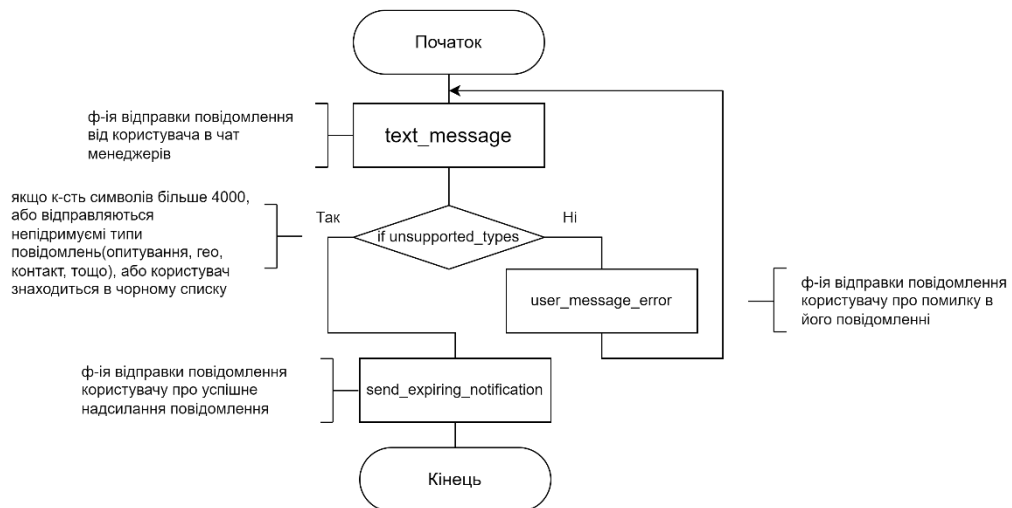


Рис. 3.28. Робота боту barkimberli на сторони клієнтів(схема алгоритму)

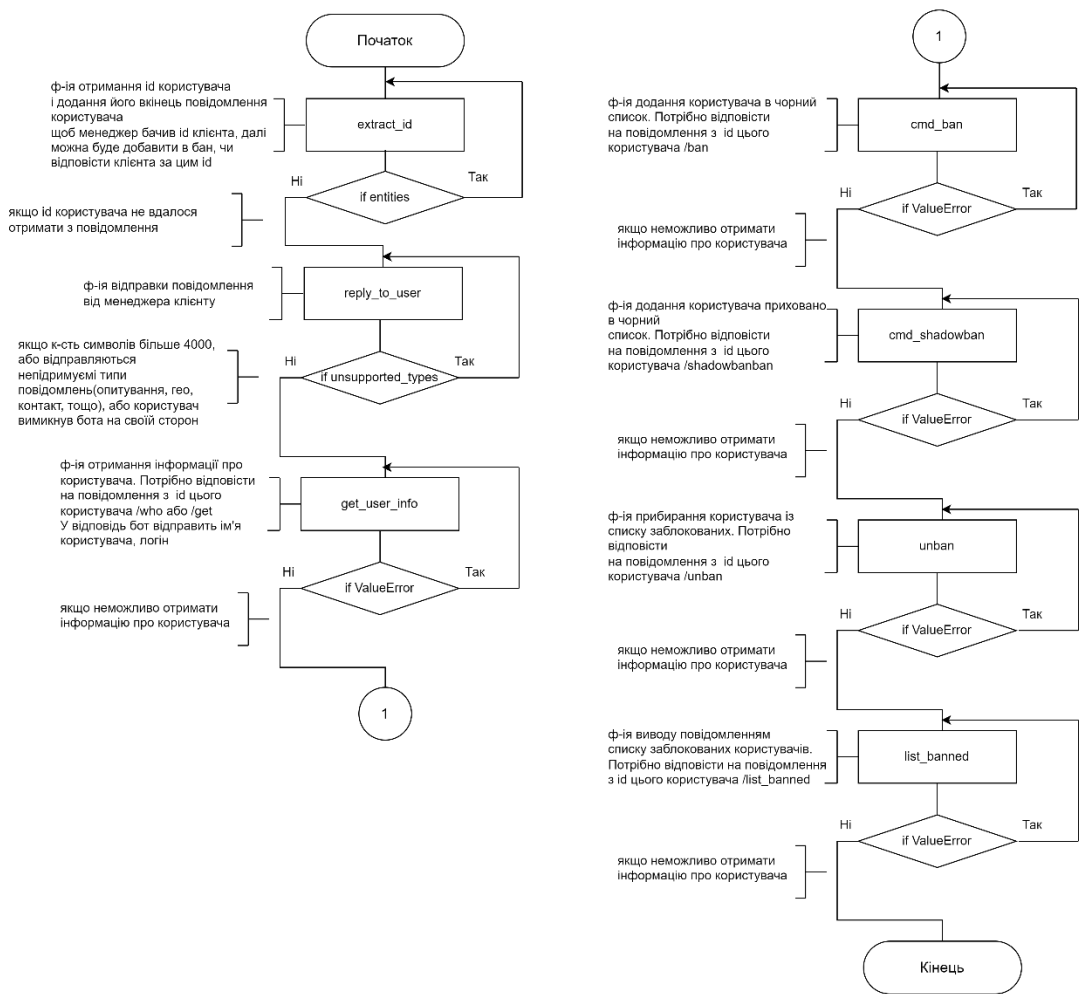


Рис. 3.29. Робота боту bakrimberli на стороні менеджерів(схема алгоритму)

### 3.3. Висновки до розділу

Цей розділ містить опис BotFather і процесу створення власного боту. Пояснено функції та перелік можливостей цього інтерфейсу. API Telegram Bot дозволяє підключати ботів до системи. BotFather — це головний бот всього месенджеру Telegram, за допомогою нього створюють нових ботів, видаляють старих, можна редагувати назву боту, опис, головне фото бота, поле "Про бота".

Наведено опис розробки та впровадження нового бота для закладу харчування у програмі Telegram на мові програмування Python високого рівня. Програма успішно протестована. Наведені рисунки, на яких показана реалізація кнопок, відтворення кнопок та функціоналу у роботі з ботом, а також зовнішній вигляд боту.

## ВИСНОВКИ

В ході виконання дипломного проєкту було проведено дослідження та аналіз програмного продукту «Telegram-бот для підтримки клієнтів закладів громадського харчування» в рамках дипломного проєкту в сучасному додатку-месенджеру Telegram, розглянуто недоліки та пропозиції щодо вдосконалення підходів, методів програмного продукту та алгоритми, а також етапи та кроки розробки.

Після дослідження та виявлення проблеми створення програмного продукту для надання швидкої відповіді на запитання клієнтам закладів харчування, в Telegram був створений бот з необхідними можливостями.

У процесі реалізації даного дипломного проєкту також було визначено інструменти, що використовуються для створення ботів, і навіть для подальшого управління розробленими ботами і основні проблеми, що виникають під час створення даного програмного продукту.

Також були враховані основні складові процесу розробки даного програмного продукту, принципи створення надбудови, ієрархія взаємодії користувача та бота, бібліотеки, їх призначення та сфери застосування. З'ясувалося, що для будь-якого робота в додатку Telegram необхідно використовувати інтерфейс BotFather та його логіку.

Було порівняно різні популярні Telegram-боти та їх призначення, особливості та можливості. Проаналізувавши методи-аналоги розробки, було визначено їх переваги, їх колективне використання, синхронізацію тощо.

У рамках дипломного проєкту в додатку Telegram було розроблено бота, який має на меті оперативно відповідати на запитання клієнтів ресторану. Цей бот призначений для швидкого підключення менеджера та спілкування з клієнтом. Також є можливість додавати та видаляти клієнтів із «чорного» списку.

Перевага розробленого додатку в тому, що він адаптований і розроблений у вигляді безкоштовної програми і доступної для всіх користувачів, що сприяє більшому поширенню розробленого боту.

Основними результатами цієї роботи є:

- повний аналіз проблеми створення програмного продукту «Telegram-бот для підтримки клієнтів закладів»;
- розгляд недоліків та пропозиції щодо вдосконалення підходів, методів та алгоритмів створення програмних продуктів;
- розроблено додаток Telegram-бот, метою якого є швидко надати відповідь клієнтам ресторану на поставлене запитання.

## СПИСОК БІБЛЮГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Schütze, Hinrich, Christopher D. Manning, Raghavan, Prabhakar: Introduction to information retrieval, Prabhakar/ Cambridge, UK: Cambridge University Press, 2008. – 256 s. – Bibliogr.: s. 121– 125.
2. Telegram bot – [Електронний ресурс] Режим доступу: <https://evergreens.com.ua/ru/articles/bot-reviews.html/>(дата звернення 29.05.2022) – Назва з екрана.
3. Python – [Електронний ресурс] Режим доступу: <http://www.plug.org.ua/documentation/about-python/>(дата звернення 29.05.2022) – Назва з екрана
4. Автоматизовані інформаційно-пошукові мови. [Електронний ресурс]. — Режим доступу: <http://ubooks.com.ua/books/00092/inx13.php> (дата звернення 30.05.2022 р) — Назва з екрана.
5. Sergey Brin, Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine/ Computer Science Department, Stanford University, Stanfor, 1998. – 126 s. – Bibliogr.: s. 45– 48.
6. Benjamin M. Schmidt & Matthew M. Chingos: Ranking Doctoral Programs by Placement A New Method/ Cambridge, UK: Cambridge University Press, 2007. – 654 s. – Bibliogr.: s. 523–529.
7. Matthew Richardson, Pedro Domingos: The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank/The Politics of Search, 2001. – 456 s. – Bibliogr.: s. 364–374.
8. Бойченко С.В., Іванченко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.
9. ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення».

10. ГОСТ 2.106-96 «Единая система конструкторской документации. Текстовые документы».