

Київський національний торговельно-економічний університет
Кафедра програмної інженерії та інформаційних систем

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Розробка системи адміністрування віртуальних спільнот на платформі Telegram Bot»

Студента 4 курсу, 7 групи,
спеціальності 121 «Інженерія
програмного забезпечення»

Мілевського Дмитра
Володимировича

Науковий керівник
кандидат педагогічних наук,
старший викладач

Жирова Тетяна
Олександрівна

Гарант освітньої програми
кандидат технічних наук,
доцент

Цензура Микола
Олександрович

КИЇВ – 2020

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ОСНОВНІ ВІДОМОСТІ ТА ТЕХНІЧНЕ ЗАВДАННЯ.....	6
1.1. Загальні відомості	6
1.2. Застосування ботів для адміністрування спільнот.....	8
1.3. Технічне завдання	8
1.4. Теоретичні відомості.....	9
1.4.1. AsyncIO.....	9
1.4.2. База даних.....	12
1.4.3. Система керування базами даних	12
1.4.4. PostgreSQL.....	14
1.4.5. MySQL.....	17
1.4.6. MariaDB.....	21
1.5. Вимоги до системи.....	21
1.6. Висновки до розділу 1	21
РОЗДІЛ 2. АНАЛІЗ ПРЕДМЕТУ ДОСЛІДЖЕННЯ ТА ЙОГО ХАРАКТЕРИСТИКА	22
2.1. Опис програми «PyCharm».....	22
2.2. Конкурентність програми.....	23
2.2.1. ControllerBot	23
2.2.2. LivegramBot.....	24
2.2.3. VoteBot	25
2.3. Висновки до розділу 2	25
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АДМІНІСТРУВАННЯ ВІРТУАЛЬНИХ СПІЛЬНОТ.....	26
3.1. Програмне забезпечення «pyGroupMgr».....	26
3.2. Діаграми 27	
3.3. Інтерфейс користувача.....	29

					<i>КНТЕУ 121 07-11.БР</i>		
					<i>Система адміністрування віртуальних спільнот на платформі Telegram Bot</i>		
					<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Зміст	2	44
Зав. каф.		Криворучко О.В.			<i>Зміст</i>		
Керівник		Жирова Т.О.					
Гарант		Цензура М.О.					
Розроб		Мілевський Д.В.					
					Факультет інформаційних технологій, 4 курс, 7 група		

3.4. Висновки до розділу 3.....	41
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	43
ДОДАТКИ.....	45

						Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата	КНТЕУ 121 07-11.БР	
					3	

ВСТУП

Telegram є одним із найпопулярніших месенджерів, програмного забезпечення для персональних комп'ютерів, смартфонів та планшетів, що дозволяє користувачам обмінюватись текстовими повідомленнями, медіа файлами, а також безкоштовно телефонувати іншим користувачам.

Для месенджера Telegram був створений спеціальний протокол MTProto, що передбачає використання декількох протоколів шифрування. При авторизації та аутентифікації використовуються алгоритми RSA-2048, DH-2048 для шифрування. При передачі повідомлень протоколу у мережу вони шифруються AES із ключем, що відомий клієнту та серверу. Із переходом на протокол MTProto 2.0 застосовується криптографічний хеш-алгоритм SHA-256.

У 2013р. в месенджері з'явився режим секретних чатів. Цей режим реалізує шифрування, при якому лише відправник та отримувач володіють спільним ключем (end-to-end шифрування), із використанням алгоритму AES-256 в режимі IGE (Infinite Garble Extension) для відправки повідомлень. На відміну від звичайного режиму, повідомлення у секретних чатах не розшифровуються сервером, історія переписки зберігається лише на тих двох пристроях, на котрих було створено чат.

При обміні файлами можна відправляти як файли пристрою, так і шукати медіа контент в інтернеті, в тому випадку, коли використовується мобільна версія для iOS або Android. Розмір файлів, що передаються, обмежено в 1.5 Гб. Програма використовує систему довантаження файлів після втрати підключення.

Існує можливість змінювати форматування тексту, роблячи його жирним, курсивом або моноширним.

Наразі офіційні клієнти для месенджеру Telegram включаються в себе:

					<i>КНТЕУ 121 07-11.БР</i>			
					<i>Система адміністрування віртуальних спільнот на платформі Telegram Bot</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		В	4	44
Зав. каф.		Криворучко О.В.			<i>Вступ</i>			
Керівник		Жирова Т.О.						
Гарант		Цензура М.О.						
Розроб		Мілевський Д.В.			Факультет інформаційних технологій, 4 курс, 7 група			

- Мобільний додаток для Android та iOS/iPadOS;
- Настільні додатки для Windows, Linux та macOS;
- Веб-додаток, веб-додаток для Chrome, веб додаток для React.

Окрім звичайних та секретних чатів у Telegram є групи (спільноти) та канали. Групи – це окремий вид чатів, в котрому можливе спілкування одразу великої кількості учасників. Спочатку групи могли вміщати лише до 200 учасників.

У 2015 р. супергрупи могли вміщати до 1000 учасників.

Потім у 2016 р. ліміт було збільшено до 5000 учасників.

У 2017 р. розмір супергруп збільшився до 10000 учасників, пізніше у 2018 р. – до 100000.

З 2019 р. і дотепер, супергрупи можуть вміщати до 200000 учасників.

Кількість спільнот та їх розміри постійно зростають, а з цим зростає і складність їх адміністрування. Власник спільноти може надавати адміністративні права іншим учасникам, що, звісно, полегшує процес адміністрування. Проте деякі процеси можна навіть автоматизувати, для цього месенджер Telegram надає Telegram Bot API.

За допомогою Bot API інші розробники можуть створювати «ботів», спеціальні акаунти, що керуються програмами. Типові боти відповідають на спеціальні команди у персональних та групових чатах, також вони можуть здійснювати пошук в інтернеті або виконувати інші задачі. Боти застосовуються для розважальних цілях та бізнесі. Telegram надає досить великі можливості ботам, серед яких можна виділити альтернативні клавіатури, клавіатури під повідомленнями, вбудований режим, за допомогою якого бота можна використовувати у будь-якому чаті, навіть якщо він не є його учасником. Одним із гарних прикладів застосування вбудованого режиму є боти, що здійснюють пошук в інтернеті.

					<i>КНТЕУ 121 07-11.БР</i>	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		5

РОЗДІЛ 1.

ОСНОВНІ ВІДОМОСТІ ТА ТЕХНІЧНЕ ЗАВДАННЯ

1.1. Загальні відомості

Спільноти можуть вміщати багатьох користувачів. Вони завжди мають одного власника та можуть мати декількох адміністраторів. Також у спільнотах можуть бути присутні боти. Вони мають режим приватності, при увімкненні якого власником боту, він не буде отримувати усі повідомлення групи, а лише ті, які містять його ім'я користувача або є командою бота, тобто повідомлення, що починаються із символу « / » (слеш). Власник групи також може зробити іншого користувача власником, проте втратить свої права та стане просто адміністратором.

Адміністратори можуть змінювати глобальні та персональні дозволи для учасників

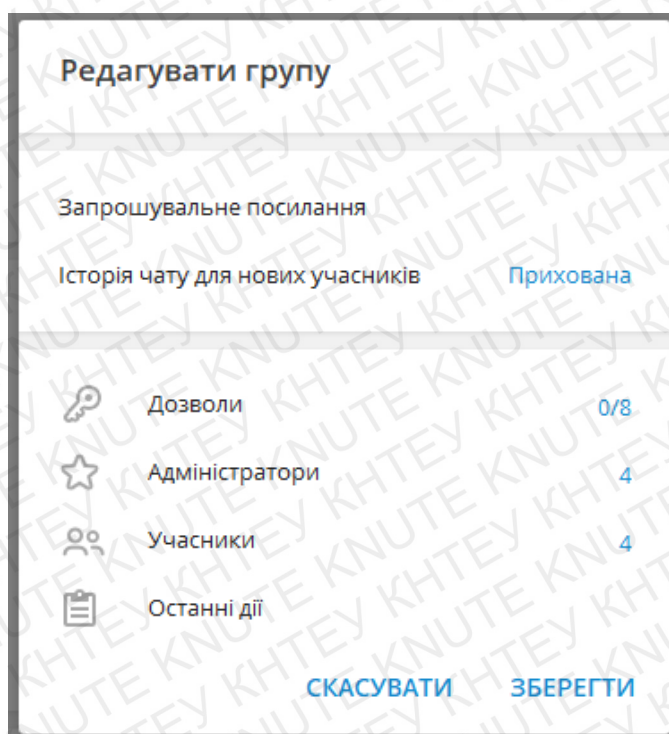


Рис 1.1. діалогове вікно керування групою

					<i>КНТЕУ 121 07-11.БР</i>			
					<i>Система адміністрування віртуальних спільнот на платформі Telegram Bot</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		Р1	6	44
Зав. каф.		Криворучко О.В.			<i>Основні відомості та технічне завдання</i>	Факультет інформаційних технологій, 4 курс, 7 група		
Керівник		Жирова Т.О.						
Гарант		Цензура М.О.						
Розроб		Мілевський Д.В.						

На рис. 1.1. зображено вікно керування групою. Тут адміністратори можуть налаштувати приховування старих повідомлень для нових учасників, перейти у меню керування дозволами, адміністраторами, учасниками та переглядати останні дії, такі як зміна назви групи, дозволів, видалення або редагування повідомлень учасниками групи.

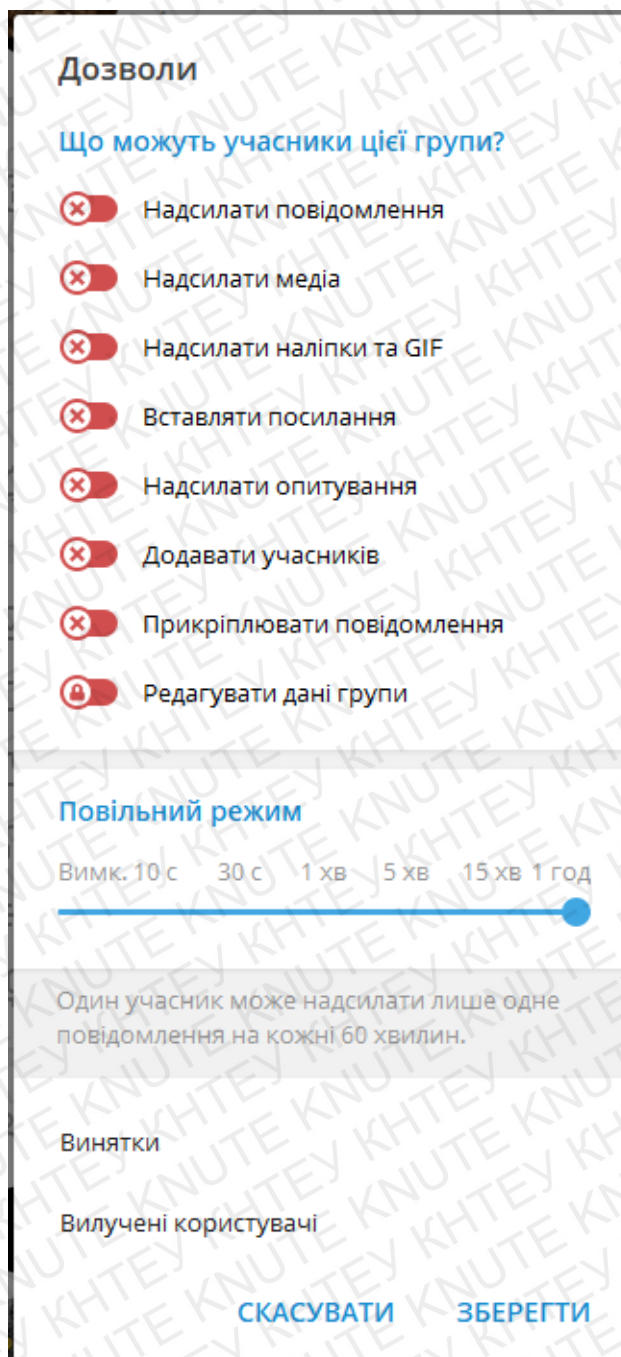


Рис 1.2. Вікно керування дозволами

На рис. 1.2. зображено вікно керування дозволами, де адміністратори можуть змінювати як глобальні дозволи, так і індивідуальні для кожного

					Аркуш
					7
Изм.	Аркуш	№ докум	Подпись	Дата	

учасника групи (меню «Винятки»). Також тут вони можуть прибирати користувачів зі списку вилучених, що дозволить їм знову приєднатися до групи. Повільний режим встановлює, через який час користувач зможе відправити нове повідомлення після відправки попереднього. Це корисно для груп із дуже великими кількостями користувачів

1.2. Застосування ботів для адміністрування спільнот

Ботам надаються великі можливості. Telegram вже має такі функції, як опитування, повільний режим та ін., проте боти дозволяють альтернативні, персоналізовані під потреби користувачів та груп варіанти. Наприклад замість часу, що користувач має чекати між відправками повідомлень, можна зробити ліміт на кількість повідомлень за певний час. Так само можна персоналізувати і опитування, або просто використовувати бота для автоматичного їх створення, відправки та моніторингу.

1.3. Технічне завдання

Бот має допомагати в адмініструванні груп із великою кількістю учасників та мати такі функції:

- Базові адміністративні активності такі як вилучення, блокування
- Система попереджень
- Автоматичне вилучення неактивних користувачів
- Привітання нових учасників
- Тимчасове прибирання прав відправки повідомлень для нових учасників
- Публічна база вилучень та сповіщення при вході користувача, що був вилучений з іншої групи
- Фільтрація флуду
- Встановлення мови спільноти та автоматичний переклад

Для реалізації таких цілей було обрано мову програмування Python. Програми, написані на Python легко переносяться, адже ця мова програмування є крос-платформною, а також інтерпретуємою, тобто не має потреби компілювати

					КНТЕУ 121 07-11.БР	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		8

код програм під різні пристрої. Найголовнішою причиною вибору мови програмування Python є AsyncIO (асинхронний ввід та вивід).

1.4. Теоретичні відомості

1.4.1. AsyncIO

AsyncIO – одна із стандартних бібліотек мови програмування Python, що дає можливість писати конкурентний код використовуючи синтаксис `async/await`. Вона забезпечує інфраструктуру для написання конкурентних однопотокових додатків з використанням співпрограм, мультиплексування доступу до I/O через сокети та інші ресурси, запуск мережових клієнтів та серверів і інших схожих задач. Із AsyncIO використовуються такі терміни, як цикл подій (Event Loop), співпрограми (Coroutine) та задачі (Task), майбутнє (Future).

Цикл подій є основою. Це цикл, що викликає асинхронні задачі, проводить мережові операції вводу та виводу та викликає співпрограми.

Співпрограми – це асинхронні функції. Вони забирають управління циклу подій при виконанні та повертають його тій, що її викликала. Співпрограми не виконуються одразу після виклику. Замість цього повертається спеціальний об'єкт, котрий має бути переданий до циклу подій для виконання.

Майбутнє – це об'єкт, що є результатом ще не виконаної роботи. Цикл подій спостерігає за цим об'єктом та чекає поки робота не буде завершена.

Задача – обгортка для співпрограми та субклас майбутнього.

						Аркуш
					<i>КНТЕУ 121 07-11.БР</i>	9
Изм.	Аркуш	№ докум	Подпись	Дата		

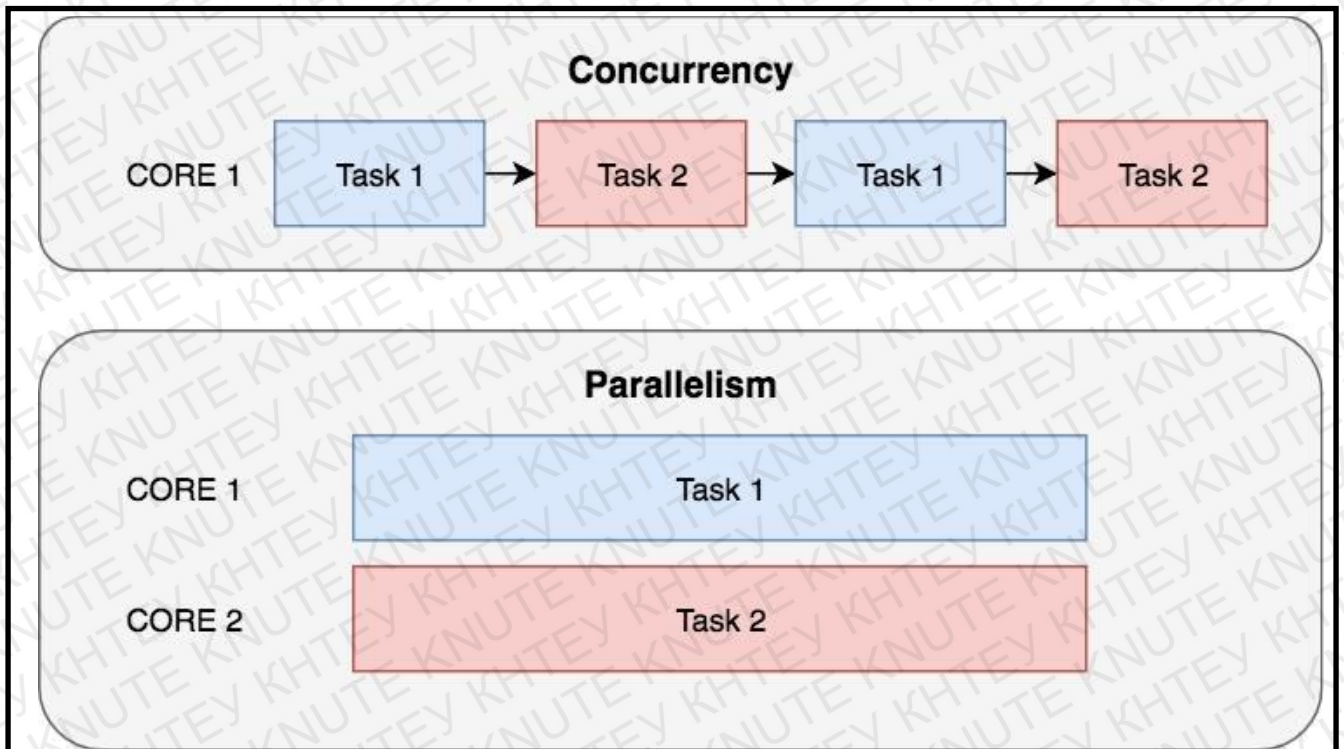


Рис. 1.3. приклад роботи конкуренції та паралелізму

На рис. 1.3. можна побачити, як паралелізм відрізняється від конкуренції, котру надає AsyncIO. При паралелізмі кожна задача виконується в окремому потоці, а при конкуренції відбувається постійне перемикання між задачами в середині одного потоку. Одна конкурентна функція передає управління циклу подій при очікуванні певного результату, наприклад підключення до серверу через мережу. А цикл подій передає управління наступній конкурентній функції.

event loop

no events
enter m1
pause m1
enter m2
pause m2
check events
resume m2
pause m2
check events
resume m1
pause m1
resume m2
...

method 1

```
async def m1():
    prepare request
    await send request
    await receive request
    process request
    return result
```

method 2

```
async def m2():
    prepare parameters
    await send query
    await receive answer
    process answer
    return result
```

event loop

no events
enter m1
pause m1
enter m2
pause m2
check events
resume m2
pause m2
check events
resume m1
pause m1
resume m2
...

method 1

```
async def m1():
    prepare request
    await send request
    await receive request
    process request
    return result
```

method 2

```
async def m2():
    prepare parameters
    await send query
    await receive answer
    process answer
    return result
```

Рис. 1.4. Приклад роботи циклу подій

1.4.2. База даних

База даних є організованою колекцією даних, що загалом зберігається та використовується електронно на комп'ютерній системі. Складні бази даних зазвичай створюються із використанням технологій моделювання. Дані логічно структуровані (систематизовані) для забезпечення можливості їх швидкого пошуку та обробки. База даних включає метадані, що описують її логічну структуру у формальному вигляді.

1.4.3. Система керування базами даних

Система керування базами даних (СКБД) – програмне забезпечення, що взаємодіє із кінцевими користувачами, додатками, та самими базами даних для перехоплення та аналізу даних. Вона надає можливість маніпулювати даними (додавання, оновлення, вилучення та вибір). СКБД забезпечує безпечність, надійність зберігання та цілісність даних, а також надає інструменти адміністрування.

За моделлю даних, СУБД можна класифікувати наступним чином:

- Ієрархічні
- Мережеві
- Реляційні
- Об'єктно-орієнтовані
- Об'єктно-реляційні

Ієрархічна модель даних – це модель даних, де використовується представлення бази даних у вигляді деревоподібної (ієрархічної) структури, що містить об'єкти (дані) різних рівнів. Між об'єктами існують зв'язки, кожен об'єкт може включати в себе декілька об'єктів нижчого рівня. Такі об'єкти знаходяться у відношенні батьківського (об'єкт нижчий до кореня) до нащадка (об'єкт більш низького рівня), при цьому можлива ситуація, коли об'єкт-батько має декілька нащадків, в той час як у об'єкта-нащадка обов'язковий лише один батьківський. Об'єкти, що мають спільний батьківський об'єкт, називаються близнюками. Бази

						Аркуш
					<i>КНТЕУ 121 07-11.БР</i>	12
Изм.	Аркуш	№ докум	Подпись	Дата		

даних із ієрархічною моделлю є одними із найстаріших та стали першими СКБД для мейнфреймів.

Мережева модель даних – логічна модель даних, що є розширенням ієрархічного підходу, строга математична теорія, що описує структурний аспект, аспект цілісності та аспект обробки даних у мережевих базах даних. Різниця між ієрархічною моделлю даних та мережевою полягає в тому, що у ієрархічних структурах запис-нащадок повинен мати саме одного батька, а у мережевій структурі даних у нащадка може бути будь-яка кількість батьків. Мережева БД є набором екземплярів певного типу запису та набору екземплярів певного типу зв'язків між цими записами. Тип зв'язку визначається для двох типів запису: батька та нащадка. Екземпляр типу зв'язку складається з одного екземпляру типу запису батька та впорядкованого набору екземплярів типу запису нащадка. Для даного типу зв'язку L із типом запису батька P та типом запису нащадка C повинні виконуватися наступні дві умови:

- кожен екземпляр типу запису P є батьком тільки в одному екземплярі типу зв'язку L;
- кожен екземпляр типу зв'язку C є батьком не більше ніж в одному екземплярі типу зв'язку L.

Реляційна СКБД – СКБД, керуюча реляційними базами даних. Практично усі розробники сучасних, підтримуючих зв'язок із системами баз даних, орієнтуються на реляційні СКБД. Ціллю нормалізації реляційної бази даних є вилучення недоліків структури баз даних, що приводять до надмірності, котра в свою чергу потенційно приводить до різних аномалій та порушень цілісності даних.

Об'єктно-орієнтована (об'єктна) СКБД – система керування базами даних, заснована на об'єктній моделі даних. Ця система обробляє дані як абстрактні об'єкти, наділені властивостями, що також використовують методи взаємодії із іншими об'єктами навколишнього світу (об'єктами БД).

						Аркуш
					КНТЕУ 121 07-11.БР	13
Ізм.	Аркуш	№ докум	Підпись	Дата		

Об'єктно-реляційна СКБД – реляційна СКБД, що підтримує деякі технології, властиві об'єктно-орієнтованим СКБД, що також реалізують об'єктно-орієнтований підхід: об'єкти, класи та наслідування реалізовані у структурі баз даних та мові запитів. Об'єктно-реляційними СКБД є широко відомі Oracle Database, Informix, DB2, PostgreSQL.

Для даного проекту було обрано PostgreSQL

1.4.4. PostgreSQL

PostgreSQL – вільна об'єктно-реляційна система керування базами даних. Існує у реалізації для великої кількості UNIX-подібних платформ, а також Microsoft Windows.

Базується на мові SQL та підтримує багато можливостей стандарту SQL:2011. Версія 12 має наступні обмеження:

Таблиця 1.1

ОБМЕЖЕННЯ POSTGRESQL

Тип обмеження	Обмеження
Максимальний розмір бази даних	Немає обмеження
Максимальний розмір таблиці	32 Тбайт
Максимальний розмір поля	1 Гбайт
Максимум записів у таблиці	Обмежено розмірами таблиці
Максимум полів у записі	250-1600, в залежності від типів полів
Максимум індексів у таблиці	Немає обмеження

Сильними сторонами PostgreSQL вважаються:

- високопродуктивні на надійні механізми транзакцій та реплікації;
 - система вбудованих мов програмування, що може розширюватися.
- Стандартно підтримуються PL/pgSQL, PL/Perl, PL/Python, PL/Tcl, також є підтримка завантаження модулів розширення на мові C;

						Аркуш
КНТЕУ 121 07-11.БР						14
Изм.	Аркуш	№ докум	Подпись	Дата		

- наслідування;
- можливість індексування геометричних об'єктів та наявність розширення PostGIS, що базується на ній;
- вбудована підтримка слабоструктурованих даних формату JSON з можливістю їх індексації;
- потенціал розширення (можливість створювати нові типи даних, типи індексів, мови програмування, підключати будь-які зовнішні джерела даних)

pgAdmin – найпопулярніша та найбільш функціональна платформа для розробки та адміністрування PostgreSQL. Функціонал pgAdmin:

- Кросплатформність: підтримуються ОС Windows, Linux та macOS
- Створена для багатьох версій PostgreSQL. Версії 9.5 та вище
- Обширна документація: документація додатку pgAdmin та посилання на документацію PostgreSQL
- Декілька моделей розгортання: десктопний режим, використовуючи спеціально побудоване середовище виконання. Серверний режим, підтримуючий декількох користувачів через мережу
- Інструмент запиту із кольоровою підсвіткою синтаксису
- Графічне відображення плану запиту
- Налаштовувач процедурної мови
- Інструмент відмінності для менеджменту відмінностей між схемами
- Менеджмент автоматичного вакууму
- Панель приладів
- Резервне копіювання, відновлення, вакуум та аналіз за потреби
- Агент планування партій задач

						Аркуш
КНТЕУ 121 07-11.БР						15
Изм.	Аркуш	№ докум	Подпись	Дата		



Рис. 1.5. Панель приладів pgAdmin

Також pgAdmin має графічний дизайнер бази даних, за допомогою якого можна легко створювати таблиці та зв'язки між ними

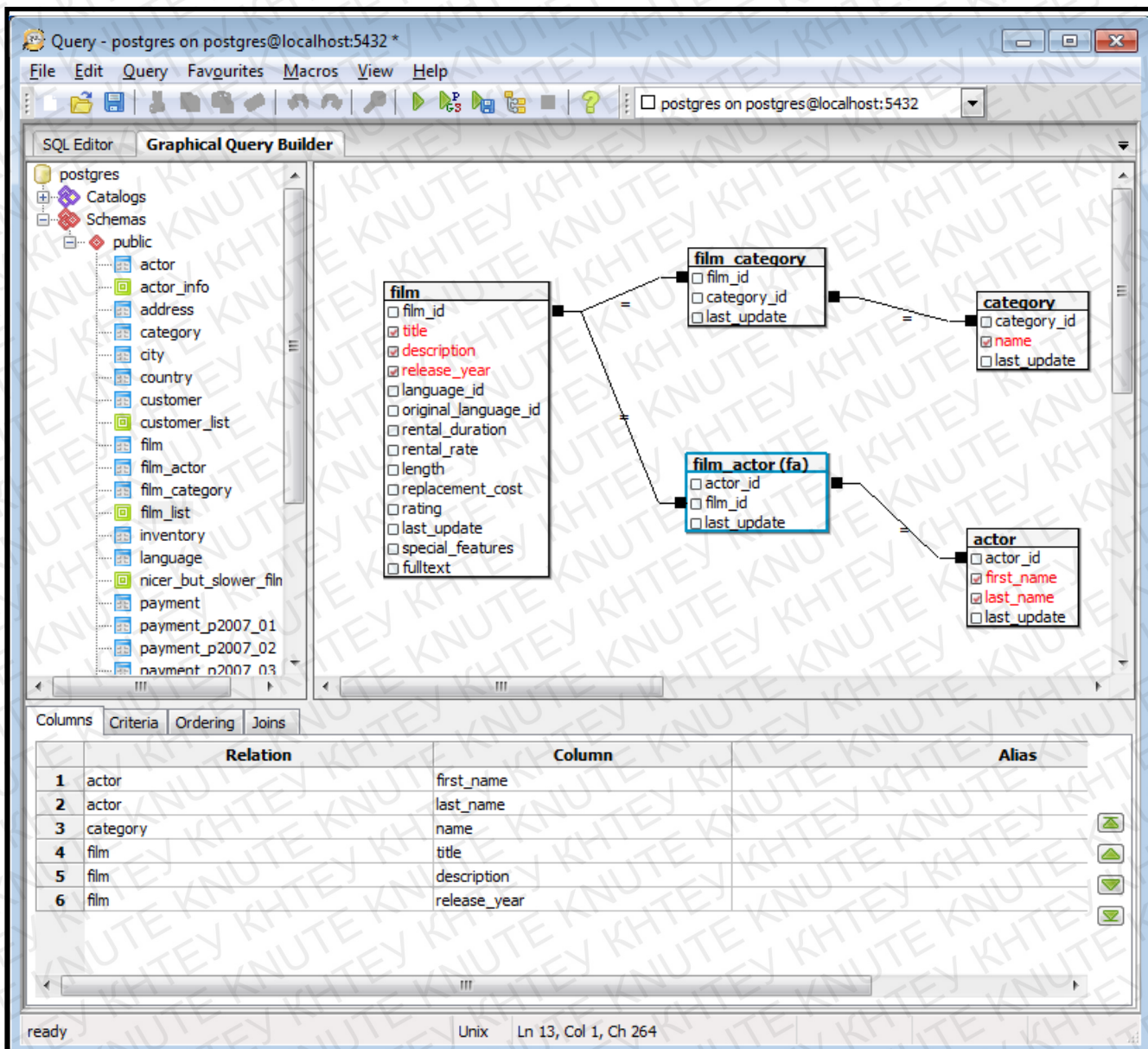


Рис. 1.6. Графічний дизайнер pgAdmin

1.4.5. MySQL

MySQL – вільна система керування базами даних. Розробку та підтримку MySQL здійснює корпорація Oracle, що отримала торгову марку разом із поглиненою Sun Microsystems, котра раніше придбала шведську компанію MySQL AB. Продукт розповсюджується як під GNU General Public License, так і під комерційною ліцензією. Крім цього, розробники створюють функціональність згідно замовлення ліцензованих користувачів. Саме завдяки такому замовленню майже в найбільш ранніх версіях виник механізм реплікації.

Входить до складу серверів WAMP, AppServ, LAMP та в портативні збірки серверів Денвер, XAMPP, VertigoServ. Зазвичай MySQL використовується в якості серверу, до якого звертаються локальні або віддалені клієнти, проте у дистрибутив входить бібліотека внутрішнього серверу, що дозволяє включати MySQL в автономні програми.

Гнучкість СКБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть обрати як таблиці типу MyISAM, підтримуючі повнотекстовий пошук, так і таблиці InnoDB, підтримуючі транзакції на рівні окремих записів. Більш того, СКБД MySQL поставляється із спеціальним типом таблиць EXAMPLE, що демонструють принципи створення нових типів таблиць. Завдяки відкритій архітектурі та GPL-ліцензуванню, у СКБД MySQL постійно з'являються нові типи таблиць.

MySQL портована на велику кількість платформ, серед яких Linux, BSD, Windows. Також існує порт MySQL до OpenVMS. На офіційному сайті СКБД для вільного завантаження надаються не лише вихідні коди, а й скомпільовані та оптимізовані під конкретні операційні системи готові виконувати модулі СКБД MySQL.

MySQL має API та конектори для мов Delphi, C, C++, Java, Lisp, Perl, PHP, Python, Ruby, Smalltalk, Tcl, Lua та ін., бібліотеки для мов платформи .NET, а також забезпечує підтримку для ODBC за допомогою ODBC-драйверу MyODBC. MyODBC представляє собою драйвер ODBC рівня 0 (із деякими можливостями рівнів 1 та 2) для підключення сумісного з ODBC додатку до MySQL. MyODBC працює на всіх системах Microsoft Windows та на більшості платформ Unix.

MySQL Workbench – інструмент візуального дизайну баз даних, в котру інтегровано SQL розробку, адміністрування, створення та підтримку в цілому інтегрованому середовищі розробки для системи баз даних MySQL. Функціонал MySQL Workbench:

- Підтримка повного скриптування за допомогою Python та Lua
- Підтримка плагінів
- Редактор SQL

						Аркуш
						18
Изм.	Аркуш	№ докум	Подпись	Дата	КНТЕУ 121 07-11.БР	

- Моделювання даних
- Адміністрування баз даних
- Моніторинг продуктивності
- Міграція баз даних

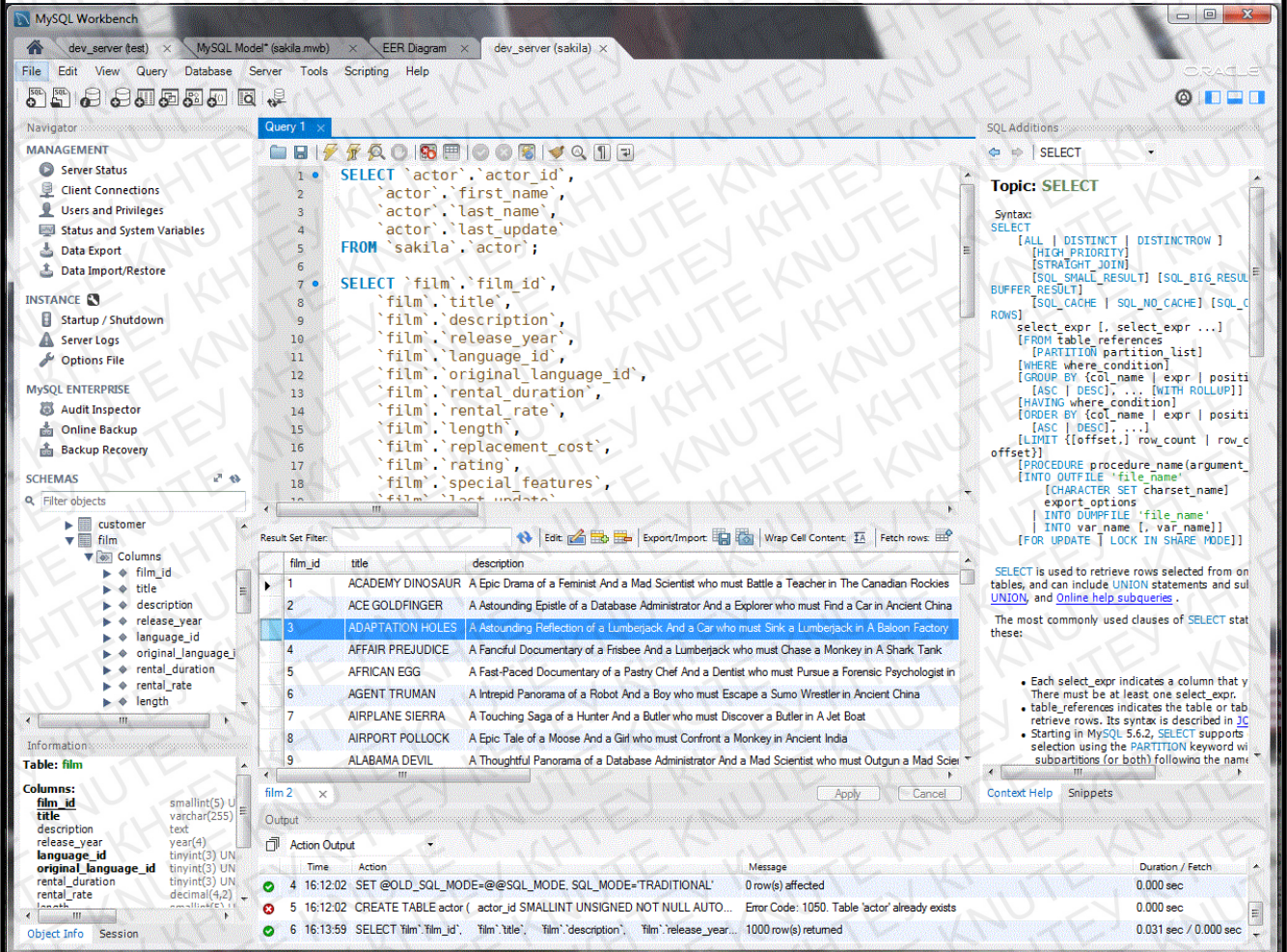


Рис. 1.7. Редактор запиту

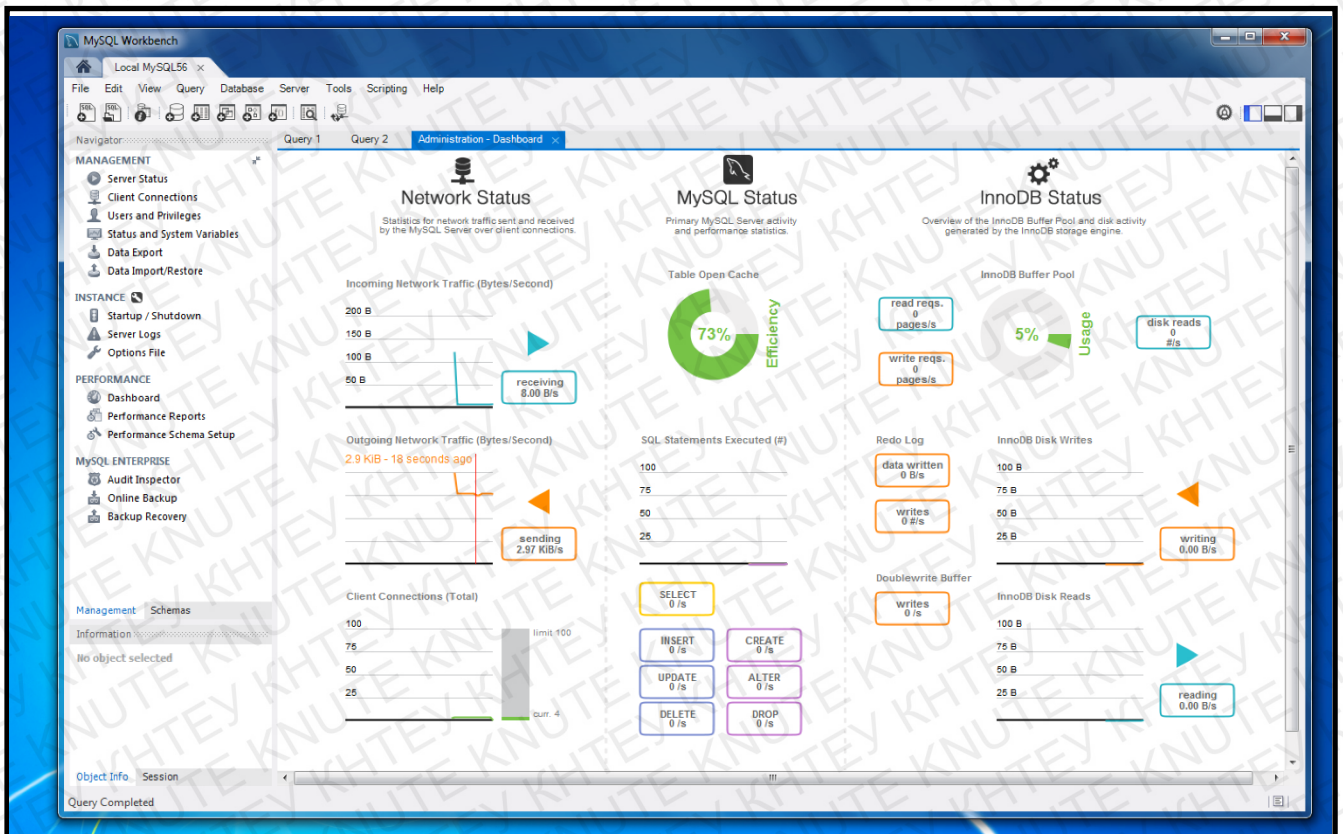


Рис. 1.8. Панель приладів

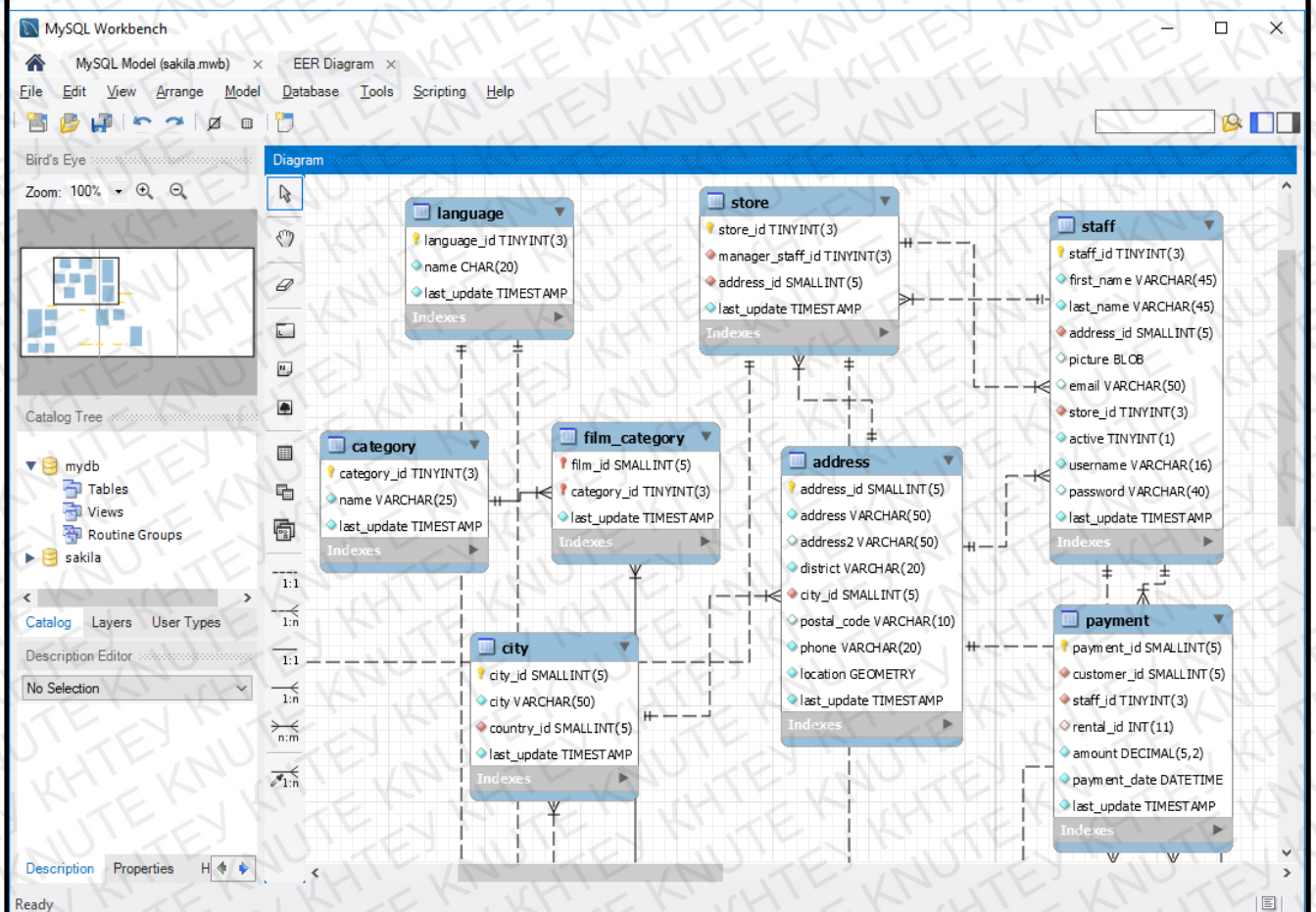


Рис. 1.9. Графічний дизайнер бази даних

1.4.6. MariaDB

MariaDB – відгалуження від системи керування базами даних MySQL, що розроблюється спільнотою під ліцензією GNU GPL. Розробку та підтримку MariaDB здійснює компанія MariaDB Corporation Ab та фонд MariaDB Foundation.

Поштовхом до створення стала необхідність забезпечення вільного статусу СКБД, на противагу політиці ліцензування MySQL компанією Oracle. Засновникам проекту виступили початкові розробники MySQL. Система ліцензування MariaDB зобов'язує учасників, що хочуть додати свій код в основну гілку СКБД, обмінюватися своїми авторськими правами із MariaDB Foundation для охорони ліцензії та можливості створювати критичні виправлення для MySQL.

MariaDB має намір підтримувати велику сумісність з MySQL, забезпечуючи точну відповідність API та командами MySQL. В склад MariaDB включені підсистеми зберігання даних XtraDB для можливості заміни InnoDB, як основної підсистеми зберігання. Також включені підсистеми Aria, PBXT та FederateX.

API та протоколи у MariaDB відповідають тим, що використовуються у MySQL, а це означає, що всі бібліотеки та додатки, котрі працюють у MySQL, повинні працювати і з MariaDB. В зв'язку з цим розробники Fedora у 19 версії перейшли з MySQL на MariaDB через побоювання, що Oracle зробить із MySQL більш закритий проект.

1.5. Вимоги до системи

Система повинна бути зручною, корисною, надійною.

1.6. Висновки до розділу 1

В даному розділі автором було розглянуто технічне завдання, можливість реалізації та вимоги, після чого був зроблений на пояснений вибір мови програмування та бази даних.

					КНТЕУ 121 07-11.БР	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		21

РОЗДІЛ 2.

АНАЛІЗ ПРЕДМЕТУ ДОСЛІДЖЕННЯ ТА ЙОГО ХАРАКТЕРИСТИКА

2.1. Опис програми «PyCharm»

PyCharm – це інтегроване середовище розробки, що використовується для програмування, а саме для мови програмування Python. Її розробила чеська компанія JetBrains. Вона надає аналіз коду, графічне відлагодження, інтегрований юніт тестер, інтеграцію із системами керування версіями та підтримує веб-розробку із Django.

PyCharm крос-платформна, із версіями для Windows, macOS та Linux. Безкоштовна версія поширюється за ліцензією Apache, також є професійна версія із додатковим функціоналом, має власницьку ліцензію.

До функціоналу PyCharm входять:

- Кодовий асистент та аналіз, з заповненням коду, виділенням синтаксису та помилок, інтеграцією із linter та швидкими виправленнями
- Навігація по проекту та коду: спеціалізовані відображення проекту, відображення файлової структури та швидкі переходи між файлами, класами, методами та використаннями
- Рефакторинг Python: включає в себе перейменування, видобування методу, оголошення змінної, оголошення константи, зміщення вгору та вниз та багато інших
- Підтримка веб фреймворків: Django, web2py та Flask (лише професійна версія)
- Інтегрований налагоджувач Python
- Інтегроване середовище тестування із порядковим покриттям
- Python розробка із Google App Engine (лише професійна версія)

					<i>КНТЕУ 121 07-11.БР</i>			
					<i>Система адміністрування віртуальних спільнот на платформі Telegram Bot</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.			<i>Аналіз предмету дослідження та його характеристика</i>	Р2	22	44
Керівник		Жирова Т.О.						
Гарант		Цензура М.О.						
Розроб		Мілевський Д.В.			Факультет інформаційних технологій, 4 курс, 7 група			

- Інтеграція систем керування версіями: Mercurial, Git, Subversion, Perforce, CVS

Плагіни – PyCharm надає API щоб розробники могли писати власні плагіни для розширення функціоналу PyCharm. Деякі плагіни із інших середовищ розробки JetBrains сумісні із PyCharm, а також понад 1000 інших плагінів.

2.2. Конкурентність програми

Автором було виявлено декілька інших систем адміністрування

2.2.1. ControllerBot

ControllerBot – це бот для власників каналів. Він дозволяє:

- Написання постів із розмітками Markdown/HTML
- Додавати кнопки реакцій (смайли, текст або обидва варіанти)
- Додавати кнопку для відкриття коментарів посту
- Відкладення постів
- Додавати таймер для самознищення тимчасових постів
- Переглядати статистику каналу



Рис. 2.1. Приклад постів із кнопками реакцій

ControllerBot є непоганим ботом, проте він створений лише для того, щоб додати можливості залишення коментарів та реакцій для постів. До того ж, такі

					Аркуш
					23
Изм.	Аркуш	№ докум	Подпись	Дата	

функції як написання постів із розмітками та відкладення постів стали непотрібними, бо Telegram надає такий функціонал усім.

2.2.2. LivegramBot

LivegramBot – бот-конструктор для збирання відгуків.

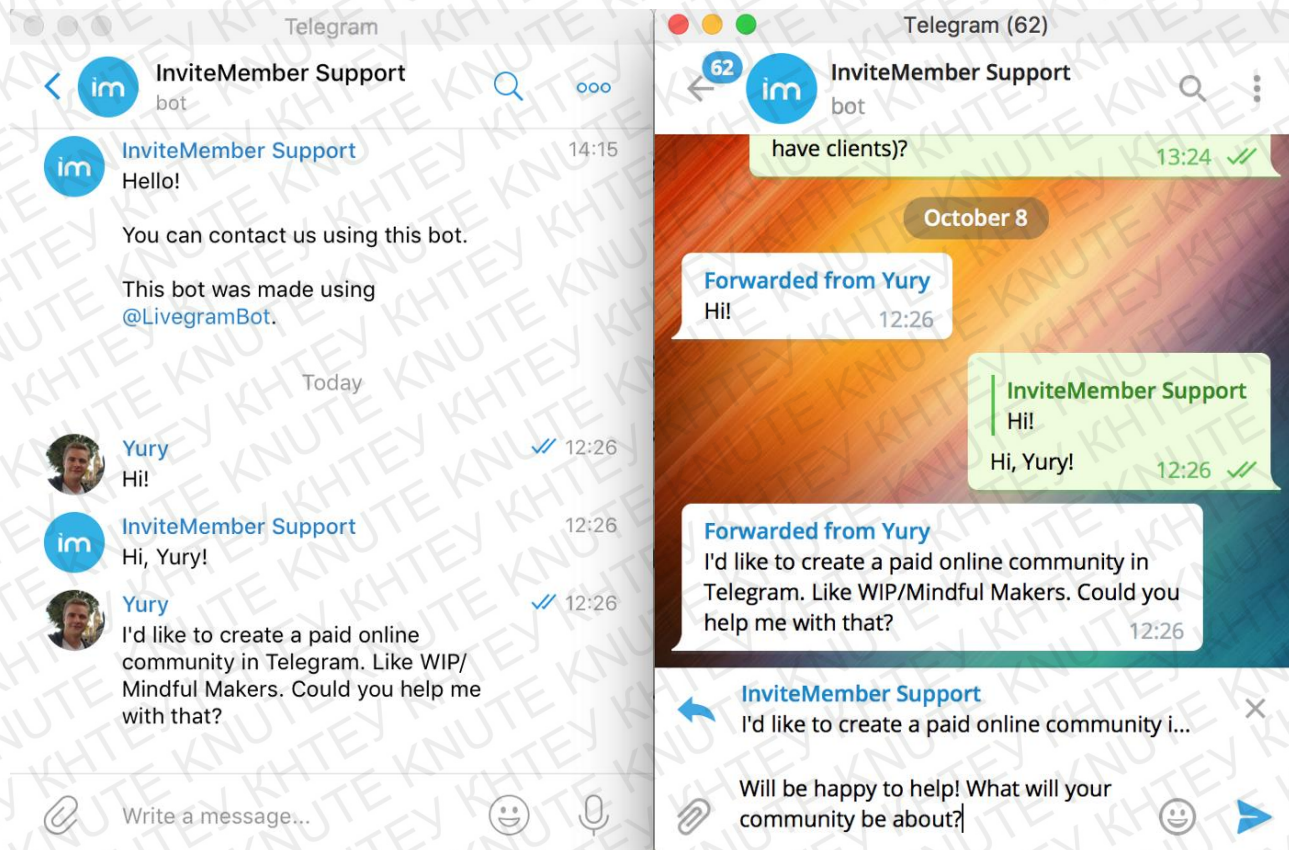


Рис. 2.2. взаємодія з LivegramBot

LivegramBot є дуже простим та має всього лише одну функцію – конструювати збирачів відгуків.

2.2.3. VoteBot

VoteBot – бот для створення голосувань, котрі можна пересилати друзям, в канали або групи. Він був дуже корисним, проте Telegram пізніше вбудували підтримку голосувань у клієнти і тепер для цього не потрібен бот.

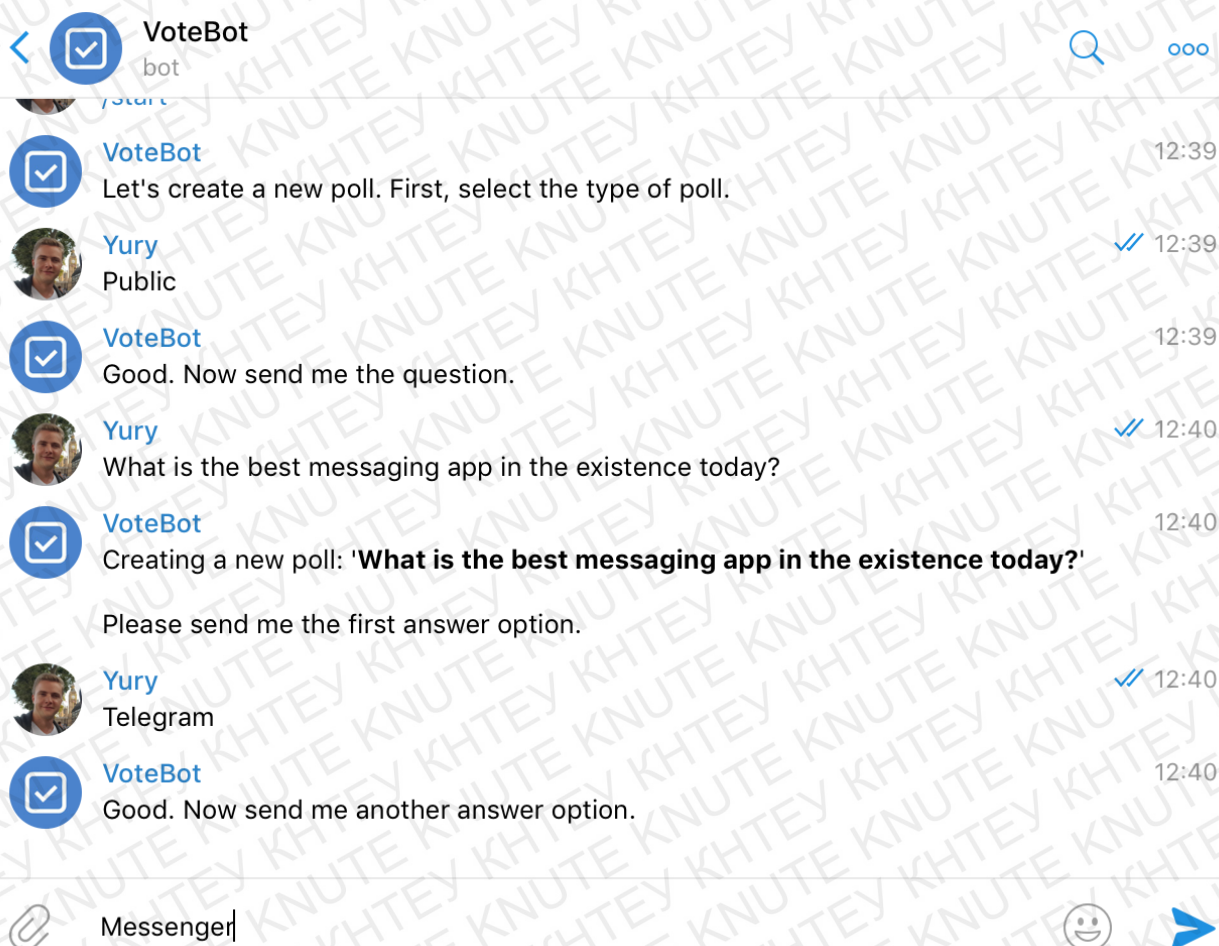


Рис. 2.3. Створення голосування у VoteBot

2.3. Висновки до розділу 2

Із наданої інформації можна зробити висновок, що система є конкурентоспроможною, адже вона надає можливості адміністрування спільнот та налаштування автоматичних дій, таких як переклад, фільтрація флуду, видалення неактивних користувачів.

					Аркуш
					25
Изм.	Аркуш	№ докум	Подпись	Дата	

РОЗДІЛ 3.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АДМІНІСТРУВАННЯ ВІРТУАЛЬНИХ СПІЛЬНОТ

3.1. Програмне забезпечення «ruGroupMgr»

Програмний засіб «ruGroupMgr» призначений для адміністрування віртуальних спільнот, а саме:

- Такі адміністративні дії, як вилучення та блокування користувачів
- Привітання нових учасників
- Керування попередженнями користувачів (порушення правил спільноти)
- Тимчасове обмеження нових користувачів для того, щоб вони ознайомилися із правилами спільноти
- Обмеження мови спільноти
- Попередження учасників спільноти при появі нового, що порушував правила інших спільнот
- Автоматичне вилучення неактивних користувачів

					<i>КНТЕУ 121 07-11.БР</i>			
					<i>Система адміністрування віртуальних спільнот на платформі Telegram Bot</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		РЗ	26	44
Зав. каф.		Криворучко О.В.			<i>Розробка програмного забезпечення системи адміністрування віртуальних спільнот</i>	Факультет інформаційних технологій, 4 курс, 7 група		
Керівник		Жирова Т.О.						
Гарант		Цензура М.О.						
Розроб		Мілевський Д.В.						

3.2. Діаграми

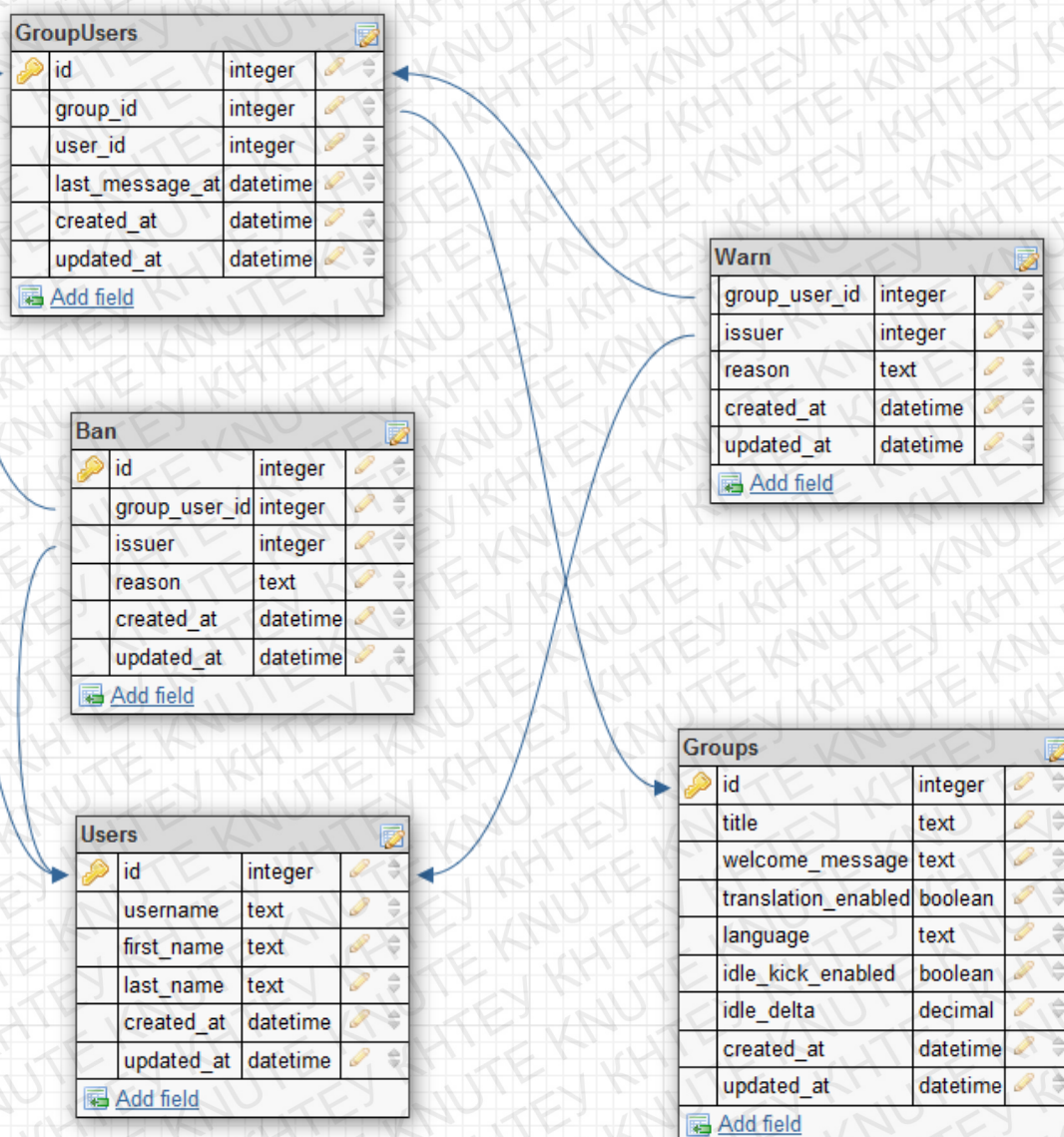


Рис. 3.1. Модель бази даних

Програма «руGroupMgr» є модульною. Обробники, проміжні програми та сервіси було розділено в першу чергу на пакети Python, а пакети – на файли.

В програмі «руGroupMgr» присутні дві проміжні програми: «ACLMiddleware» та «AntiFloodMiddleware».

ACLMiddleware автоматично зберігає усі нові групи та користувачів у базу даних. Якщо користувач вже існував у базі даних, оновлюється дата його останньої активності у групі, де його було побачено. Діаграму станів ACLMiddleware можна побачити на рис. 3.2.

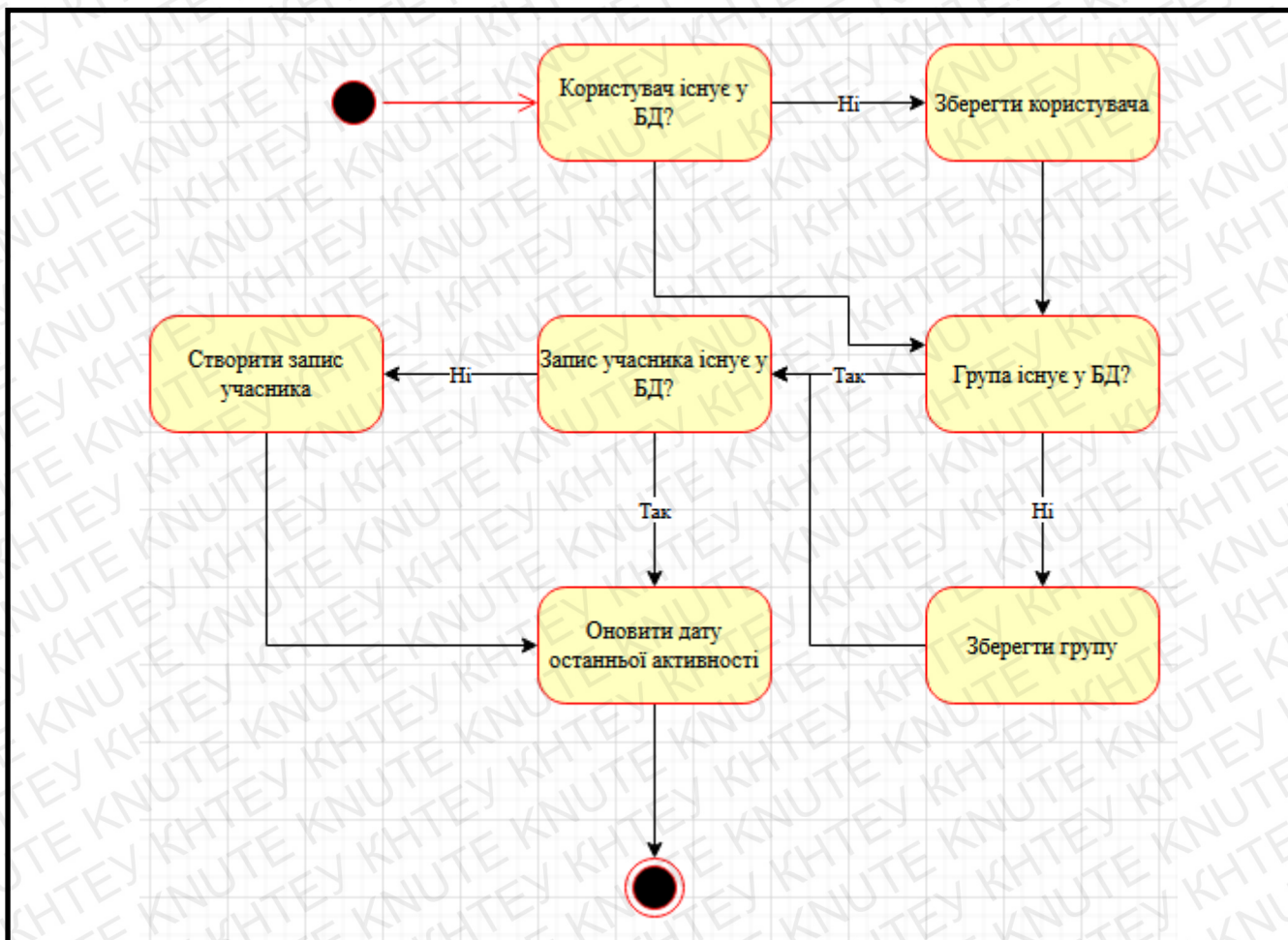


Рис. 3.2. Діаграма станів ACLMiddleware

«AntiFloodMiddleware» служить для запобігання флуду командами боту. При отриманні 10 або більше повідомлень у групі протягом однієї хвилини, обробка оновлення скасовується і бот не буде відповідати.

«руGroupMgr» має багато обробників, вони були розділені на такі групи:

- Базові
- Допомога
- Вилучення неактивних
- Нові учасники
- Переклади
- Привітання

Базові обробники відповідають за такі команди, як вилучення, попередження, блокування, скасування поточної дії

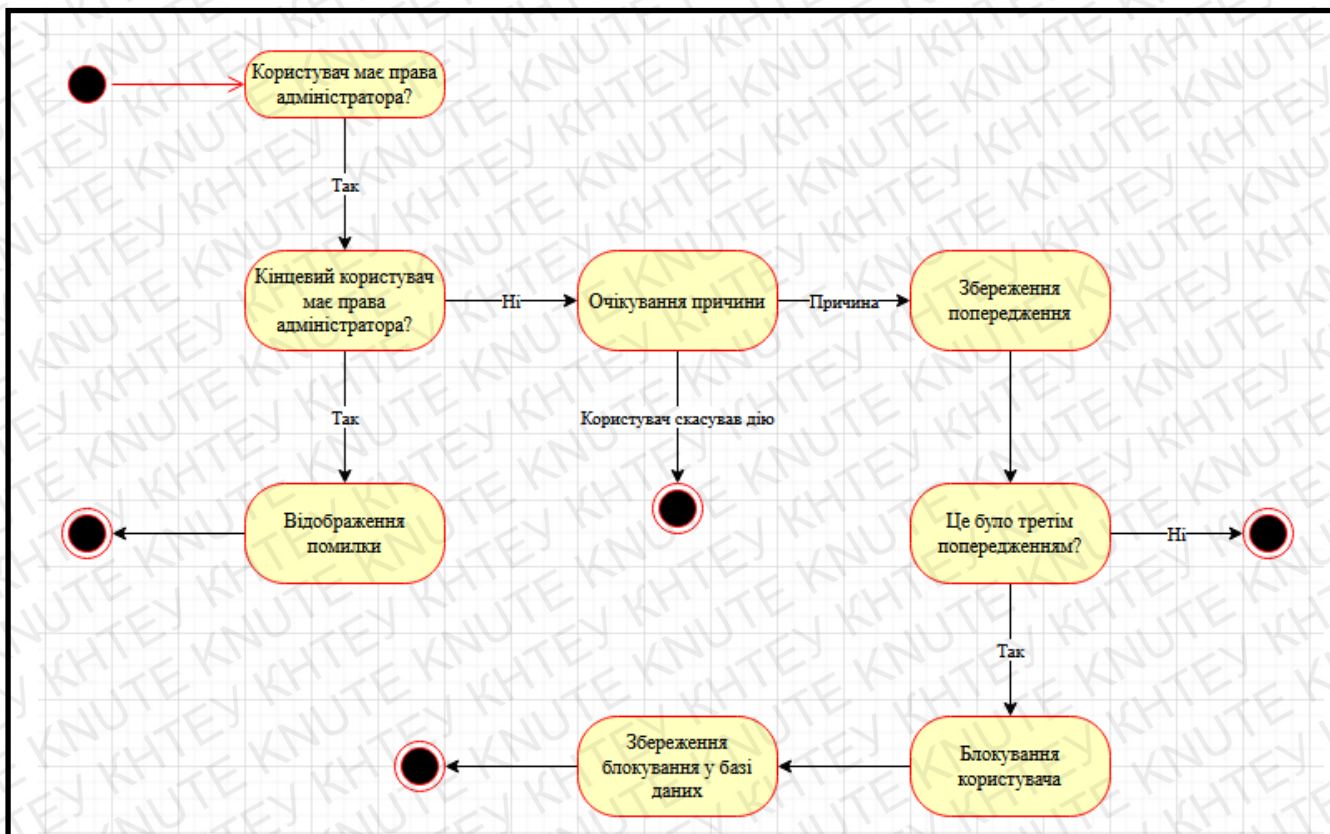


Рис. 3.3. Діаграма станів при попередженні

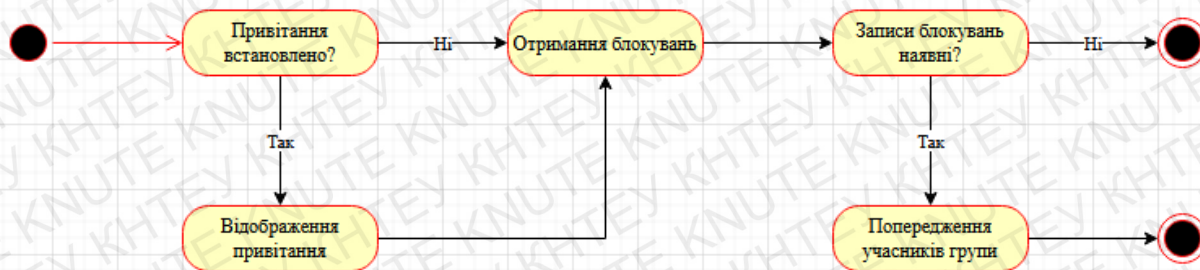


Рис. 3.4. Діаграма станів при появі нового учасника

3.3. Інтерфейс користувача

Вся взаємодія користувача із «ruGroupMgr» проходить через Telegram. Для максимальної зручності було використано не тільки звичайні текстові повідомлення, а й інші, надані платформою Telegram Bot, інструменти, такі як вбудовані клавіатури, спеціальні клавіатури та команди.

При доданні бота до групи він привітається та порадить використати команду /help, що відобразить доступні команди та їх опис.

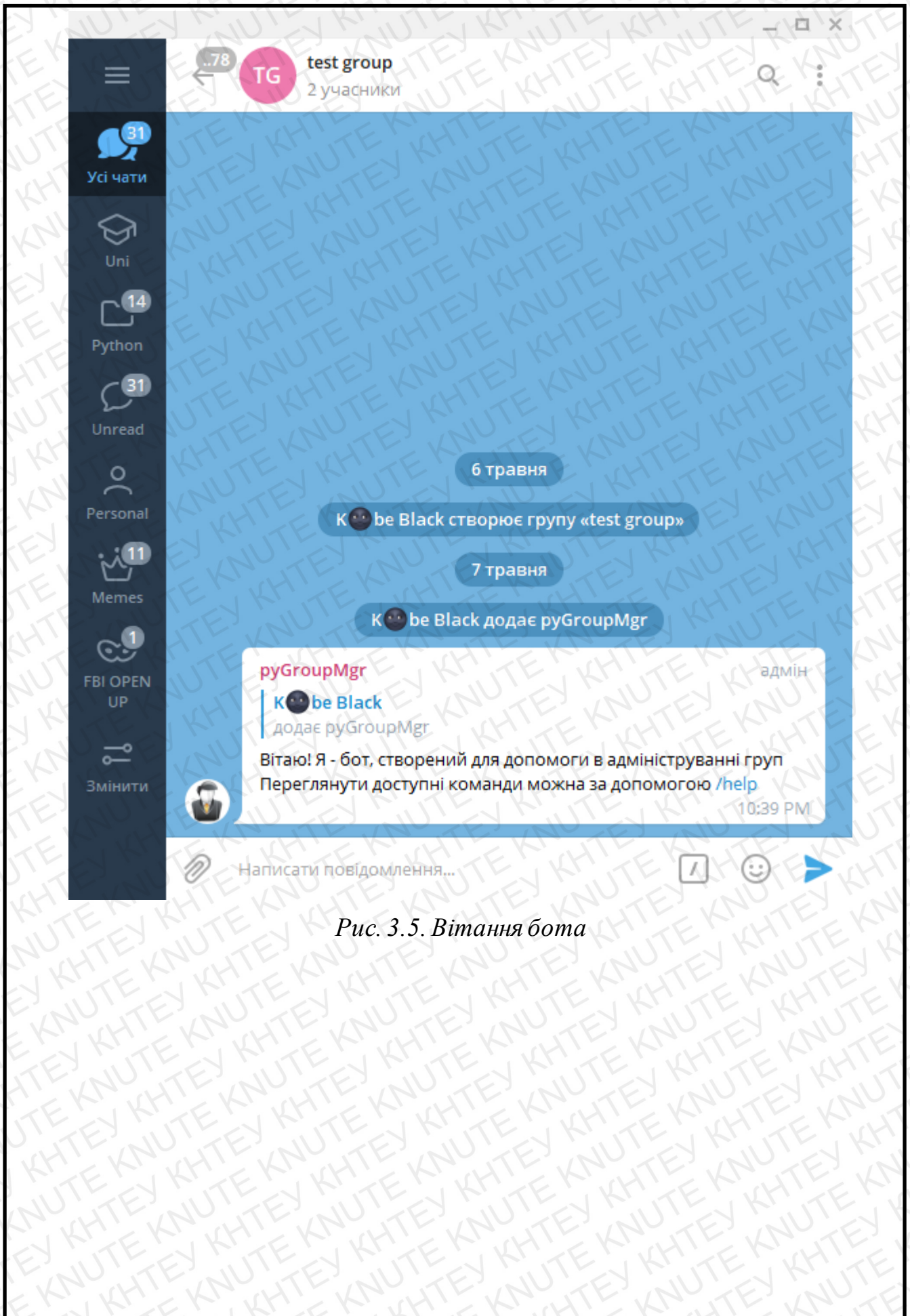


Рис. 3.5. Вітання бота

					Аркуш
					30
Изм.	Аркуш	№ докум	Подпись	Дата	

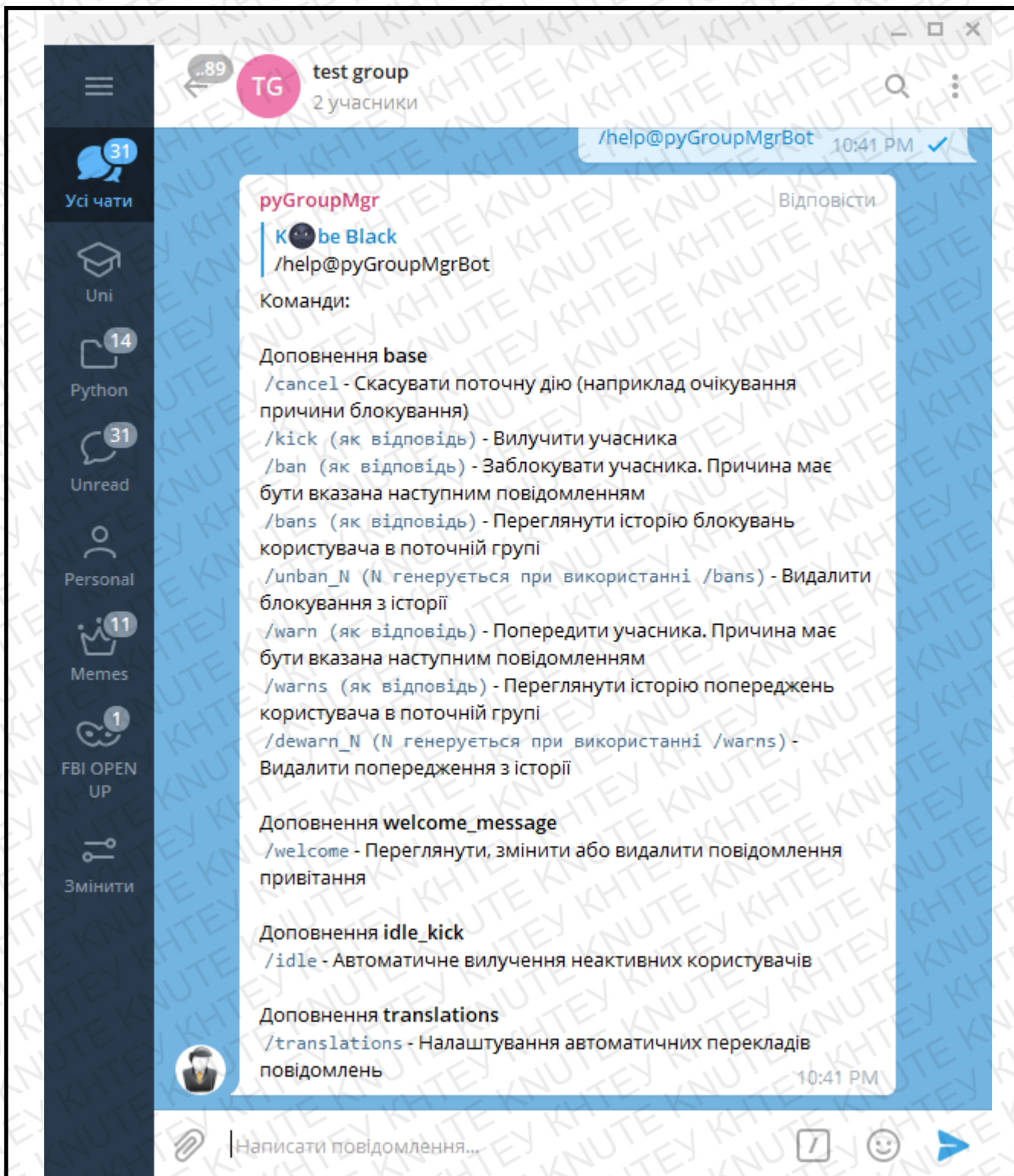


Рис. 3.6. Команди бота

Команди `/kick` (вилучення), `/ban` (блокування) та `/warn` (попередження) потрібно відправляти як відповідь на повідомлення іншого користувача. Таким чином бот одразу знає, якого користувача слід покарати. Команда вилучення діє миттєво та просто прибирає учасника з групи, в той час як команди блокування та попередження запитують причину.

					Аркуш
					31
Изм.	Аркуш	№ докум	Подпись	Дата	

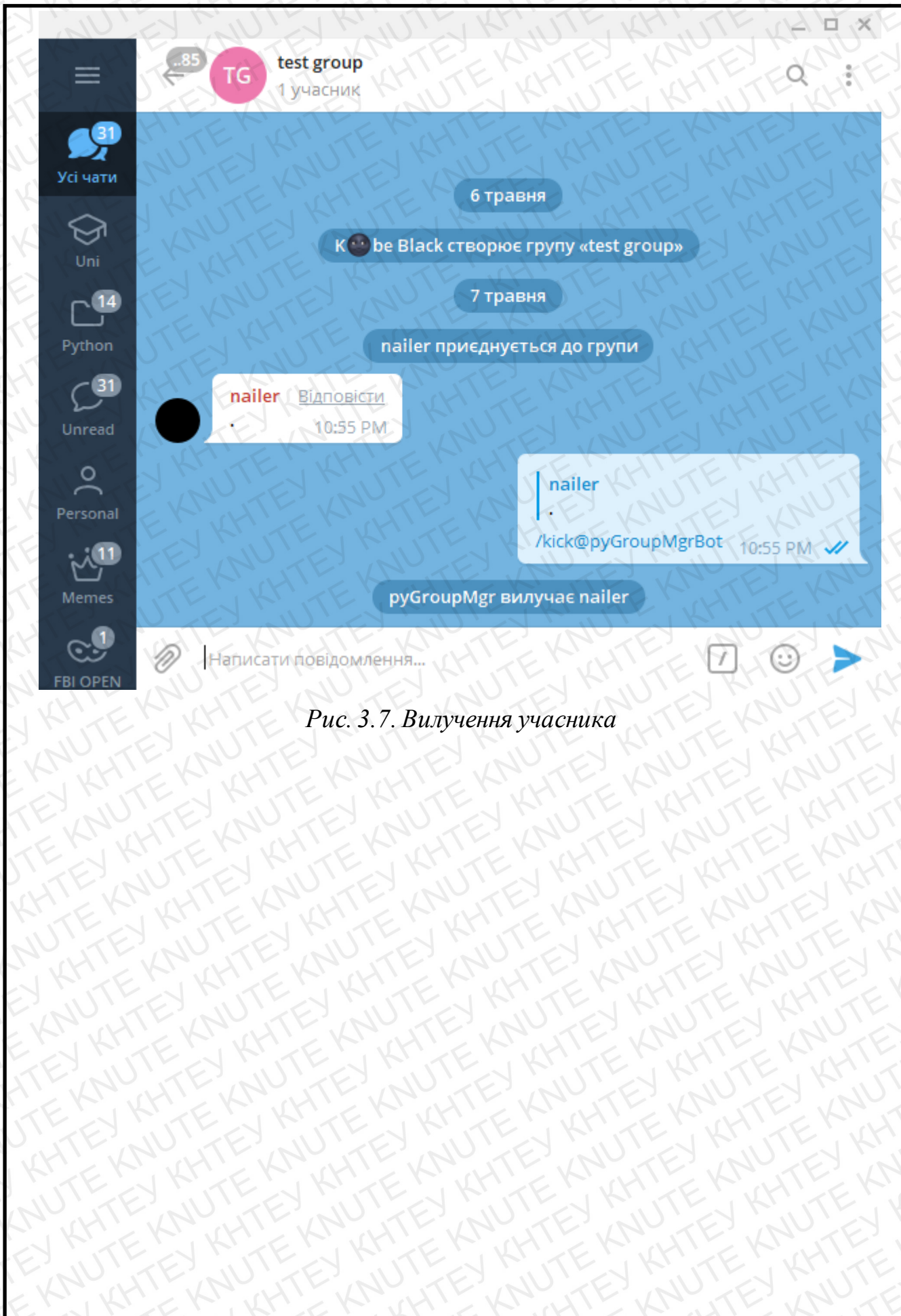


Рис. 3.7. Вилучення учасника

					Аркуш
					32
Изм.	Аркуш	№ докум	Подпись	Дата	

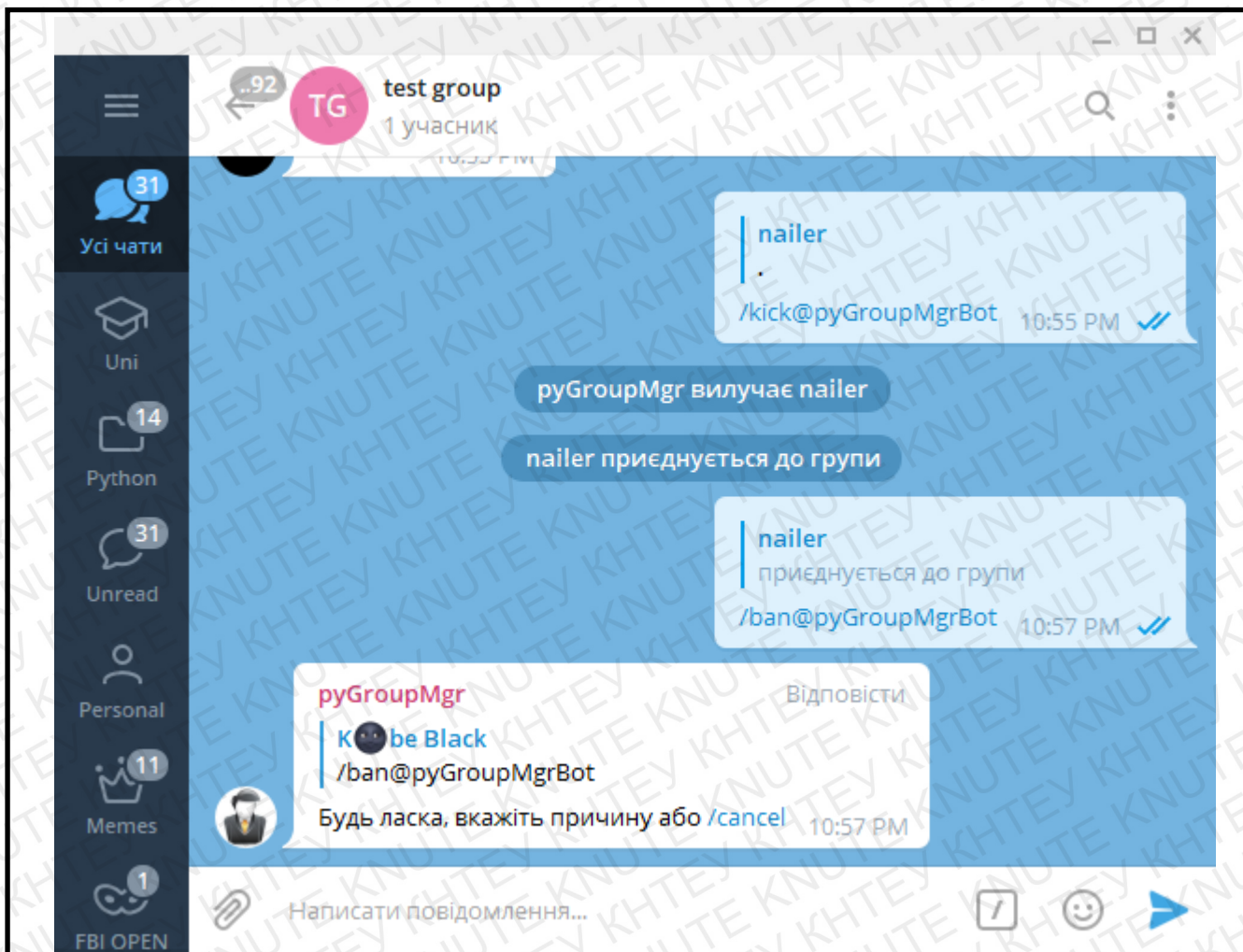


Рис. 3.8. Запит причини блокування

Після отримання причини, бот блокує користувача. Запис про це блокування буде зберігатися у базі даних, а також, бот про нього нагадає, коли користувач повернеться в ту саму (якщо користувача розблокували) або іншу групу, що можна побачити на рис. 3.9.

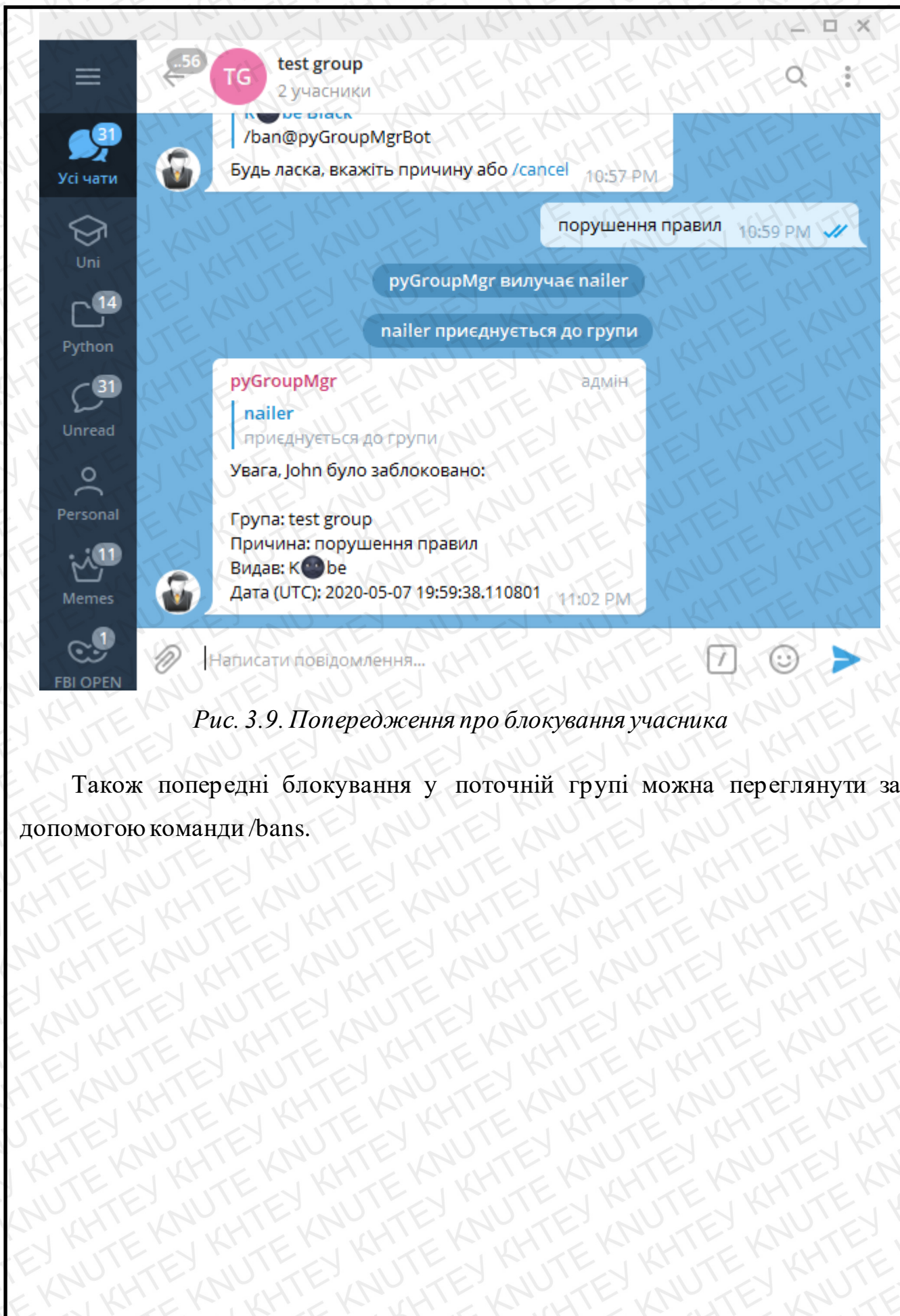


Рис. 3.9. Попередження про блокування учасника

Також попередні блокування у поточній групі можна переглянути за допомогою команди /bans.

					Аркуш
					34
Изм.	Аркуш	№ докум	Подпись	Дата	

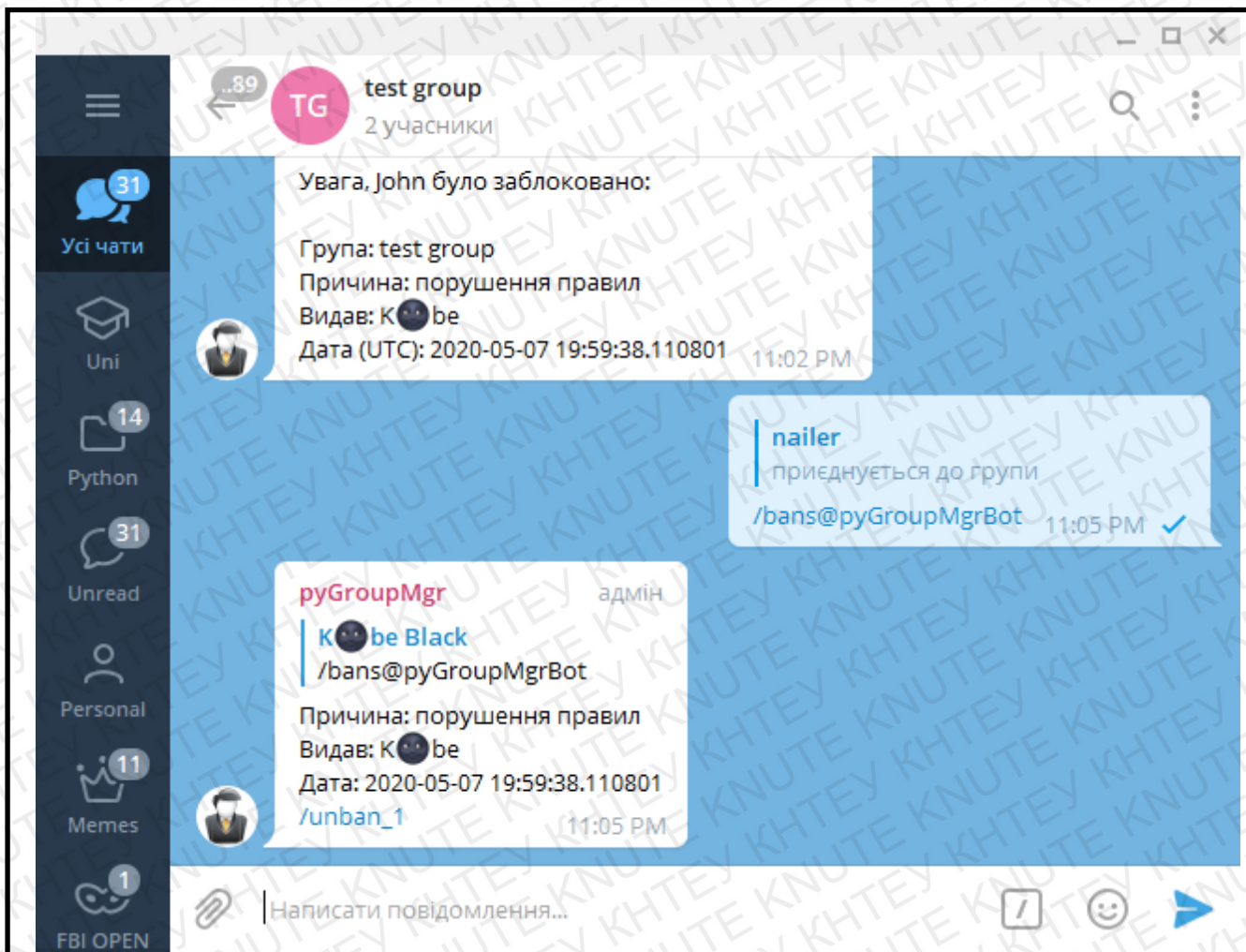


Рис. 3.10. Перегляд попередніх блокувань

Як видно на рис. 3.10. під кожним блокуванням можна побачити команду `/unban` з порядковим номером. При її використанні одним із адміністраторів групи, запис про блокування учасника видалиться із бази даних. Такий варіант взаємодії із користувачем було обрано для зручності, адже клієнти месенджеру Telegram одразу відправляють команду при її натисканні. Це зручніше, ніж змушувати користувача шукати, наприклад, потрібну кнопку для видалення, адже блокування можуть бути однотипними.

Попередження працюють за схожим принципом, проте бот автоматично заблокує користувача, як тільки той отримає третє попередження. При цьому ботом буде вказана причина «Було отримано три попередження».

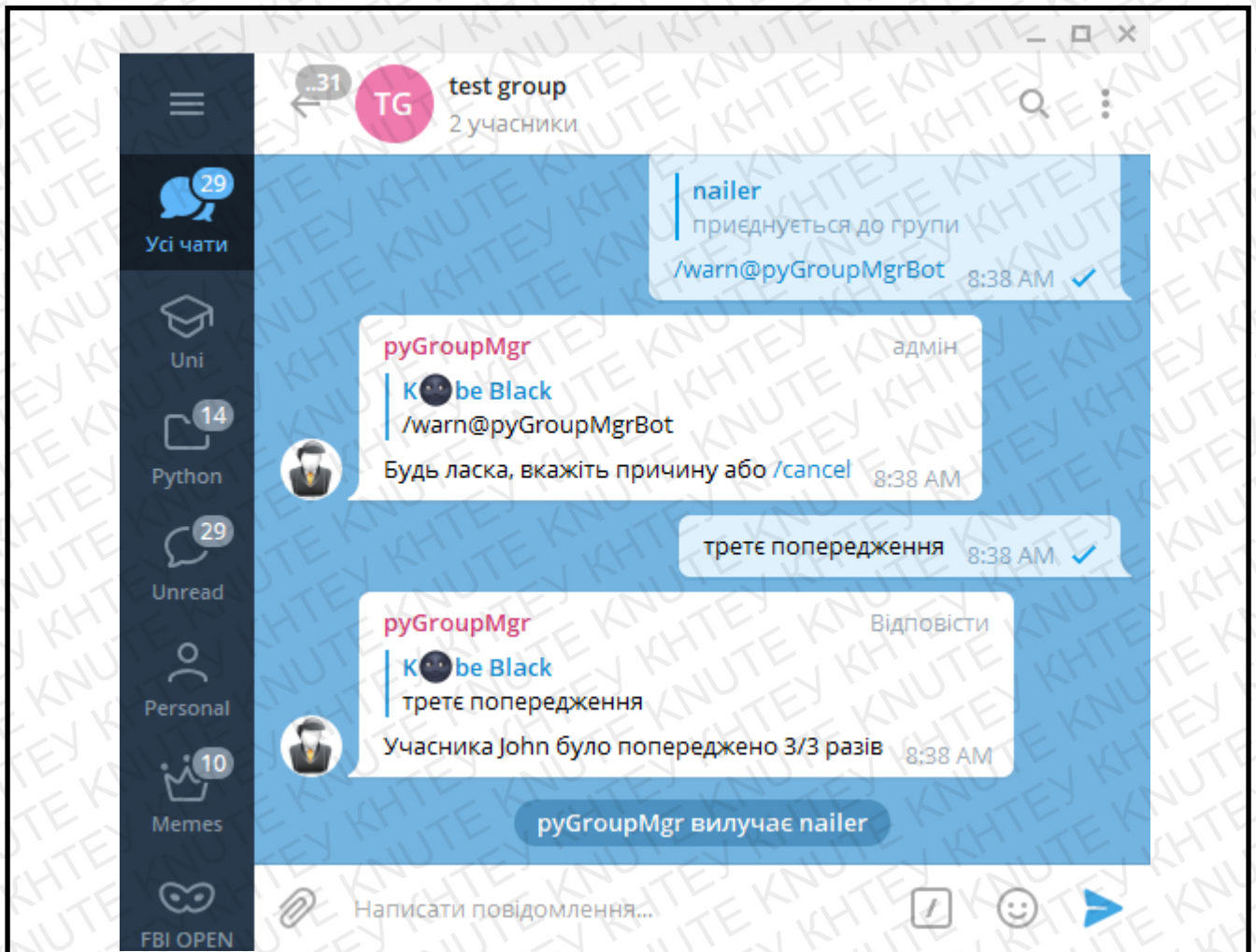


Рис. 3.11. Отримання трьох попереджень

Є можливість встановити привітання, що бот буде відправляти новим учасникам групи. Для керування цією функцією адміністратори мають використовувати команду /welcome. Бот відобразить встановлене повідомлення (якщо є) та кнопки встановлення нового та видалення (якщо є)

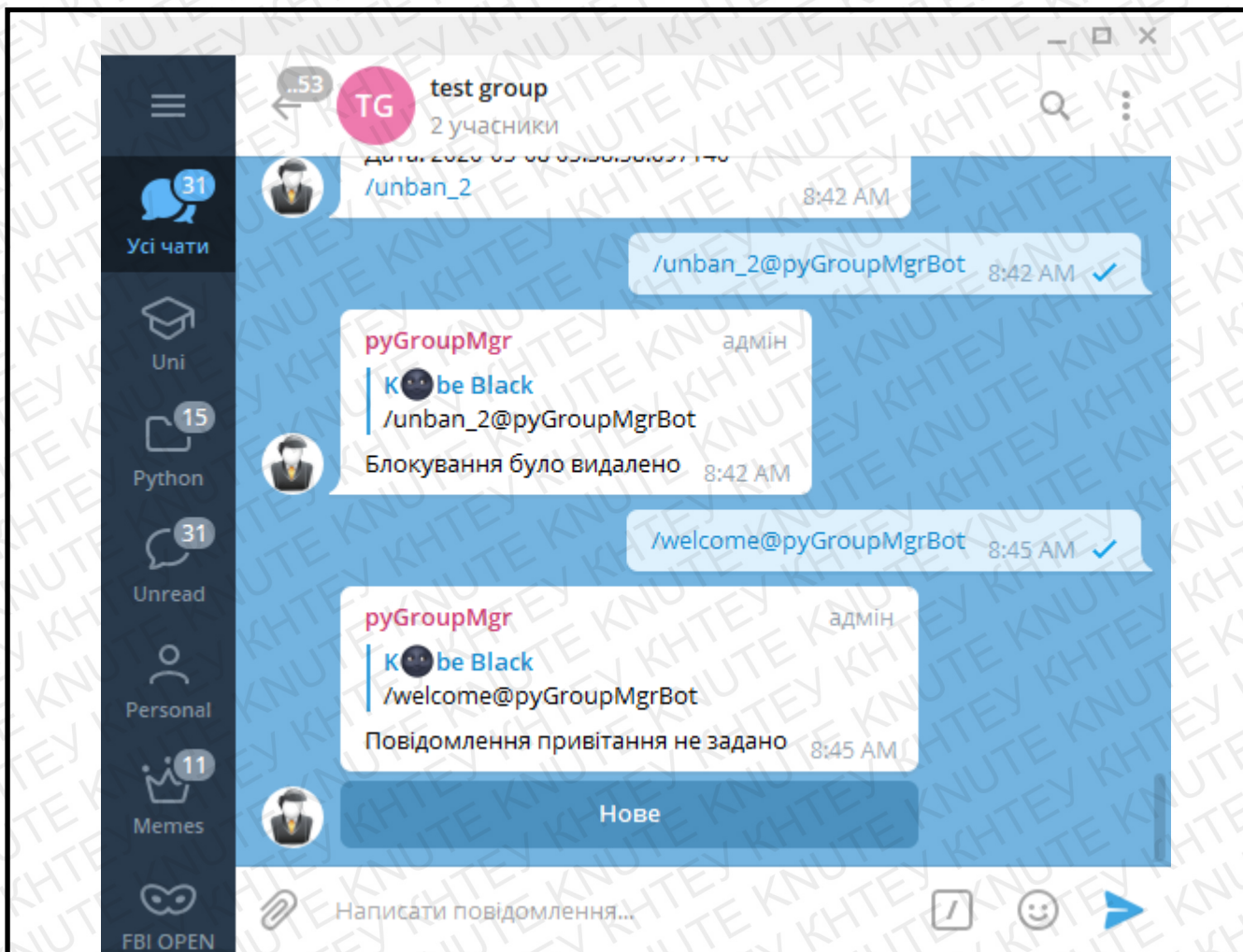


Рис. 3.12. Керування привітанням

					Аркуш
					37
Ізм.	Аркуш	№ докум	Підпись	Дата	

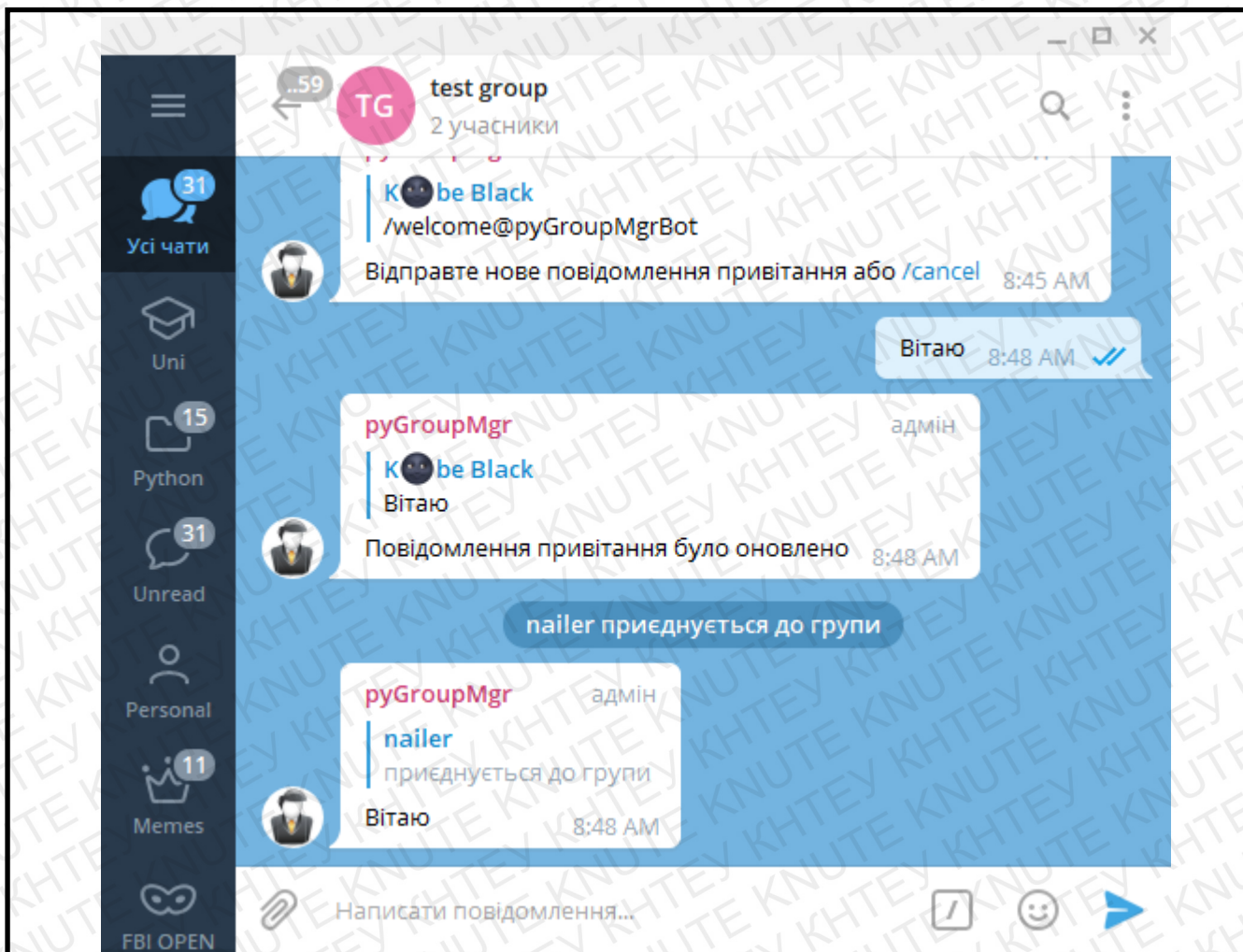


Рис. 3.13 Встановлення повідомлення та привітання учасника

Бот також має функцію автоматичних перекладів повідомлень. Для керування нею використовується команда `/translations`. Після її використання можна побачити наразі встановлену мову групи та опції встановлення нової та вимкнення функції (якщо наразі активна).

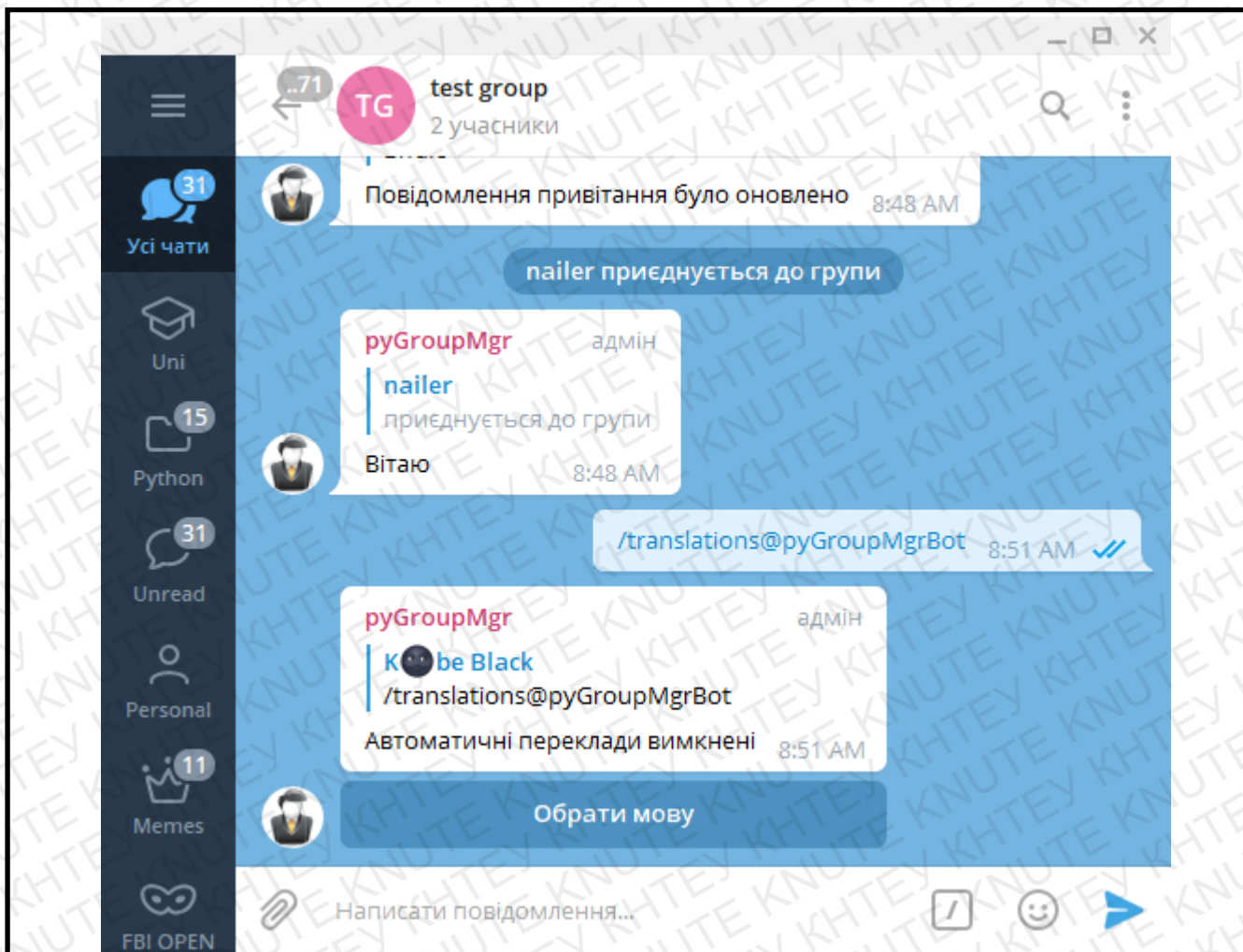


Рис. 3.14. Керування перекладами

Після натискання на кнопку вибору, бот відобразить клавіатуру з великою кількістю мов, що підтримуються.

						Аркуш
						39
Ізм.	Аркуш	№ докум	Підпись	Дата		

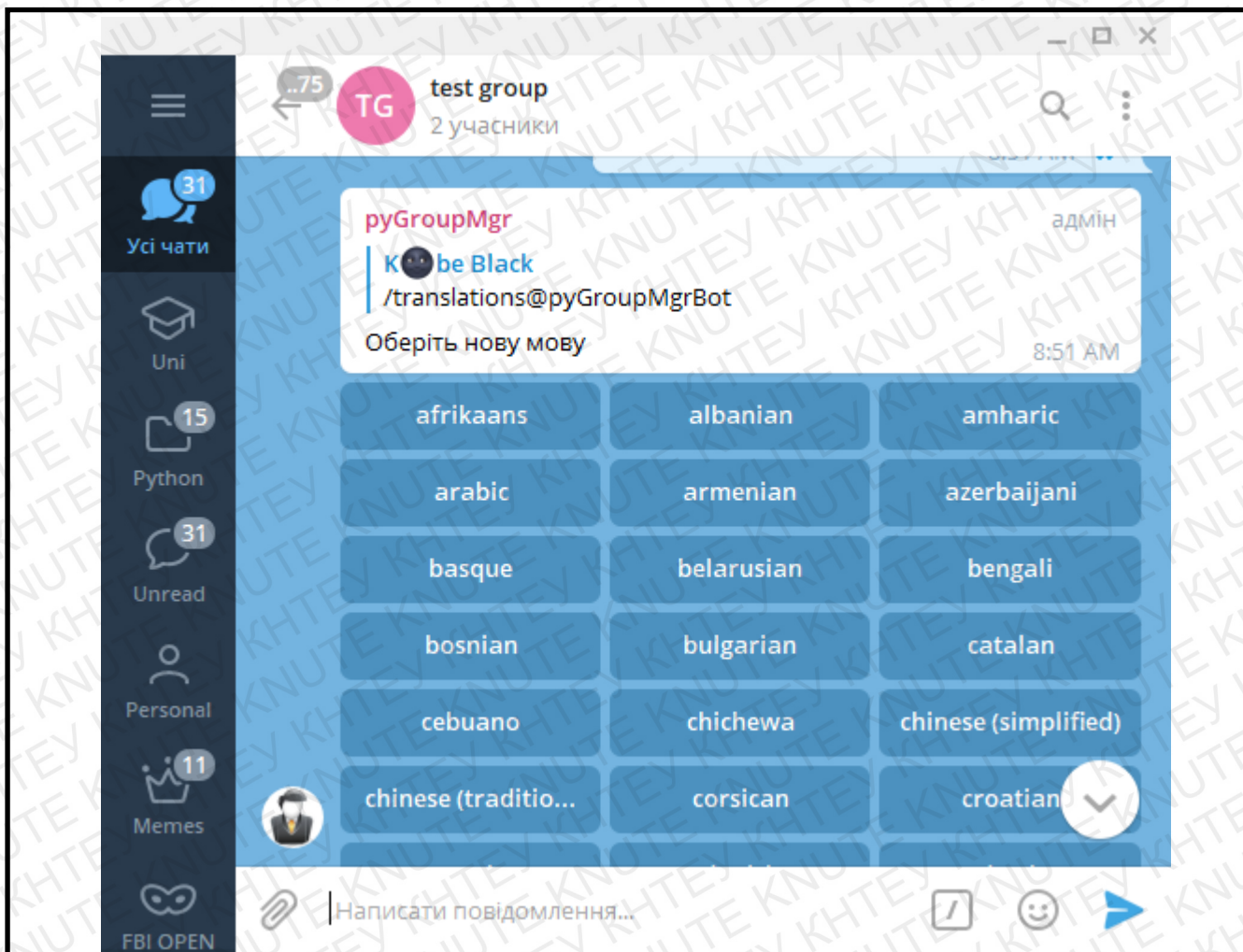


Рис. 3.15. Меню мов

Після вибору мови бот слідкуватиме за повідомленнями, що відправляються в цій групі. Якщо мова повідомлення не співпадає із заданою, бот видалить відправлене користувачем повідомлення та відправить нове, перекладене на мову, що встановлена для групи, зазначивши хто відправляв та оригінальну мову повідомлення

						Аркуш
						40
Изм.	Аркуш	№ докум	Подпись	Дата		

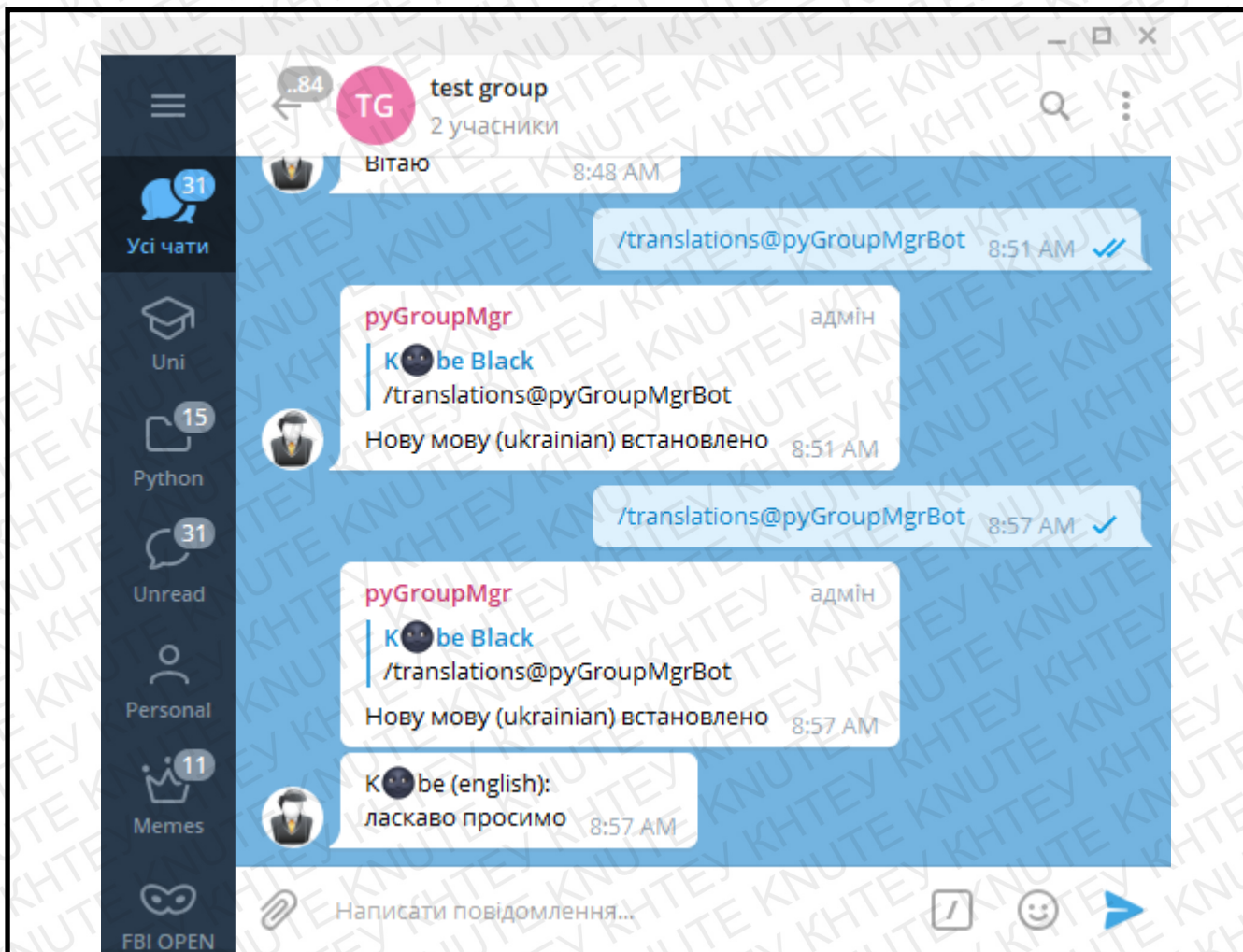


Рис. 3.16. Приклад автоматичного перекладу

3.4. Висновки до розділу 3

В даному розділі було розглянуто інтерфейс програми, усі можливості її можливості та було наведено приклади роботи. Також було наведено діаграму бази даних та діаграми основних алгоритмів програми.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В даному проекті було розглянуто різні рішення поставленої задачі. Також було відображено інтерфейс користувача

Була розглянута конкурентоспроможність порівняння з альтернативними програмами для адміністрування.

Рекомендацією автора буде визначення потреб кінцевого користувача, адже адміністрування – досить обширне поняття у такому месенджері, як Telegram. Програми конкуренти не є гіршими або кращими – вони просто знаходяться в іншій категорії адміністрування. Наприклад якщо користувач є власником каналів, йому буде потрібна програма із такими функціями, як менеджмент підписників, збирання відгуків та коментарів від них. В той час як власнику великої групи, особливо публічної, буде потрібен бот такого типу, як було розроблено в даній роботі – для контролю над групою.

					<i>КНТЕУ 121 07-11.БР</i>			
					<i>Система адміністрування віртуальних спільнот на платформі Telegram Bot</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		ВП	42	44
Зав. каф.		Криворучко О.В.			<i>Висновки та пропозиції</i>	Факультет інформаційних технологій, 4 курс, 7 група		
Керівник		Жирова Т.О.						
Гарант		Цензура М.О.						
Розроб		Мілевський Д.В.						

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Telegram (software) [Електронний ресурс] – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Telegram_\(software\)](https://en.wikipedia.org/wiki/Telegram_(software)).
2. Бази даних - класифікація [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Database#Classification>.
3. Бази даних - системи керування базами даних [Електронний ресурс] – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Database#Database management system](https://en.wikipedia.org/wiki/Database#Database_management_system).
4. Ієрархічна модель бази даних [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Hierarchical_database_model.
5. Об'єктна база даних [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Object_database.
6. Об'єктно-реляційна база даних [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D1%96%D0%B9%D0%BD%D0%B0%D0%B1%D0%B0%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D1%85>.
7. Реляційна модель даних [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Relational_model.
8. Python. asyncio - Asynchronous I/O [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/library/asyncio.html>.
9. Python asyncio - Coroutines and Tasks [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/library/asyncio-task.html>.

					<i>КНТЕУ 121 07-11.БР</i>			
					<i>Система адміністрування віртуальних спільнот на платформі Telegram Bot</i>	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		СД	43	44
Зав. каф.		Криворучко О.В.			<i>Список використаних джерел</i>	Факультет інформаційних технологій, 4 курс, 7 група		
Керівник		Жирова Т.О.						
Гарант		Цензура М.О.						
Розроб		Мілевський Д.В.						

10. codeguida Основы asyncio [Электронный ресурс] – Режим доступа до ресурсу: <https://codeguida.com/post/495>.
11. pgAdmin - PostgreSQL Tools [Электронный ресурс] – Режим доступа до ресурсу: <https://www.pgadmin.org/>.
12. MySQL Workbench [Электронный ресурс] – Режим доступа до ресурсу: https://en.wikipedia.org/wiki/MySQL_Workbench.

					<i>КНТЕУ 121 07-11.БР</i>	Аркуш
Изм.	Аркуш	№ докум	Подпись	Дата		44

ДОДАТКИ

Додаток А

Клас ACLMiddleware

```
class ACLMiddleware(BaseMiddleware):
    def __init__(self):
        log.info('ACLMiddleware loaded')
        super(ACLMiddleware, self).__init__()

    async def on_pre_process_message(self, message: types.Message, data: dict):
        users = []
        if message.new_chat_members:
            for u in message.new_chat_members:
                users.append(await self._process_user(u))

        users.append(await self._process_user(message.from_user))
        group = await self._process_chat(message.chat)

        for user in users:
            if not user:
                continue
            group_user = await user.user_groups.filter(group=group).first()
            if not group_user:
                await models.GroupUser.create(group=group,
                                                user=user)
            else:
                group_user.last_message_at = datetime.utcnow()
                await group_user.save(update_fields=['last_message_at'])

    async def _process_user(self, user: types.User) -> Optional[models.User]:
        if user.id == await Me.id():
            return

        entity = await models.User.get_or_none(id=user.id)
        if not entity:
            entity = await models.User.create(id=user.id,
                                             username=user.username,
                                             first_name=user.first_name,
                                             last_name=user.last_name)
```


return entity

```
async def _process_chat(self, chat: types.Chat) -> Optional[models.Group]:
```

```
    if not types.ChatType.is_group_or_super_group(chat):
```

```
        return
```

```
        group = await models.Group.get_or_none(id=chat.id)
```

```
        if not group:
```

```
            group = await models.Group.create(id=chat.id,  
                                             title=chat.title)
```

```
        return group
```

База даних додатку

```
class TimedMixin:
```

```
    created_at = fields.DateTimeField(null=True, default=datetime.utcnow)
```

```
    updated_at = fields.DateTimeField(null=True, default=datetime.utcnow)
```

```
class Group(Model, TimedMixin):
```

```
    id = fields.BigIntField(pk=True, generated=False)
```

```
    title = fields.TextField()
```

```
    welcome_message = fields.TextField(null=True)
```

```
    translation_enabled = fields.BooleanField(default=False)
```

```
    language = fields.TextField(null=True)
```

```
    idle_kick_enabled = fields.BooleanField(default=False)
```

```
    idle_delta = fields.TimeDeltaField(null=True)
```

```
    group_users: fields.ReverseRelation['GroupUser']
```

```
class User(Model, TimedMixin):
```

```
    id = fields.IntField(pk=True, generated=False)
```

```
    username = fields.TextField(null=True)
```

```
    first_name = fields.TextField()
```

```
    last_name = fields.TextField(null=True)
```

```
    issued_warns: fields.ReverseRelation['Warn']
```

```
    issued_bans: fields.ReverseRelation['Ban']
```

```
    user_groups: fields.ReverseRelation['GroupUser']
```

```
class GroupUser(Model, TimedMixin):
```

```
    group: fields.ForeignKeyRelation[Group] = fields.ForeignKeyField('models.Group',  
related_name='group_users')
```

```
    user: fields.ForeignKeyRelation[User] = fields.ForeignKeyField('models.User',  
related_name='user_groups')
```

```
    last_message_at = fields.DateTimeField(null=True, default=datetime.utcnow)
```

```
    warns: fields.ReverseRelation['Warn']
```

```
    bans: fields.ReverseRelation['Ban']
```



```
class Warn(Model, TimedMixin):
    group_user: fields.ForeignKeyRelation[GroupUser] =
fields.ForeignKeyField('models.GroupUser', related_name='warns')
    issuer: fields.ForeignKeyRelation[User] = fields.ForeignKeyField('models.User',
related_name='issued_warns')
    reason = fields.TextField(null=False)
```

```
class Ban(Model, TimedMixin):
    group_user: fields.ForeignKeyRelation[GroupUser] =
fields.ForeignKeyField('models.GroupUser', related_name='bans')
    issuer: fields.ForeignKeyRelation[User] = fields.ForeignKeyField('models.User',
related_name='issued_bans')
    reason = fields.TextField(null=False)
```

```
async def setup():
    log.info('DB engine init')
    await Tortoise.init(db_url=DB_URL,
modules={'models': ['app.models']})
    log.info('Generating schemas')
    await Tortoise.generate_schemas()
```