

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ**  
Фізико-математичний факультет  
Кафедра інформатики та прикладної математики

«Допущено до захисту»

Завідувач кафедри

\_\_\_\_\_ Моїсеєнко Н.В.

Реєстраційний № \_\_\_\_\_

«\_\_\_» \_\_\_\_\_ 2024 р.

«\_\_\_» \_\_\_\_\_ 20\_\_ р.

**РОЗРОБКА ЧАТ-БОТА «ПОМІЧНИК АБИТУРІЄНТА  
УНІВЕРСИТЕТУ»**

Кваліфікаційна робота студента

групи I-20

ступінь вищої освіти «бакалавр»

спеціальності

014 Середня освіта (Інформатика)

**Крестьянова Артема Денисовича**

Керівник: асистент кафедри

інформатики та прикладної

математики

**Закарлюка Ірина Станіславівна**

Оцінка:

Національна шкала \_\_\_\_\_

Шкала ECTS \_\_\_ Кількість балів \_\_\_\_\_

Голова ЕК \_\_\_\_\_

(підпис)

(прізвище, ініціали)

Члени ЕК \_\_\_\_\_

(підпис)

(прізвище, ініціали)

(підпис)

(прізвище, ініціали)

(підпис)

(прізвище, ініціали)

## ЗАПЕВНЕННЯ

Я, Крестьянов Артем Денисович, розумію і підтримую політику Криворізького державного педагогічного університету з академічної доброчесності. Запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають покликання на відповідне джерело. Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Криворізького державного педагогічного університету ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

## ЗМІСТ

<b>ВСТУП</b> .....	4
<b>РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ АВТОМАТИЗОВАНИХ СИСТЕМ КОМУНІКАЦІЙ</b> .....	6
1.1. Роль CRM – систем в організації освітнього процесу .....	6
1.2. Дослідження аналогів розробки .....	13
1.3. Моделювання автоматизованої системи комунікації .....	17
Висновки до розділу 1 .....	21
<b>РОЗДІЛ 2. РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ КОМУНІКАЦІЇ ІЗ ЗАСТОСУВАННЯМ TELEGRAM API</b> .....	22
2.1. Обґрунтування вибору системи керування базами даних .	22
2.2. Розробка бази даних .....	26
2.3. Вибір інструментів для розробки системи.....	28
2.4. Реалізація взаємодії абітурієнта з системою .....	32
Висновки до розділу 2 .....	43
<b>ВИСНОВКИ</b> .....	45
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	46
<b>ДОДАТКИ</b> .....	48

## ВСТУП

У сучасній системі вищої освіти комунікація між здобувачами освіти, навчальними/навчально науковими працівникам та представникам органів адміністрування освітнього процесу відіграє важливу роль у забезпеченні якості надання та здобуття освіти. Але з урахуванням сучасних тенденцій та постійних світових метаморфоз традиційні канали комунікації втрачають свою актуальність на фоні недостатньої оперативності та зручності.

За рахунок збільшення кількості інформація страждає її якість і це призводить до того що певна частина обміну нею постійно повторюється, але з різними дійовими особами. Саме тому важливо досліджувати можливості автоматизувати та оптимізувати ці процеси, задля збереження людського ресурсу.

Найкращий спосіб зробити це – забезпечити доступність актуальної інформації для здобувачів освіти в зрозумілому вигляді та в зручному місці. Дошки оголошень на стінах корпусів вже давно не є такими, у той час як простори соціальних мереж точно мають великий потенціал для цього. Оперативний доступ до потрібних даних там, де здобувач освіти проводить найбільшу кількість часу – це неабиякий крок не лише до впевненості в тому, що інформація буде отримана, але й до імплементації освітнього процесу в його зону комфорту. Завдяки цьому можливе також підвищення залученості здобувачів освіти.

При цьому важливо забезпечити цей доступ не витрачаючи забагато ресурсів. Для цього в нагоді може стати автоматизована система для комунікації із здобувачами освіти. З урахуванням потреби її розміщення у сфері інтересів користувачів, чудовим варіантом буде створити її безпосередньо в рамках застосунку, який є популярним серед визначеної аудиторії. Таким можна вважати Telegram Messenger, який в свою чергу має всі потрібні інструменти для побудови подібної системи у вигляді власного API.

Наразі, застосування соціальної мережі Telegram Messenger для автоматизації комунікації у сфері вищої освіти є порівняно новим напрямком, відповідно ступінь вивченості проблеми є доволі малою.

**Мета дослідження** полягає у дослідженні та реалізації проекту автоматизованої системи через Telegram API задля оптимізації та підвищення ефективності комунікації між закладом вищої освіти та здобувачами (абітурієнтами).

Для досягнення поставленої мети було визначено наступні **завдання**: проаналізувати процеси комунікації у закладах вищої освіти; дослідити існуючі автоматизовані системи комунікації; створити модель такої системи; обрати інструменти для її реалізації; побудувати структуру збереження даних; відповідно до створених моделей, розробити програмне забезпечення, яке задовольнятиме меті.

В процесі проведення дослідження використовувалися такі методи: аналіз наукової літератури; вивчення документації; системний аналіз; проектування програмного забезпечення; кодування; тестування.

**Предметом дослідження** є процес розробки системи автоматизованої комунікації із використанням технологій Telegram API.

**Об'єктом дослідження** виступає автоматизована система для оптимізації взаємодії та комунікації між здобувачами та закладом вищої освіти через Telegram API.

Наукова цінність дослідження полягає у розвитку теоретичної та практичної складової автоматизованих підходів до комунікації в сфері вищої освіти.

Практична значущість результатів дослідження визначається тим, що розроблена система може бути запроваджена у закладі вищої освіти задля оптимізації процесів комунікації із здобувачами освіт.

# РОЗДІЛ 1

## ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ АВТОМАТИЗОВАНИХ СИСТЕМ КОМУНІКАЦІЙ

### 1.1. Роль CRM – систем в організації освітнього процесу

Через збільшення потреби передавати та отримувати інформацію у дистанційному режимі в зв'язку із пандемією коронавірусної хвороби 2019 випадки використання цифрових інструментів, як і їх кількість, почали зростати з великою швидкістю і те, що колись було особливою ознакою для певних закладів стало потребою для кожного. Платформи та застосунки, які отримали попит розподіляються на три категорії (рис. 1.1).

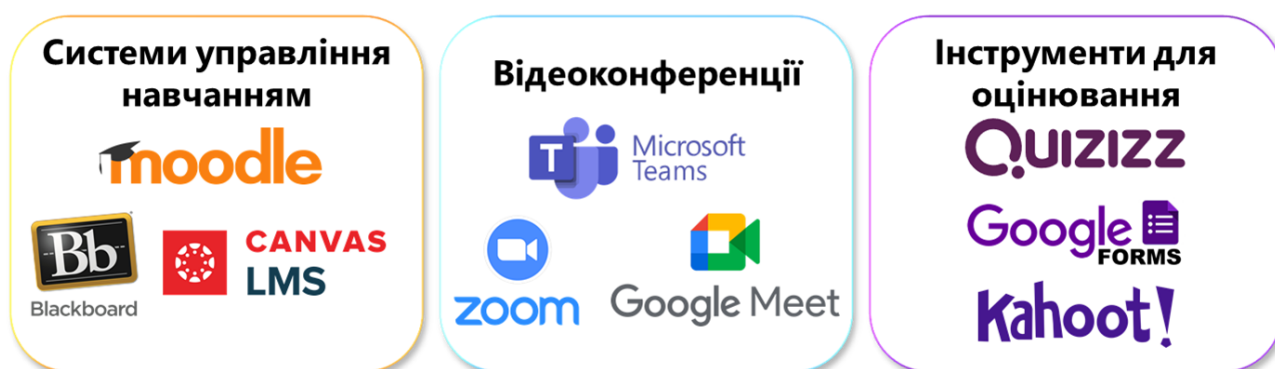


Рис. 1.1 Платформи та застосунки для організації освітнього процесу

Окремої уваги заслуговує той факт, що платформи для організації відеоконференцій, які почали використовувати для організації освітнього процесу до цього були розроблені та здебільшого застосовувалися у корпоративній сфері. Це призвело до деяких проблем, пов'язаних з їх адаптацією до потреб нового середовища. В першу чергу – це технічні складнощі. Деякі з платформ мають потребу у високій пропускну здатності інтернету, яка не завжди могла бути задоволена наявною у здобувачів освіти. Виникали питання щодо зручності та інтуїтивної зрозумілості інтерфейсу, адже цей фактор є дуже важливим у випадку, коли аудиторія не мала досвіду роботи

із онлайн-інструментами. Відсутність спеціалізованих інструментів, які б стали у нагоді для додавання інтерактивності освітньому процесу також мала місце серед складнощів, із якими користувачі стикалися на початку дистанційного навчання. Враховуючи психологічні аспекти пов'язані зі різкою зміною умов та навколишнього середовища, ця модернізація освіти стала надскладним викликом для усіх, хто був причетний до цього.

З часом ситуація стабілізувалася, технології розвинулися, були створені взагалі нові платформи, наприклад, Google Classroom, Flipgrid та Miro, і, вже сьогодні, коли більшість здобувачів вищої освіти не стикалася із університетськими аудиторіями, дистанційна форма навчання має можливість стати однією із основних. Але навіть якщо цього не станеться, сьогодні, після форсованих розробок у сфері освіти, а також підвищення цифрової обізнаності його учасників, ми маємо велику кількість можливостей для покращення процесів комунікації шляхом їх оптимізації та автоматизації, як це вже відбувається із процесом оцінювання здобувачів освіти: для проведення тестувань або контролю знань навчальними/навчально-науковими працівниками використовуються платформи, які мають можливість налаштування автоматичного оцінювання. Одним з шляхів до цього є розробка систем мета яких полягатиме у зменшенні потреби живої участі, там де це реально, задля вивільнення можливості займатися іншими задачами, які наразі не виконуються програмним способом. При цьому важливо, щоб це програмне забезпечення було реалізовано на платформах зручних для користувачів та тих, які задовольнятимуть цілі розробки [5].

Університет, публікуючи інформацію або залучаючи суспільство до співпраці, може застосовувати різні способи взаємодії. Як цільову групу можуть бути обрані як школярі, так і вчителі шкіл, батьки учнів, або організації, які співпрацюють з університетом. У рамках наступного огляду позначимо різні цільові групи як зацікавлених у інформації та співпраці осіб.

Визначимо кілька груп застосовуваних інформаційних систем.

1. Офіційні веб-сайти університетів, які містять інформацію як для студентів та співробітників, так і для всіх зацікавлених осіб.

2. Групи та профілі університетів у соціальних мережах.

3. Веб-сайти з інтерактивним або роз'яснювальним контентом для майбутніх студентів.

Як приклади можна навести веб-сайт Університету Центрального Ланкашира (The University of Central Lancashire, UCLan) [6] або веб-сайт Единбурзького Університету імені Джона Нап'є (Edinburgh Napier University) [7]. Основний контент, представлений на подібних сайтах, – інформація про університет та карта корпусу, а також корисна інформація щодо студентського життя.

4. CRM - системи, що застосовуються зараз у багатьох університетах по всьому світу. Серед розробок - "Система управління взаємовідносинами для освітньої установи" компанії "VPgroup", основним функціоналом яких є управління базою контактів відповідно до цілей і завдань ЗВО та повна інтеграція взаємовідносин, починаючи зі шкільної лави [2].

5. Веб-сайти із матеріалами лекцій, практик, можливістю навчання. Прикладами є опубліковані в мережі лекції Стендфордського університету як проект «Stanford Online» [3] та проект у сфері онлайн-освіти Coursera, що не належить, однак, до певного університету [1].

6. Веб-сайти центрів додаткової освіти, основна діяльність яких – організація та проведення різноманітних конкурсів у сфері освіти. Вони також не належать до діяльності університетів, однак можуть бути посередником між університетом та суспільством, про які згадувалося раніше.

Як приклад університету, що активно взаємодіє з суспільством і тим самим привертає увагу нових абітурієнтів, варто навести офіційний веб-сайт Кембриджського університету. Сторінки сайту присвячені окремим проектам, що реалізуються за допомогою університету, інформації для учасників та волонтерів, а також новинам та майбутнім подіям. Веб-сайт університету виконує лише інформаційні функції; можливість залишити відгук чи отримати



консультацію відсутня. Таким чином, серед цілей, наведених у першому розділі, виконується лише перший – інформування. Таку мету мають, інтерактивні веб-сайти та веб-сайти з освітнім контентом.

Інша мета – консультування – може бути досягнута за допомогою створення профілю у соціальних мережах або використання інших інструментів, як, наприклад, сервісів інтерактивних опитувань або менеджера заходів з функціями анкетування та реєстрації. Однак у такому разі відсутня можливість створення єдиної бази даних зацікавлених осіб для подальшої взаємодії. Використання різних інструментів призведе до накопичення розрізнених даних, що знижує ефективність роботи.

Для аналізу переваг та недоліків кожної з груп визначимо кілька критеріїв:

- 1) взаємодія з різними групами користувачів;
- 2) необхідні зусилля для залучення аудиторії;
- 3) наявність персональних даних про користувачів;
- 4) мотивація користувачів використовувати запропоновану систему;
- 5) можливість інтеграції з існуючими системами.

Порівняння класів інформаційних систем згідно з обраними критеріями наведено в таблиці 1.1.

При підбитті підсумків порівняння слід зазначити кілька моментів. Насамперед, це важливість тривалої взаємодії зі школярами та їх оточенням, що веде до необхідності реєстрації в ІС. Іншими словами, для ефективного функціонування ІС необхідно, щоб користувачі не просто переглядали сторінки веб-сайту, але й залишали дані для подальшого інформування та втягування в життя університету. По-друге, інформація, що містяться в цих даних, повинна чітко класифікувати тип користувача (школяр/вчитель), що важливо для подальшої взаємодії та мотивування.

Таблиця 1.1.

### Порівняння існуючих інформаційних систем

Критерій	Офіційні веб-сайти	Веб-сайти з дод. матеріалами та навчальні веб-сайти	CRM – системи
Взаємодія	Надається інформація для всіх зацікавлених контрагентів	Надана інформація залежить від спрямованості	Рішення суворо орієнтовані взаємодію з абітурієнтами
Охоплення аудиторії	Необхідно додатково залучати аудиторію	Необхідно додатково залучати аудиторію	Необхідне створення повної БД користувачів
Надання персональних даних	Не надаються; відсутня можливість подальшої взаємодії	Надаються за бажанням користувача	Надаються
Мотивація для відвідування	Пошук інформації	Пошук та використання дод. матеріалів; дод. навчання	відсутній
Можливість інтеграції	Існує	відсутній	відсутній

Кожен із класів існуючих систем має як переваги, так і недоліки. Основним недоліком офіційних веб-сайтів є неможливість встановити конкретних користувачів і продовжити взаємодію. Іншими словами, взаємодія здійснюватиметься лише за відповідної ініціативи користувача. Більш того, за допомогою цього типу ІС відсутня можливість будь-якої мотивації користувачів. Аналогічний недолік притаманний і веб-сайтам з інтерактивним контентом.

Групи в соціальних мережах, що є на даний момент одним з найбільш популярних методів взаємодії, мають безліч переваг, як, наприклад, охоплення аудиторії та можливість її сегментування. Проте, по-перше, відсутня можливість інтеграції, а по-друге, подібна взаємодія носить переважно неофіційний характер і найчастіше виступає як підтримка офіційного джерела.

Веб-сайти з освітнім контентом та веб-сайти центрів додаткової освіти є прикладами інформаційних систем з певними мотивами використання. Додавання подібних сторінок з актуальною та навчальною інформацією дозволило б додатково залучити аудиторію.

Використання CRM – систем дозволяє зберігати базу даних користувачів та сегментувати їх; має розширений функціонал проти іншими рішеннями. Але недоліками такого підходу є, по-перше, необхідність спочатку повної бази даних; по-друге, націленість існуючих рішень на взаємодію виключно з абітурієнтами. Крім того, подібні рішення мають високу вартість та спрямовані, як було зазначено вище, на комплексний супровід навчального процесу до випуску студента [8].

Тоді як, сучасні месенджери є одними з найпопулярніших засобів комунікації в світі. Вони дозволяють обмінюватися фото, текстовими повідомленнями, відео, аудіофайлами та іншими типами контенту у реальному часі. Месенджери мають ряд переваг порівняно з іншими засобами комунікації, такими як телефонні дзвінки або електронна пошта. Вони більш зручні, оскільки дозволяють користувачам спілкуватися з будь-якої точки світу, де є доступний Інтернет. Месенджери також більш економічні, оскільки вони не вимагають оплати за хвилини розмови або обсяг переданих даних. Функціональні можливості теж постійно розширюються. Крім базових функцій обміну повідомленнями, сучасні месенджери пропонують широкий спектр додаткових можливостей, таких як:

- голосові та відеодзвінки. Месенджери дозволяють користувачам спілкуватися в режимі реального часу за допомогою голосу або відео. Це дає можливість бачити та чути один одного в реальному часі, що робить спілкування більш природним та емоційним;
- групові чати. Месенджери дозволяють створювати групи для спілкування з декількома людьми одночасно. Це робить їх ідеальним інструментом для спілкування з родиною та друзями, а також для проведення зустрічей або конференцій.

Telegram Messenger – це хмарна програма обміну повідомленнями для смартфонів, портативних та стаціонарних комп’ютерних пристроїв, що може бути використане для цього. На кінець 2023-го року 76% громадян України були активними користувачами цієї платформи, 72% використовували її для отримання новин, 75% – для спілкування. Ці показники є найбільшими у порівнянні із іншими застосунками, тому для розробки автоматизованої системи комунікації із здобувачами освіти був обраний саме Telegram Messenger [9].

Основний і єдиний інструмент для реалізації подібних проектів в рамках цієї платформи – це Telegram API, інтерфейс програмування застосунків. За допомогою нього розробники отримують доступ до функцій, які використовуються для роботи з Telegram Messenger. Це відбувається за методом RESTful API, коли сервер застосунку отримує запит через HTTP, обробляє його та надсилає відповідь у форматі JSON. Завдяки ньому існує можливість створювати ботів, які виступають автоматизованими системами комунікації у поданому середовищі. Для того аби використовувати Telegram API розробник з початку повинен отримати токен API – набір символів потрібний для аутентифікації запитів. Це можна зробити з допомогою «BotFather», через який усі боти створюються (рис. 1.2). Також він використовується для проведення додаткових налаштувань: додавання та редагування піктограми, опису, вітального повідомлення, швидких команд; режим та правила використання безпосередньо в інших чатах, зокрема групових; підключення платежів; налаштування меню; передача права володіння; остаточного видалення.

Із інструментарієм який надається розробникам через Telegram API можна створити найрізноманітніші системи, наприклад, ботів помічників, для новин, розваг, спілкування, інтеграції зі сторонніми сервісами, творчості, інвестування, криптовалют, тощо. Відповідно, потреби програми, мета якої допомагати у наданні та передачі інформації від та для учасників освітнього процесу, будуть повністю задоволені функціями, які має цей інтерфейс програмування застосунків.



Рис. 1.2. Telegram-бот через який створюються усі інші боти

## 1.2. Дослідження аналогів розробки

Перш ніж зробити висновок, щодо того як повинна виглядати автоматизована система комунікації створена із застосування Telegram API, пропонується розглянути вже функціонуючі боти закладів вищої освіти. Один з таких був розроблений у Міжнародному Європейському університеті (рис. 1.3).

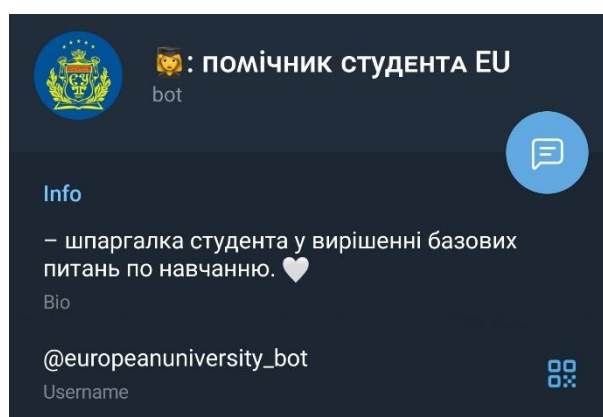


Рис. 1.3. Telegram-бот Міжнародного європейського університету

Функціонал системи дозволяє здобувачам освіти отримати відповіді на найпоширеніші питання за категоріями (рис. 1.4). Натискання кнопки «Навчальний процес» надає доступ до інших:

- «Коли розпочнеться навчання?»;
- «Де можна знайти розклад занять?»;

- «Коди доступів до навчальних ресурсів»;
- «Як і де замовити де замовити довідку про навчання?»;
- «Де знайти контакти університету? Як зв'язатись з деканатом?».

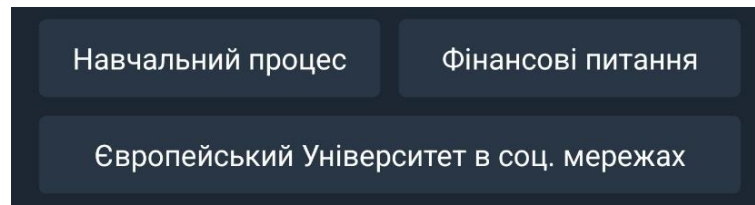


Рис. 1.4. Головне меню Telegram-бота Міжнародного європейського університету

Взаємодія з кожним елементом керування призведе до надсилання ботом повідомлення із відповіддю на певне питання. Розділ «Фінансові питання», до якого також можна потрапити натиснувши кнопку в головному меню, діє за тим же принципом, але стосується інших пунктів:

- «Як зв'язатись з касою? Які питання можливо вирішити через неї?»;
- «Місцезнаходження каси»
- «Реквізити для сплати онлайн»
- «Онлайн оплата»

У свою чергу третій елемент із підписом «Європейський Університет в соц. мережах» просто надсилає повідомлення із посиланнями, а не веде до іншого розділу. Також, варто зауважити, що кожне підменю має кнопку «Назад», яка дозволяє повернутися до головного меню.

Відповідно до інформації, яка з'являється при запуску бота командою «/start», він був створений із використанням іншого – «Manybot». Він дозволяє користувачам створювати свої власні системи без використання мов програмування.

Інша автоматизована системи комунікації, яка буде розглянута використовується в Університеті Короля Данила. Відповідно до назви: «Розклад Університету Короля Данила», – зрозуміло, що основне призначення

боту полягає у наданні розкладу. Після детального дослідження функціоналу, було виявлено, що у системі доступні інструменти як для здобувачів освіти, так і для навчальних/навчально-наукових працівників. Отримання відбувається із застосуванням фільтрів:

- за днем тижня;
- поточний тиждень;
- наступний тиждень.

Використання системи починається з обрання академічної групи.

Користувач отримує в користуванні три варіанти пошуку:

1. За шифром академічної групи з використанням умовних полів зі списками, які формуються на основі попереднього вибору: «Навчальний заклад  $\Rightarrow$  форма навчання  $\Rightarrow$  спеціальність  $\Rightarrow$  курс  $\Rightarrow$  група».
2. За шифром академічної групи шляхом його введення та надсилання. Важливо ввести усі символи в правильному порядку від початку та до кінця, реєстр при цьому різниці не має.
3. За прізвищем, ім'ям та по батькові навчального/навчально наукового працівника. Можливий пошук з використанням лише прізвища, лише ім'я та лише по батькові, а також їх довільних комбінацій.

Ботом надсилається повідомлення із розкладом на поточний день, але його можна змінити шляхом натискання на відповідні кнопки (рис. 1.5). Обраний варіант замість тексту відображатиме біле коло. Також є функція для його додавання до списку обраних, що дозволяє швидко отримувати доступ до нього з розділу «Обране» без використання пошуку. Крім того, розклад можна зробити основним для свого облікового запису, щоб мати доступ до нього одразу з головного меню через кнопку «Мій розклад» та отримувати нагадування про заняття, які повинні відбутися, якщо така функція була ввімкнена у налаштуваннях бота.

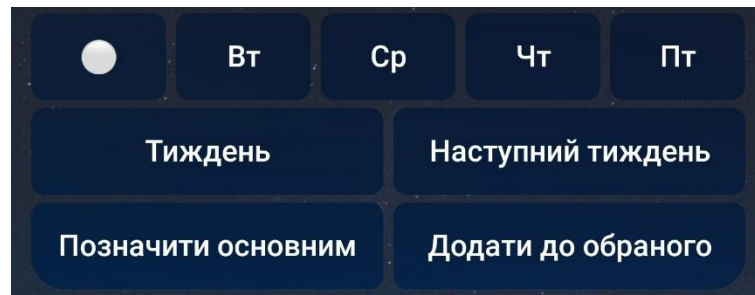


Рис. 1.5. Параметри для налаштування повідомлення з розкладом

Відповідно до наведеного інструментарію, систему розроблену в Університеті Короля Данила можна вважати гарним прикладом впровадження оптимізації та зручності для своїх здобувачів освіти та навчальних/навчально-наукових працівників. Особливо вирізняється система нагадувань, що є гарним доповненням і допомогою в питанні організації власного часу.

Останньою до розгляду пропонується автоматизована система комунікації Державного податкового університету. Відповідно до опису боту на Facebook сторінці та у Telegram каналі закладу вищої освіти, користувачам пропонується широкий спектр функцій:

- розклад навчального процесу персоналізований за академічною групою та відокремлений за часовим проміжком;
- інформацію щодо отриманої та повернутої літератури в науковій бібліотеці Університету;
- дані про фінансову заборгованість;
- список навчальних підрозділів закладу та посилання на їх сторінки;
- посилання на останні 10 записів із розділу «Новини» на сайті Університету;
- можливість залишити відгук щодо роботи боту та пропозиції щодо можливих покращень.

Із наведеного функціоналу у вільному доступі знаходяться інструменти для отримання розкладу, списку навчальних підрозділів університету та останніх новин. Усі інші потребують додаткової авторизації здобувача освіти за номером телефону. І хоча це заважає дослідженню, у той же час,



персоналізований підхід можна вважати неабиякою перевагою сервісу, адже таким чином користувач має змогу отримуватиме більшу кількість даних у яких він зацікавлений. Крім того, той час який би працівники закладу освіти використовували для проведення консультацій щодо фінансової заборгованості або ж контролю обігу літератури, можна використати на інші справи. Це є гарним кроком до оптимізації процесів та автоматизації процесу комунікації.

### **1.3 Моделювання автоматизованої системи комунікації**

Грунтуючись на результатах аналізу існуючих ботів закладів вищої освіти, пропонується створити модель її роботи. Для цього визначимо складові архітектури. Вона складатиметься з наступних компонентів:

1. Інтерфейс користувача, яким виступатиме бот у Telegram Messenger і через який відбуватиметься взаємодія із системою.
2. Серверна частина, яка виконуватиме обробку запитів, зберігання, зміну даних, взаємодіятиме з Telegram API тощо.
3. База даних у якій зберігатиметься інформація щодо користувачів та дані потрібні для роботи системи.

Можливості обраної технології дозволяють втілити в реальність велику кількість сценаріїв взаємодії, але для ефективності розробки визначимо ключові моменти та зосередимося на них. Таким, зокрема, є процес реєстрації користувачів у системі (рис. 1.6). Він є важливим, адже з нього почнеться накопичення користувацького досвіду, відповідно з'явиться перше враження від роботи з ботом.

Виконання сценарію розпочнеться після першої взаємодії із системою шляхом надсилання відповідної команди, яка є стандартною для Telegram Messenger. Відповідно до отриманого запиту сервером буде ініційовано початок реєстрації, про що буде створено запис у базі даних. Користувач отримає повідомлення про потребу надання додаткової персональної інформації.

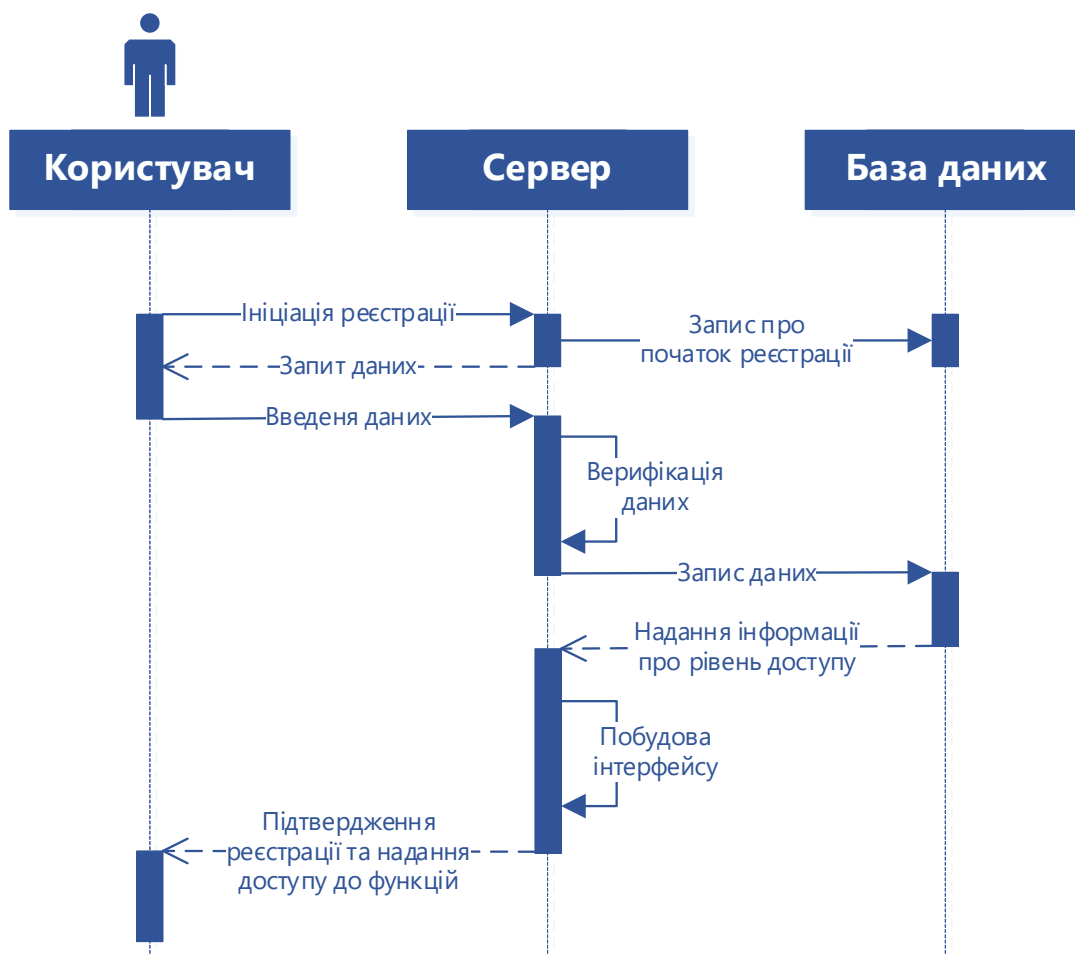


Рис. 1.6 Реєстрація користувача у системі

Щойно всі потрібні дані будуть надіслані, буде проведена верифікація даних для забезпечення відповідності формату, який використовується для їх збереження. Якщо цього не станеться – операція по запиту даних повториться. У іншому ж випадку відбудеться запис інформації до бази даних. Паралельно буде здійснено операцію по отриманню даних про рівень доступу користувача до функціоналу, на основі них визначено та створено структуру інтерфейсу. Врешті решт буде надіслано повідомлення із підтвердження про завершення реєстрації та забезпечено доступ до використання системи.

Наступний можливий сценарій – це запит даних від користувача (рис. 1.7). Концепція системи дозволяє припустити, що він буде використовуватися найчастіше, адже в своїй більшості боти закладів вищої освіти виконують функцію інформування з розповсюджених питань. Цей процес розпочинається в момент отримання запиту від користувача.

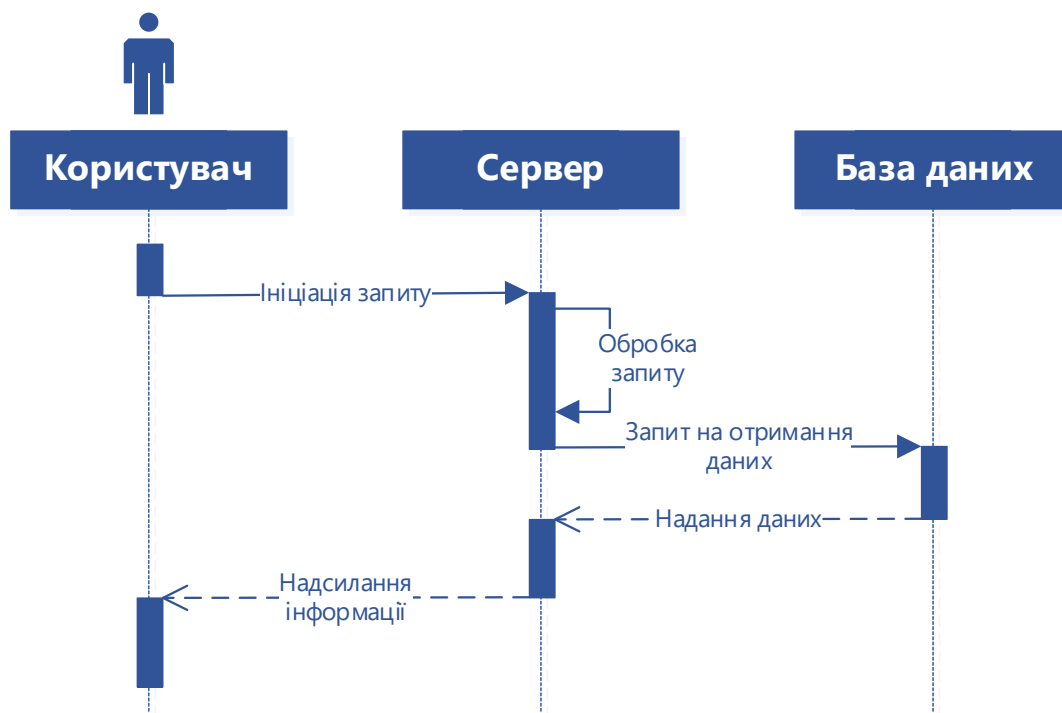


Рис. 1.7. Запит даних

Як тільки сервер його отримує, відбувається його обробка. Її мета визначити правильність запиту, та які саме дані та у якому форматі повинні бути отримані, адже, як вже було зазначено, операції за цією моделлю буде багато і важливо забезпечити коректність їх виконання. Після успішної перевірки, сервер запитує дані в базі даних та отримавши їх формує остаточне повідомлення та надсилає інформацію користувачеві із можливістю додаткових уточнень.

Важливим моментом є те, що такі запити можуть бути як поодинокі, так і складатися один з одним задля отримання більш детальних підсумкових даних. Тому у повідомленнях, які надсилатимуться системою, можлива наявність інструментів для реалізації додаткових запитів.

Так як головною метою є розробка автоматизованої системи комунікації, важливим є забезпечення можливості масової розсилки повідомлень користувачам (рис. 1.8).



Рис. 1.8. Розсилка повідомлень користувачам

У цьому випадку для ініціації сценарію потрібен адміністраторський рівень доступу. Щойно процес буде розпочато від серверу до користувача буде надісланий перелік уточнюючих запитів задля того аби обмежити, або навпаки збільшити коло здобувачів освіти, які отримають повідомлення. Коли всі дані для фільтрування будуть отримані та перевірені, відбудеться запит до бази даних для отримання ідентифікаторів користувачів, обмежених встановленими умовами. В момент, коли потрібна інформація буде отримана, вона буде підготована до використання та додатково перевірена на відповідність формату.

Наступний крок – запит на введення тексту повідомлення адміністратором. Після його отримання, сервер записує його до бази та просить

користувача підтвердити його зміст. Щойно це відбудеться, програмою буде розпочато процес розсилки та виконання дій для збереження статистики про нього, як от користувачі які отримали або не отримали повідомлення. Врешті решт усі дані про виконання операції будуть записані, а інформаційне повідомлення про результат надіслане адміністратору.

## **Висновки до розділу I**

Основне завдання роботи полягає у створенні автоматизованої системи комунікації із здобувачами освіти з використанням Telegram API. Мета розробки – оптимізація процесів у закладі вищої освіти та надання користувачам зручного доступу до інформації, при цьому використовуючи новітні засоби спілкування, які мають широкий попит. Також програма буде корисною у випадку, коли виникне потреба масового оповіщення здобувачів освіти.

Деякі заклади вищої освіти наразі вже мають подібні системи та використовують їх. Із наявного функціоналу вдалося визначити, які саме інструменти є популярними. Це дозволило розробити моделі основних сценаріїв які відбуватимуться при взаємодії користувача із ботом та забезпечити можливість їх масштабування, за рахунок чіткого визначення процесів.

## РОЗДІЛ 2

### РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ КОМУНІКАЦІЇ ІЗ ЗАСТОСУВАННЯМ TELEGRAM API

#### 2.1. Вибір системи керування базами даних

Маючи моделі сценаріїв роботи потрібної автоматизованої системи комунікації, а відповідно і розуміння того які данні можуть бути використані, можна приступити до розробки бази даних для програмного застосунку. Але навіть із наявної інформацією це є складним завданням, яке включає в себе критично важливі етапи, які можуть вплинути на роботу системи або її подальший розвиток. Тому, перш за все, є необхідність провести детальний аналіз вимог, доступних інструментів, створити проєкт структури даних та їх взаємозв'язки, важливо визначити які дані повинні бути збереженими, їх тип і вигляд.

Першочерговою задачею постає обрання СКБД, яка задовольнятиме потребам застосунку. За своїм принципом вони поділяються на:

- реляційні;
- об'єктно-орієнтовані;
- NoSQL.

RDBMS – це одні з найпоширеніших типів СКБД, побудовані на реляційній моделі даних, яка була запропонована Едгаром Коддом у 1970 році. Такі системи організовані в структуровані таблиці, кожна з яких складається з рядків, які містять окремі записи, і стовпців, у яких знаходяться значення певного типу даних.

Невід'ємною ознакою реляційних баз даних є ключі, які поділяються на два види. Перший – первинні ключі, Primary Key, виступають унікальними ідентифікаторами для кожного запису в таблиці. Таким чином забезпечується цілісність сутностей. Другий вид – зовнішні ключі, Foreign Key, поля які є посилання на первинні ключі інших таблиць. Вони виступають основою

забезпечення механізму цілісності посилань та зв'язку між різними наборами даних всередині бази. Для взаємодії із RDBMS використовується мова SQL. З її використанням відбувається визначення, управління та контроль даних. Це процеси які включають в себе створення таблиць, додавання, оновлення або видалення даних, запити на їх отримання із бази.

Реляційні СКБД мають великий коефіцієнт використання в таких галузях:

- освіта;
- охорона здоров'я;
- фінанси;
- електронна комерція тощо.

Їх перевага – це структурованість, яка забезпечує легку організацію та є зручною для запитів. Завдяки принципам їх роботи забезпечується висока цілісність даних, підтримка транзакцій, а також система контролю доступу і шифрування даних, що підвищує їх безпеку. RDBMS підтримують масштабованість, а це є запорукою для обробки великих обсягів даних і одночасного обслуговування багатьох користувачів.

В той же час, вони мають недолік у вигляді складного адміністрування та налаштування, що може стати викликом для організацій з обмеженими ресурсами. Вимоги до апаратної частини також можуть стати проблемою, якщо СКБД планується розміщувати на слабкому серверному обладнанні. Крім того, у випадках коли наявний великий обсяг неструктурованих даних і потреба роботи з ними саме в такому форматі, реляційні системи поступаються базам даних NoSQL. У цьому ж порівнянні вони явно програють в плані своєї гнучкості для певних сценаріїв.

Наступними пропонується розглянути об'єктно-орієнтовані СКБД. Такі системи надають можливість зберігання об'єктів безпосередньо у базі даних із їх властивостями та поведінкою. Це дозволяє розробникам та адміністраторам працювати із даними у більш природний спосіб та сприяє більшій гнучкості моделі, адже є можливість використовувати складні структури, такі як вкладені об'єкти, колекції, спадкування та інкапсуляцію. На додачу відпадає потреба у

перетворені інформації в табличний вигляд, як це потрібно для реляційних систем, що спрощує розробку та обслуговування програмного забезпечення.

Задля зменшення розриву між програмною логікою та даними OODBMS інтегруються з мовами програмування, такими як Java, C++ або C#. Це є перевагою для деяких, в більшості складних, розробок. Також вони підтримують транзакції, чим забезпечують високу цілісність даних та надійність системи.

При цьому вони мають недолік в обмеженості стандартів. Для роботи з ними потрібні розробники із спеціалізованими знаннями і це є складністю через недостатньо велику їх спільноту. Об'єктно-орієнтовані СКБД можуть потребувати більший об'єм дискового простору для зберігання даних у порівнянні з реляційними та демонструвати нижчу продуктивність запитів, особливо простих.

Вже згадані вище NoSQL системи відрізняються можливістю збереження та обробки великих обсягів неструктурованої або ж напівструктурованих даних. Вони створені для подолання обмежень RDBMS, пропонують високий рівень масштабованості та гнучкості. Для реалізації цієї переваги у NoSQL СКБД відсутня схема, завдяки чому дозволено збереження даних у вільному форматі. Це означає, що з часом дані можуть змінювати структуру, не вимагаючи при цьому внесення складних змін до схеми бази даних. Завдяки цьому вони виступають особливо корисними для застосунків у яких структура даних не визначена наперед або здебільшого нестабільна.

Системи NoSQL класифікуються як:

- документо-орієнтовані – зберігають інформацію, відповідно, у документах, які в свою чергу можуть містити у собі вкладені структури і масиви;
- графові – користуються графам для організації даних, що є запорукою ефективного моделювання зв'язків та створення запитів;



- колонкові – організація матеріалу відбувається у стовпцях, це забезпечує високу пропускну здатність під час роботи великими обсягами інформації та аналітичними запитами;
- бази даних типу «ключ-значення» – збереження даних у вигляді пари «ключ-значення» робить такі системи корисними для випадків, коли потрібно швидко проводити прості операції записи та зчитування.

Основна перевага всіх NoSQL СКБД – здатність до горизонтального масштабування, яка дозволяє додавати нові сервери, щоби розподілити навантаження та збільшити обсяги для зберігання даних без втрати продуктивності. Додатковим плюсом стане їх велика доступність і стійкаість до відмов, яка забезпечується використанням розподілених архітектур. Подібні властивості особливо корисні для веб-додатків, які обробляють великий обсяг інформації в реальному часі.

У свою ж чергу недоліки проявляються у відсутності стандартів, інтерфейсі, що ускладнює потенційну інтеграцію з іншими системами. Деякі СКДБ компенсують свою високу продуктивність та масштабованість шляхом відмови від підтримки транзакцій та гарантій цілісності, що є критичним для більшості застосунків.

Характер майбутньою автоматизованої системи комунікації, а також результат аналізу інших подібних систем наводить на те, що цей випадок потребує більш класичного та традиційного класу – реляційних СКБД. Навіть враховуючи те, що сама по собі ідея Telegram боту скоріше схиляє до використання NoSQL, зокрема заради високої швидкості запису, важливо враховувати потребу у високій надійності зберігання даних, їх логічної розташованості, а також той факт, що RDBMS більш розповсюджені, їх користувачів більше, а отже і вища ймовірність отримати допомогу від дотичного колу, та врешті решт вони простіші для вивчення та розуміння, відповідно і роботи з ними.

Пропонується розглянути реляційні СКБД, які могли би бути використані для створення системи за темою роботи. Однією з них є MySQL. Вона має

відкритий програмний код, відома своєю високою швидкістю обробки запитів і великою спільнотою розробників. Іншим прикладом подібної СКБД виступає PostgreSQL, яка також має відкритий код. У свою чергу ця система вирізняється багатим набором інструментів та підтримкою стандартів SQL. Ці та інші ознаки популяризують її як таку, що гарно підходить для виконання складних запитів та високої надійності.

Інший прошарок СКБД, який варто розглянути – комерційні. Наприклад, Microsoft SQL Server, перевага якої полягає в тісній інтеграції з іншим програмним забезпеченням Microsoft Corporation. Для корпорацій вона також корисна завдяки широкому функціоналу для адміністрування та аналізу. Ще один варіант – це Oracle Database, відома своєю продуктивністю та надійністю, підтримкою обробки великого обсягу даних і складних транзакцій. Все це причина її авторитету серед крупніших корпоративних клієнтів.

Є також бази даних які відрізняються від названих раніше. Однією з них є SQLite, приклад реляційної системи, яка відрізняється тим, що працює вбудовано. Це означає, що для виконання дій, вона не потребує окремого серверного процесу. При цьому SQLite легковагова та ідеальна для невеликих та середніх проєктів, що ставить її на перше місце для застосування у мобільних додатках або вбудованих системах. В доповнення до цих переваг, вона також має і інші, які відносять до всіх реляційних баз даних: продуктивність, підтримку транзакцій, цілісність та надійність даних.

Врешті решт, база даних SQLite проста у розгортанні та розробці, а тому є гарним варіантом для розробки автоматизованої системи для комунікації із здобувачами освіти через Telegram API, адже не потребуватиме високого рівня професіоналізму для своєї підтримки, розвитку та модернізації.

## **2.2. Розробка бази даних**

На початку розділу було визначено, що розробка бази даних потребує ретельної підготовки. Тепер, коли був виконаний перший етап – обрання СКБД,

можна приступити до наступного – розпочати створення проєкту структури даних.

Кожна із наведених нижче таблиць має власні стовпці із визначеними типами даних, яких у SQLite є чотири:

- INTEGER, цілі числа: ідентифікатор, кількість, рейтинг, тощо;
- REAL, числа з плаваючою комою: вартість, рейтинговий бал, тощо;
- TEXT, текстові дані: ім'я, опис, адреса, повідомлення, тощо;
- BLOB, двійкові дані: зображення, аудіо, відео, файли тощо.

Також SQLite підтримує окремий тип даних «Null», який використовується для позначення значення, яке відсутнє.

Таблиця, яка є основною для всієї системи – USERS (таблиця 2.1). Її призначення полягає у збереженні даних облікового запису Telegram Messenger користувачів, а також їх контактних даних.

Таблиця 2.1

### Опис сутності «USERS»

Назва	Тип даних	Обов'язковість	Короткий опис
id	INTEGER	PRIMARY KEY; NOT NULL	Унікальний ідентифікатор користувача
chat_id	INTEGER	NOT NULL	ID чату користувача в Telegram Messenger
nickname	TEXT		Псевдонім користувача
first_name	TEXT	NOT NULL	Ім'я користувача
second_name	TEXT		По батькові користувача
last_name	TEXT	NOT NULL	Прізвище користувача
phone_number	INTEGER		Номер телефону користувача
personal_email	TEXT		Особиста електронна адреса користувача
note	TEXT		Додаткова текстова інформація про користувача
is_active	INTEGER	NOT NULL; CHECK(IN(0,1))	Прапор активності користувача

У стовпці «Обов'язковість» значення «NOT NULL» вказує на те, що для цього поля значення не можуть бути «Null», тобто обов'язково повинні бути

присутніми. Також, для поля «is\_active» встановлено додаткову перевірку «CHECK(IN(0,1))» мета якої обмежити дані, які можна ввести до умовного логічного типу.

### 2.3 Вибір інструментів для розробки системи

Початок повноцінної реалізації автоматизованої системи для комунікації із застосуванням Telegram API потребує декілька додаткових ретельних кроків які полягають в виборі інструментів розробки. Він ґрунтується декількох ключових факторах:

- визначення мови програмування серед широкого спектру тих що підтримуються;
- обрання фреймворків та/або бібліотек, які полегшують розробку ботів;
- інструменти повинні бути підібрані з метою максимально реалізувати функціональні можливості системи.

Пропонується розпочати цей етап роботи із вибору мови програмування, яка буде використана для розробки серверної частини Telegram боту. Однією з них є JavaScript під виглядом платформи Node.js, яка має широке використання у сфері створення web-застосунків. Вона відома своєю високою продуктивністю та здібністю керувати одночасними запитами, що досягається завдяки асинхронній обробці подій. В той же час, коли мова йде про ресурсоємний великий масив вхідних даних ця платформа програє іншим мовам програмування. Також, бібліотеки Telegram API доступні для Node.js є незрілими, відповідно підходять лише для розробки дуже простих систем.

Інший кандидат – це C#, об'єктно-орієнтована мова програмування яка використовується для розробки складних та масштабованих систем, зазвичай, корпоративних застосунків та ігор на платформі .NET. Ця мова забезпечує високу продуктивність та захищеність програмного коду, використовуючи статичну типізацію. Також має високу ефективність використання пам'яті, що є важливим для виконання умови забезпечення можливості обробляти великі

масиви даних. С# використовується сьогодні для виконання найрізноманітніших завдань та має бібліотеки, які спрямовані на полегшення розробки застосунків із використанням Telegram API. Та все ж таки, в своїй більшості, системи створені цією мовою орієнтовані на корпоративне середовище. Разом із великою кривою навчання для початківців це стає проблемою для нашого проєкту, підтримку якого має бути максимально легкою через, в більшості випадків, відсутність в закладах вищої освіти достатньо кваліфікованих працівників для такої задачі.

Виходом із цієї ситуації може стати мова програмування Python. Він є найпопулярнішим серед тих, які підтримують Telegram API, що засвідчується великою кількістю бібліотек для роботи з ним. Python вирізняється лаконічним та читабельним синтаксисом, через що є дуже зручним для початківців. Його гнучкість дозволяє створювати прості та складні системи, через що він підходить для різноманітних завдань, як от обробка текстових команд, робота з базами даних та машинне навчання. Маючи велику спільноту, він може похизуватися неабиякою кількістю навчальних матеріалів, документації та вирішених проблемних кейсів. Python підтримує швидке прототипування, що дозволяє розробникам створювати та тестувати нові функціональні можливості в малих проміжках часу, а також забезпечує високу продуктивність для середньозавантажених та малозавантажених додатків, що робить його ідеальним вибором для розробки Telegram ботів. Відповідно, для реалізації автоматизованої системи комунікації пропонується обрати саме цю мову програмування.

Python має велику кількість різноманітних бібліотек для інтеграції з Telegram API. Вибір правильної бібліотеки – є критично важливим для успішного впровадження проєкту. Кожна з них має свої переваги, але далеко не всі вони є універсальними і, в деяких випадках, могли бути створені задля вирішення певних дуже спеціалізованих завдань. Тому пропонується розглянути декілька найпопулярніших, які допоможуть зосередитись на розробці логіки системи, а не на технічних деталях взаємодії з API.

Telethon – це одна з таких популярних бібліотек. Вона підтримує всі функції які наразі існують в ботів, що є запорукою якісної розробки, адже проєкт не обмежується застарілістю технології. Також, цю бібліотеку можна вважати зрілою, а це означає що вона має достатню кількість документації та приклади коду. Основною особливістю Telethon виступає низькорівневий доступ до Telegram API, що у нашому випадку скоріше є недоліком, адже розрахунок йде на недосвідченого адміністратора. Це ж означає високу складність використання, що робить його не зручним для початківців. Крім того, у цій бібліотеці відсутня асинхронність за замовчуванням, що при обробці великої кількості запитів може призвести до проблем з продуктивністю[10].

Інший приклад це AIOGram. Він навпаки має простий та інтуїтивно зрозумілий API, що значно зменшує криву навчальності. Також є підтримка асинхронності та велика активна спільнота, завдяки якій можна буде швидко знайти рішення проблем, що виникатимуть. Гарним бонусом стане вбудований каталог ботів, де можна публікувати та знаходити системи які могли б стати в нагоді. Але при цьому ситуація з недоліками паралельна до Telethon: AIOGram не підтримує всі функції Telegram API, а також є молодого бібліотекою, що може означати меншу кількість наявної документації [11].

Існує компромісне рішення – бібліотека python-telegram-bot. Вона виступає зіркою першої величі серед всіх інших завдяки простоті використання, зумовленої інтуїтивним інтерфейсом, чіткій документації з великою кількістю прикладів, великою спільнотою, а завдяки регулярним оновленням підтримується актуальність функцій та зв'язок із Telegram API. При цьому за замовчуванням підтримується асинхронність, на відміну від Telethon, але вона не так гнучка як у випадку з AIOGram. Саме тому ця бібліотека може вважатися середнім значенням серед усіх інших та бути гарним варіантом для реалізації автоматизованої системи комунікації із здобувачами вищої освіти [12].

Окрім python-telegram-bot у розробці будуть використані інші бібліотеки, потрібні для реалізації тих чи інших завдань (таблиця 2.2)

Таблиця 2.2

## Перелік додаткових використаних бібліотек Python

Назва	Опис
Beautiful Soup	бібліотека Python для извлечения данных из файлов HTML и XML.
json	Бібліотека для роботи із даними у форматі JSON
sqlite3	Бібліотека, яка використовується для роботи з базами даних SQLite

Ще одним інструментом потрібним для подальшої роботи є токен Telegram боту. В разі його відсутності не вдасться отримати доступ до API. У першому розділі вже було сказано про те, що для його отримання потрібно використати інший Telegram бот, від розробників платформи, а саме «BotFather». Для створення облікового запису свого застосунку потрібно перейти до чату з цим ботом та ввести команду «/newbot». Буде надіслано повідомлення з проханням надіслати ім'я нового боту. Після його отримання, «BotFather» зробить запит на ім'я користувача. Воно повинно бути написано англійською та в подальшому стане посиланням на застосунок в межах Telegram Messenger і залишатиметься незмінним. Важливою деталлю є те, що всі назви ботів користувачів повинні закінчуватися на «bot» (рис. 2.1).

Щойно буде надіслане ім'я користувача, якого ще не існує серед Telegram ботів та підтверджено його формат користувач отримає повідомлення із токеном для доступу до Telegram API через HTTP. Розробники також попереджають, про те що він повинен бути зберігатися у безпеці, адже кожен хто заволодіє ним зможе повністю керувати ботом. Але на цей випадок передбачена можливість заміни токenu

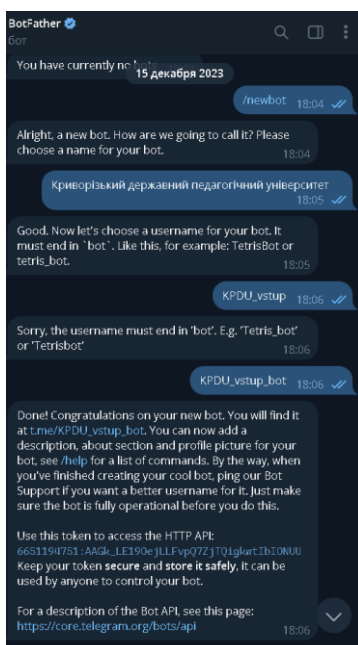


Рис. 2.1. Створення облікового запису застосунку

## 2.4. Реалізація взаємодії абітурієнта з системою

Початок роботи користувача Telegram Messenger із будь-яким ботом відбувається після надсилання стандартної команди «/start». Telegram API отримує це повідомлення та надсилає на запуснений сервер бота. Подальші операції відбуваються за участі обробників (handlers), спеціальних об'єктів, які приймають певний тип даних із урахуванням встановленого фільтру. Якщо хоч один з них зміг отримати повідомлення яке задовольняє умовам, відбувається ініціалізація прикріпленою до обробника функції.

Отже, припустимо, що маємо користувача, який вже розпочав роботу із системою. Для того аби отримати його повідомлення потрібно створити умовний сервер, додати обробник із створеною для нього функцією та запустити програму (рис. 2.2).

В результаті ми отримуємо функцію «main», у якій знаходитиметься весь програмний код пов'язаний із роботою сервера та прийомом даних від Telegram API. Із розширенням функціоналу буде збільшуватися і кількість методів у ній. Ще одна нова функція «start» характеризує те, як бот поводитиме себе отримавши команду «/start».



Після отримання команди /start, бот надсилає привітальне повідомлення (рис.2.3). В цьому повідомленні користувачеві пропонується обрати одну з категорій для отримання додаткової інформації

```
import logging
from telegram import Update, ReplyKeyboardMarkup, KeyboardButton, ParseMode
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters, CallbackContext
from my_parser import get_news
from my_sql import init_db
import sqlite3

logging.basicConfig(
    format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
    level=logging.INFO
)

logger = logging.getLogger(__name__)
ADMIN_CHAT_ID = 1139704473

# створюємо клавіатуру для головного меню
2 usages
def main_menu_keyboard() -> ReplyKeyboardMarkup:
    buttons = [
        [KeyboardButton("Новини та оголошення")],
        [KeyboardButton("Інформація про вступ")],
        [KeyboardButton("Спеціальності університету")],
        [KeyboardButton("Друга вища освіта")],
        [KeyboardButton("Поновлення та переведення")],
        [KeyboardButton("Підготовчі курси")],
        [KeyboardButton("Зворотній зв'язок")],
        [KeyboardButton("Контакти")]
    ]

1 usage
def start(update: Update, context: CallbackContext) -> None:
    update.message.reply_text(
        text="Вітаю! Я бот Університету. Оберіть один із пунктів меню:",
        reply_markup=main_menu_keyboard()
    )
```

Рис. 2.2. Ініціалізація роботи програми

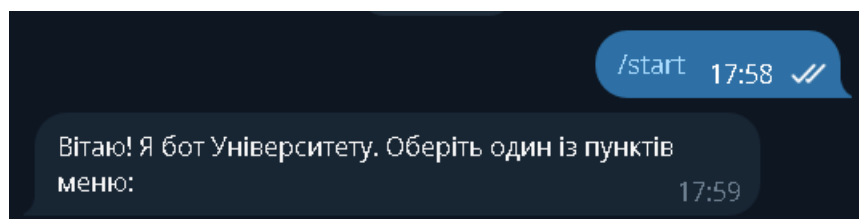


Рис.2.3. надсилає привітальне повідомлення

Бот, створений для зручної взаємодії користувачів з університетом, надає можливість швидко та легко отримати доступ до необхідної інформації. Для

цього головне меню містить основні категорії, які охоплюють всі важливі аспекти життя університету та процесу навчання.

Головне меню складається з наступних кнопок (рис. 2.4) :

- Новини та оголошення
- Інформація про вступ
- Спеціальності університету
- Друга вища освіта
- Поновлення та переведення
- Підготовчі курси
- Зворотній зв'язок
- Контакти

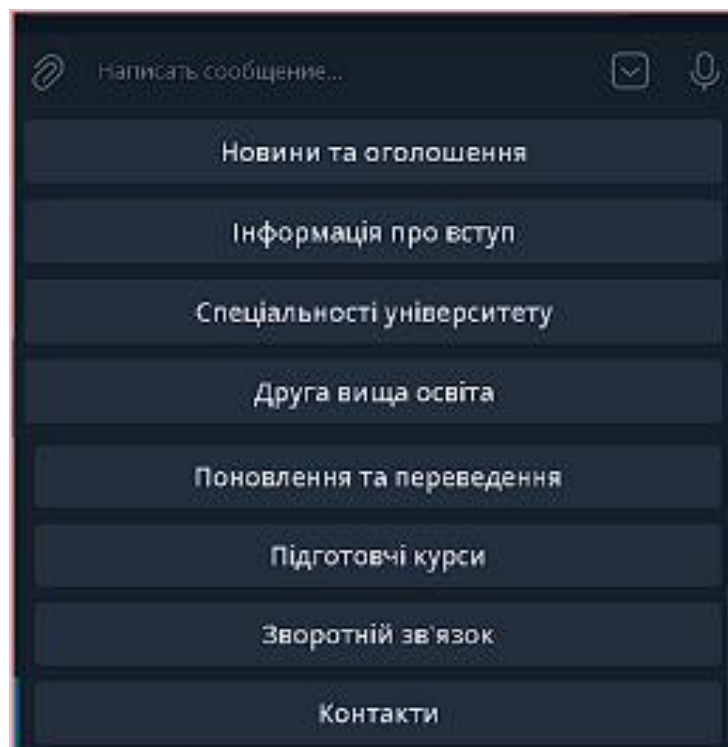


Рис. 2.4. Головне меню

Перша категорія, "Новини та оголошення", є важливою складовою роботи бота, оскільки вибору бот забезпечує користувачів актуальною інформацією про університетське життя. Ця секція створена з метою

інформування студентів, абітурієнтів та інших зацікавлених осіб про останні події, важливі повідомлення, заплановані заходи та інші значущі оновлення.

Система роботи цієї категорії побудована на основі парсера (рис 2.5), який автоматично збирає новини та оголошення з офіційного сайту університету. Парсер – це програма, яка сканує веб-сторінки, знаходить та витягує необхідні дані, в даному випадку – новини та оголошення. Використання парсера дозволяє оперативно отримувати та оновлювати інформацію, забезпечуючи її актуальність для користувачів бота [18].

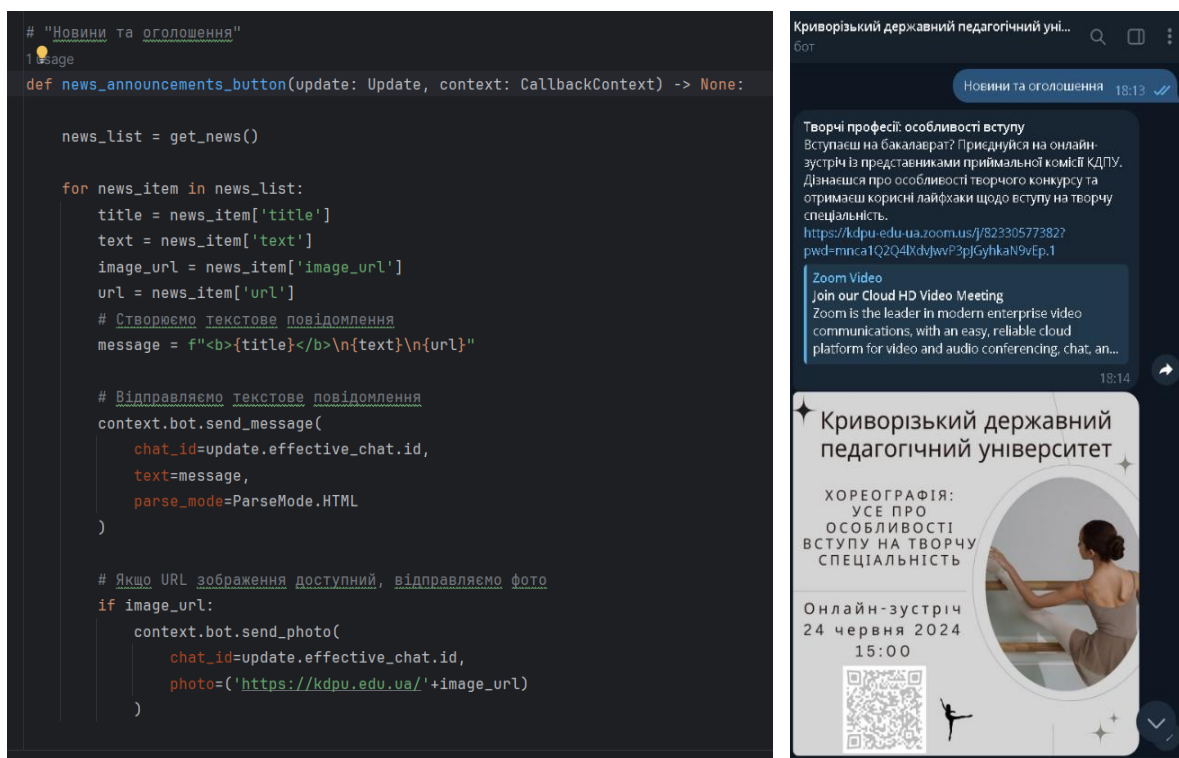


Рис 2.5. Новини та оголошення

Процес роботи парсера можна розділити на декілька ключових етапів:

1. Налаштування парсера: На першому етапі здійснюється налаштування парсера для роботи з конкретним сайтом. Це включає визначення URL-адрес сторінок, з яких будуть збиратися дані, налаштування алгоритмів пошуку та витягування необхідної інформації.

2. Сканування веб-сторінок: Парсер регулярно сканує визначені веб-сторінки університету, перевіряючи наявність нових або оновлених новин та оголошень. Це може відбуватися кілька разів на день або за іншим графіком, визначеним адміністраторами системи.
3. Збір та обробка даних: Знайдені новини та оголошення витягуються зі сторінок, після чого відбувається їх обробка. Це включає виділення заголовків, тексту новини, дати публікації, а також можливих зображень чи інших медіа-файлів.
4. Інтеграція з Telegram-ботом: Після збору та обробки, новини та оголошення інтегруються з Telegram-ботом. Користувачі можуть отримувати нову інформацію автоматично або за запитом, використовуючи відповідну команду в боті.

Далі йде розділ "Інформація про вступ"(рис. 2.6), який надає детальну інформацію про умови вступу, необхідні документи, терміни подачі заявок та інші важливі аспекти для абітурієнтів.

```
# "Інформація про вступ"
1 usage
def admission_info_keyboard() -> ReplyKeyboardMarkup:
    buttons = [
        [KeyboardButton("Майбутньому вступнику")],
        [KeyboardButton("НМТ-2024")],
        [KeyboardButton("Сторінка вступника")],
        [KeyboardButton("Для вступників з окупованих територій")],
        [KeyboardButton("Вартість навчання")],
        [KeyboardButton("Документи приймальної комісії")],
        [KeyboardButton("Розклад вступних випробувань")],
        [KeyboardButton("Вступ до магістратури")],
        [KeyboardButton("Назад до головного меню")]
    ]
    return ReplyKeyboardMarkup(buttons, resize_keyboard=True)
```

Рис. 2.6. Створення підменю

Цей розділ допоможе майбутнім студентам зрозуміти процес вступу та підготувати всі необхідні документи. Код для створення клавіатури виглядає наступним чином (рис. 2.7).

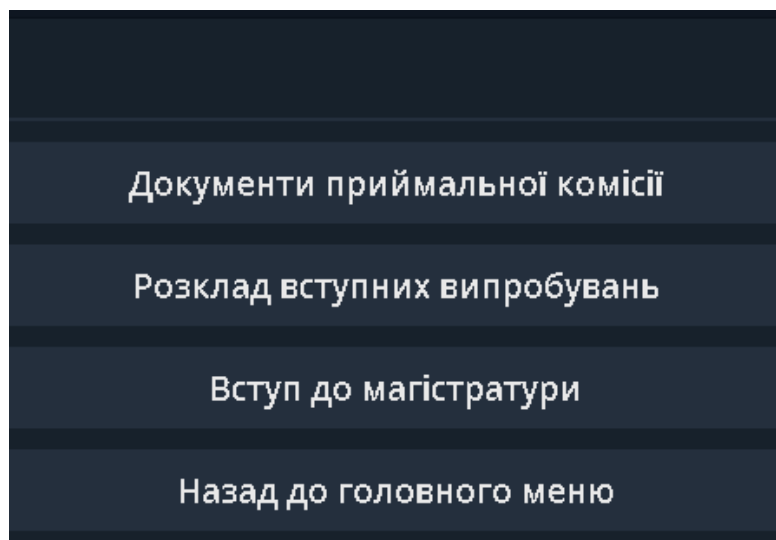


Рис. 2.7. Підменю

Цей розділ корисний для тих, хто планує вступати до нашого університету. У підменю "Майбутньому вступнику" зібрана вся базова інформація про те, як стати студентом: які потрібні документи, коли їх подавати, які є вимоги до абітурієнтів. Це своєрідна дорожня карта для тих, хто вперше стикається з процесом вступу.

Кнопка "НМТ-2024" призначена для тих, хто складає Національний мультипредметний тест у 2024 році. Тут можна знайти всю необхідну інформацію про цей іспит: дати проведення, реєстрацію, підготовчі матеріали і так далі.

"Сторінка вступника" – це особистий кабінет, де абітурієнт може слідкувати за статусом своєї заявки, перевіряти результати вступних випробувань і отримувати повідомлення від приймальної комісії. Це дуже зручно, бо все зібрано в одному місці.

Для тих, хто з окупованих територій, є окрема кнопка "Для вступників з окупованих територій". Тут розміщена інформація про особливі умови вступу, які документи потрібні і які є пільги.

Кнопка "Вартість навчання" містить інформацію про вартість навчання на різних спеціальностях. Це важливо, адже фінансовий аспект теж треба враховувати при виборі майбутньої професії.

"Документи приймальної комісії" – тут можна знайти всі необхідні форми та заяви, які треба заповнити і подати до приймальної комісії. Це полегшує процес підготовки документів, адже все, що потрібно, можна знайти в одному місці.

"Розклад вступних випробувань" містить інформацію про дати і час проведення всіх іспитів, тестів і співбесід. Це допомагає спланувати свій час і бути в курсі всіх важливих подій.

Для тих, хто планує продовжити навчання після бакалаврату, є кнопка "Вступ до магістратури". Тут розміщена вся інформація про умови вступу на магістратуру, необхідні документи і терміни подачі заявок.

І, звичайно, кнопка "Назад до головного меню" дозволяє швидко повернутися до основного меню бота, якщо потрібно знайти іншу інформацію.

Наступна категорія "Спеціальності університету" є дуже важливою для абітурієнтів, які планують обирати свою майбутню професійну кар'єру. Для реалізації цієї функції в боті створюється клавіатура з трьома основними кнопками: "Презентації факультетів", "Перелік спеціальностей" та "Назад до головного меню". Давайте розберемося, що саме ми можемо знайти у цій категорії.

"Презентації факультетів" - це, звичайно ж, перша річ, яка привертає увагу. Тут можна знайти детальну інформацію про кожен факультет університету: його спеціалізацію, перелік предметів НМТ для зарахування та відео презентація факультету . Реалізовано за допомоги функції `get_facultets(url)` (рис. 2.8), яка буде відповідальною за отримання інформації про факультети з вказаної URL-адреси. Вона буде виконувати завдання

парсингу веб-сторінки для отримання інформації про факультети. Нижче подано загальний опис функції.

Завдяки цьому абітурієнти можуть краще зрозуміти, чи відповідає факультет їхнім інтересам і цілям.

```
def get_facultets(url,):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, features='html.parser')

    title = soup.find('h2').text.strip()
    # Опис
    description_div = soup.find(name='div', style='text-align: justify;')
    description = description_div.text.strip() if description_div else ""

    specialties_header = soup.find_all('h5')[0].text.strip()
    image = soup.find('img', )
    image_url = image['src'] if image else ""
    # Промо-відео факультету
    promo_video_header = soup.find_all('h5')[1].text.strip()
    promo_video_iframe = soup.find(name='iframe', id='youtube_iijrzhUDTE')
    promo_video_url = promo_video_iframe['src'] if promo_video_iframe else ""
    # Промо-відео від студентів факультету
    student_promo_header = soup.find_all('h4')[0].text.strip()
    student_promo_iframe = soup.find(name='iframe', id='youtube_bWzhBDwz9P0')
    student_promo_url = student_promo_iframe['src'] if student_promo_iframe else ""

    faculty_data = {
        'title': title,
        'description': description,
        'specialties_header': specialties_header,
        'image_url': image_url,
        'promo_video_header': promo_video_header,
        'promo_video_url': promo_video_url,
        'student_promo_header': student_promo_header,
        'student_promo_url': student_promo_url
    }
```



Рис. 2.8. Презентації факультетів

"Перелік спеціальностей" – це той розділ, де абітурієнти можуть дізнатися про всі доступні спеціальності університету (рис. 2.9).

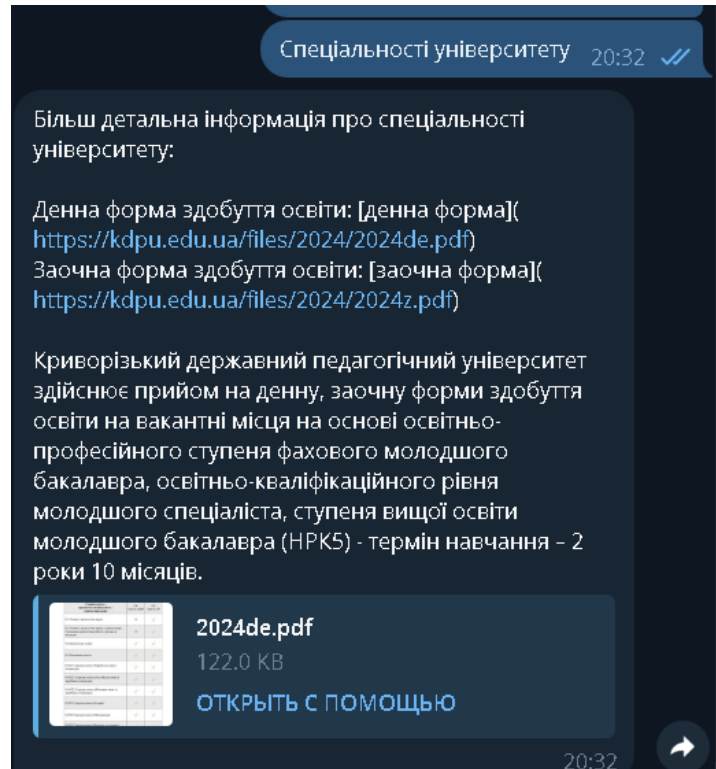


Рис. 2.9. Перелік спеціальностей

Виконано за допомоги функцій `def university_specialties_button()` та `get_links()` (рис 2. 10). Функція `get_links()` призначена для отримання посилань на файли з детальною інформацією про спеціальності університету з веб-сторінки. Ця функція допомагає абітурієнтам отримати доступ до документів, які містять важливі відомості про університетські спеціальності для подальшого вивчення.

```

def university_specialties_button(update: Update, context: CallbackContext) -> None:
    # Отримати посилання
    links = get_links()

    for links_item in links:
        day_link = links_item['a_link_one']
        correspondence_link = links_item['a_link_two']
        text_l = (
            f"Більш детальна інформація про спеціальності університету:\n\n"
            f"Денна форма здобуття освіти: [денна форма]({day_link})\n"
            f"Заочна форма здобуття освіти: [заочна форма]({correspondence_link})\n\n"
            f"Криворізький державний педагогічний університет здійснює прийом на денну"
        )
    # Відправляємо текстове повідомлення
    update.message.reply_text(text_l, parse_mode=ParseMode.HTML)

```

Рис 2. 10. функції `def university_specialties_button()` та `get_links()`



Функція починається з визначення URL-адреси, за якою знаходяться необхідні дані про спеціальності університету. Після відправлення GET-запиту та отримання відповіді, вона використовує бібліотеку BeautifulSoup для парсингу HTML-коду сторінки [18].

Далі функція знаходить блоки, які містять посилання на файли з інформацією про спеціальності університету. Для кожного такого блоку вона перевіряє, чи містить він потрібний заголовок, що вказує на наявність посилань. Якщо такий заголовок знайдено, функція шукає теги <a> в межах блоку та отримує посилання на файли.

Отримані посилання додаються до списку із словниками, де кожен словник містить два посилання: одне для денної форми навчання та одне для заочної форми навчання.

Розділ "Друга вища освіта" призначений для тих, хто вже має вищу освіту, але бажає розширити свої знання або змінити професійну сферу. Тут надається інформація про можливості та умови отримання другої вищої освіти, включаючи доступні спеціальності та вимоги до абітурієнтів. Функціонал цього розділу також може бути реалізований за допомогою парсера, який допомагає отримати та представити користувачу актуальну інформацію про навчальні програми та умови вступу.

Категорія "Поновлення та переведення" надає інформацію для студентів, які мають намір поновити навчання після перерви або перевестися з іншого навчального закладу. У цьому розділі користувач може знайти вимоги та процедури, необхідні для успішного поновлення або переведення. Аналогічно, інформацію про цей розділ можна збирати та представляти за допомогою парсера, який отримує дані з веб-сторінок університету про вимоги та процедури поновлення та переведення студентів.

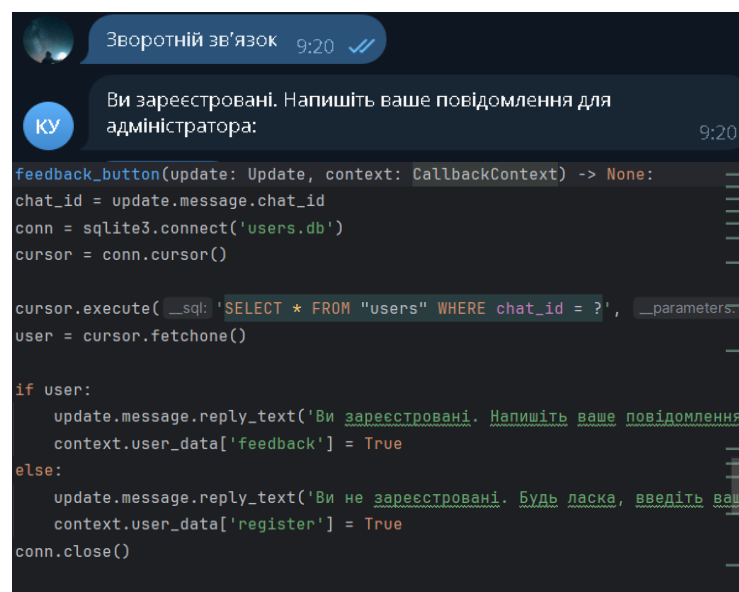
"Підготовчі курси" – це розділ для тих, хто бажає підготуватися до вступу в університет або покращити свої знання з певних предметів. Тут користувач може отримати інформацію про доступні курси, розклади занять, викладачів та

умови запису. Реалізація цих розділів також використовував парсер для отримання актуальних даних про відповідні данні.

Розділ "Зворотній зв'язок" забезпечує можливість користувачам залишати свої відгуки, пропозиції або задавати питання стосовно роботи університету чи бота.

Одним з способів реалізації цього розділу, є реєстрація користувачів. При першому контакті користувача з розділом "Зворотній зв'язок", йому запропоновано зареєструватися, вказавши свої особисті дані, такі як ім'я, прізвище, контактна інформація тощо. Після реєстрації кожен користувач буде ідентифікований у системі, що дозволить збирати та асоціювати з ним його зворотний зв'язок.

Наприклад, у функції `feedback_button` (рис. 2.11) вбудована перевірка наявності користувача у базі даних. Якщо користувач є зареєстрованим, він матиме можливість залишати відгуки та питання. Якщо користувача немає у базі даних, йому буде запропоновано зареєструватися, вписуючи необхідні дані.



The screenshot shows a Telegram chat window with a dark theme. At the top, a system message reads "Зворотній зв'язок 9:20" with a checkmark. Below it, a message from a user with initials "КУ" says "Ви зареєстровані. Напишіть ваше повідомлення для адміністратора:" followed by the time "9:20". The bottom half of the image shows a code editor with the following Python code:

```
feedback_button(update: Update, context: CallbackContext) -> None:
    chat_id = update.message.chat_id
    conn = sqlite3.connect('users.db')
    cursor = conn.cursor()

    cursor.execute(_sql: 'SELECT * FROM "users" WHERE chat_id = ?', _parameters:
    user = cursor.fetchone()

    if user:
        update.message.reply_text('Ви зареєстровані. Напишіть ваше повідомлення
        context.user_data['feedback'] = True
    else:
        update.message.reply_text('Ви не зареєстровані. Будь ласка, введіть ваш
        context.user_data['register'] = True
    conn.close()
```

Рис. 2.11. `feedback_button`

Остання категорія, "Контакти", містить всю необхідну контактну інформацію університету, включаючи адреси, телефони, електронну пошту та соціальні мережі (рис. 2.12). Це дозволяє користувачам швидко знайти необхідну інформацію та зв'язатися з відповідними відділами університету.

Функція `contacts_button` відповідає за відображення контактної інформації університету. Ця інформація включає адресу, телефон, електронну пошту, години роботи та інструкції щодо проїзду до університету. Крім того, надано посилання на офіційні сторінки університету в соціальних мережах.

Інформація надсилається у вигляді структурованого тексту з використанням форматування Markdown. Користувач отримує докладну інформацію про контакти приймальної комісії, включаючи адресу, телефони, електронну пошту, години роботи та інструкції щодо проїзду до університету. Та й можливість повернутися до головного меню.

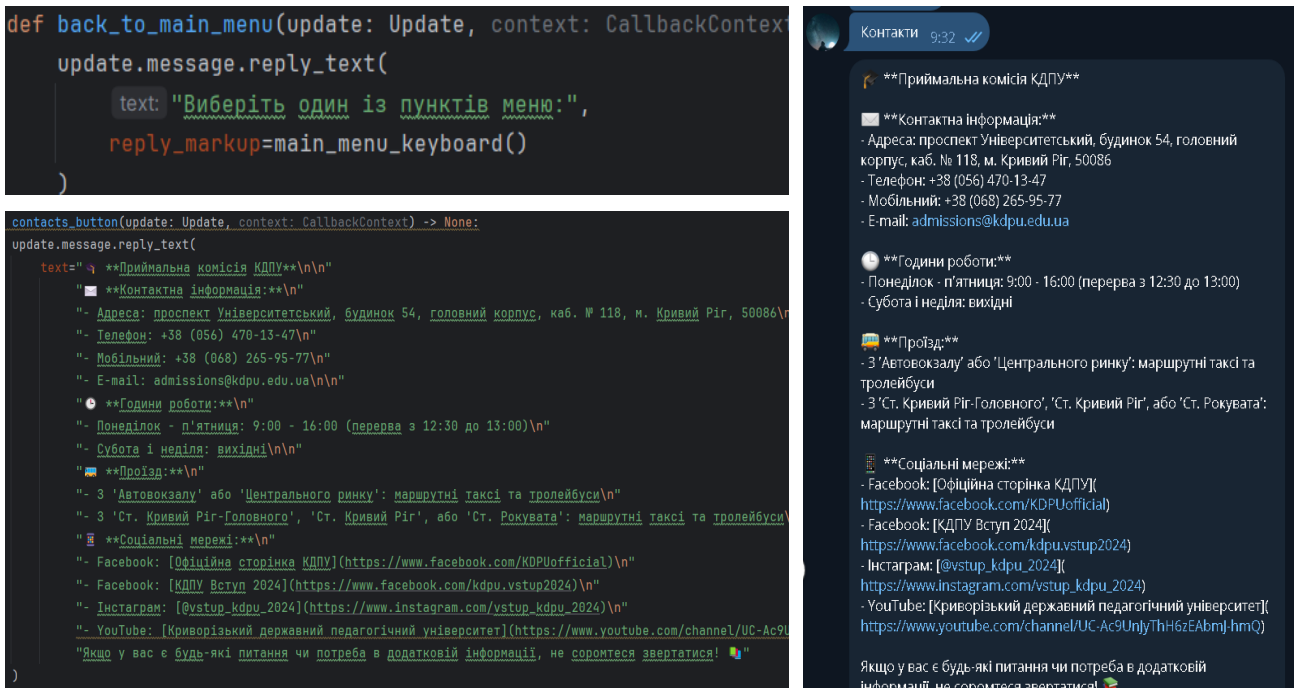


Рис. 2.12. Контакти

Розроблений бот для університету є ефективним інструментом для взаємодії з абітурієнтами та студентами, надаючи їм необхідну інформацію та підтримку в зручному та доступному форматі.

## **Висновки до розділу 2**

У другому розділі описано реалізацію автоматизованої системи для комунікації на базі Telegram API.

Для забезпечення успішної реалізації розробки важливо обрати правильний тип СКБД, адже вони є невід'ємним інструментом для побудови подібної системи. Завдяки ним відбувається створення та керування базами даних, завдяки яким з'являється можливість зберігати великий обсяг інформації у вигляді доступному для опрацювання. При виборі також варто акцентувати увагу на потрібному рівні забезпечення цілісності, безпеки та швидкості обробки даних.

У свою чергу SQLite задовольняє усі вимоги та відрізняється своєю легкістю. Також за рахунок відсутності потреби в окремому сервері та специфічних умовах, нескладного налаштування та використання вона стає фаворитом для такого проєкту, як автоматизована система для комунікації через Telegram API.

Були визначені та обґрунтовані інструменти використані для розробки, зокрема мову програмування Python, яка була обрана завдяки її прості, гнучкості та великій кількості потрібних для створення системи бібліотек, три з яких було розглянуто: Telethon, AIOGram та python-telegram-bot.

Описано взаємодію здобувачів вищої освіти з системою через обробники Telegram API, включаючи реєстраційний діалог, мета якого збір та збереження персональних даних для забезпечення роботи інших функцій, та меню користувача з доступом до різних інструментів, таких як отримання довідкової інформації, оплата освітніх послуг та отримання розкладу занять.

Було детально продемонстровано архітектуру та функціональні можливості системи, яка значно спрощує комунікацію абітурієнтів з закладом вищої освіти, надаючи зручний доступ до інформації та послуг.

## ВИСНОВКИ

В процесі виконання кваліфікаційної бакалаврської роботи було розроблено автоматизовану систему комунікації з використанням Telegram API, спрямовану на спрощення взаємодії між здобувачами освіти та працівниками університету, а також оптимізації процесів надання інформації. Проєкт розподілявся на три основні етапи:

1. Дослідження теми та пошук успішної реалізації подібних проєктів, визначення їх особливостей, переваг та недоліків.
2. Аналіз та планування, під час якого було визначено вимоги до системи, обрано інструменти та технології, розроблено архітектуру системи.
3. Розробка та реалізація серверної частини системи, із використанням бібліотек мови програмування Python для обробки Telegram API, взаємодії з базою даних та формування звітів.

Система автоматизованої комунікації виявилася ефективним інструментом для покращення комунікації між абітурєнтом та закладом вищої освіти. Вона дозволяє автоматизувати рутинні завдання, оптимізувати використання часу та ресурсів, підвищити якість обслуговування та взаємодії.

Так як під час створення системи велика увага приділялася можливостям її масштабування, вона має великий потенціал для подальшого розвитку у вигляді:

- додавання нових функцій, як от наприклад проведення опитувань;
- реалізація системи сповіщення про стан розкладу;
- надання інформації про графік освітнього процесу;
- інтеграцію з іншими системами, які можуть використовуватися у закладі вищої освіти тощо.

Загалом, проєкт розроблений в рамках виконання кваліфікаційної бакалаврської роботи можна вважати успішним та таким, що досяг поставлених цілей, демонструючи значний потенціал для подальшого вдосконалення.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. L.-Y. ZHANG, S. LIU, X. YUAN та L. LI, "Standards and Guidelines for Quality Assurance in the European Higher Education Area: Development and Inspiration", *DEStech Transactions on Social Science, Education and Human Science*, icesd, лют. 2019 URL: <https://doi.org/10.12783/dtssehs/icesd2019/28072> (дата звернення: 11.03.2024).
2. Стандарти і рекомендації щодо забезпечення якості в Європейському просторі вищої освіти. URL: <http://www.enqa.eu/ndirme/esg> (дата звернення: 19.03.2024).
3. Завгородній В.В., Ялова К.М. Концепція створення єдиного інформаційного освітнього простору України на прикладі дистанційного навчання ІТ-студентів//Вісник Кременчуцького національного університету імені Михайла Остроградського. –Кременчук: КрНУ, 2014. – Випуск 2 (85). – С. 112–118.
4. Величко В.Ю., Камишин В.В., Стрижак О.Є. Інформаційні технології формування сучасних систем знань як основа інноваційного розвитку освіти: Матеріали міждисциплінарної науково-практичної конференції «Інноваційні технології навчання обдарованої молоді» 08–09 грудня 2010 року в м. Київ. –ІОД. –168 с., 2010
5. Демида Б., Сагайдак С., Копил І. Системи дистанційного навчання: огляд, аналіз, вибір: Вісник Національного університету «Львівська політехніка». Комп'ютерні науки та інформаційні технології. – 2011. – No 694. – С. 98–107.
6. De Bra P. Web-based educational hypermedia // Book chapter in: *Data Mining in E-Learning* / [edited by C. Romero and S. Ventura]. – Universidad de Cordoba, Spain, WIT Press., P. 3–17.– ISBN 1-84564-152-3, 2006
7. Effects of online presence on learning performance in a blog-based online course / C.Y. Jie, Q.Benazir, C. Nian-Shing, M. Qiang // *Internet and Higher Education*. No 30 – P. 11–20, 2016

8. Sunkari, S. (2022). A Brief Review on CRM, Salesforce and Reasons Stating Salesforce as One of the Top CRM's. SSRN Electronic Journal. DOI: <https://doi.org/10.2139/ssrn.4158451> (дата звернення: 24.03.2024).
9. Опитування USAID-Internews щодо споживання медіа//Українські медіа, ставлення та довіра у 2023 р. URL:<https://internews.in.ua/wp-content/uploads/2023/10/Ukrainski-media-stavlennia-ta-dovira-2023r.pdf> (дата звернення: 27.03.2024).
- 10.[Електронний ресурс] Welcome to aiogram's documentation!- URL: <https://aiogram.readthedocs.io/> (дата звернення: 19.04.2024).
- 11.Telethon документація URL: <https://github.com/LonamiWebs/Telethon> (дата звернення: 20.04.2024).
- 12.python-telegram-bot документація URL: <https://github.com/python-telegram-bot/python-telegram-bot> (дата звернення: 21.04.2024).
- 13.Telegram-bot-api VS aiogram URL: <https://www.libhunt.com/compare-tdlib--telegram-bot-api-vs-aiogram> (дата звернення: 22.04.2024).
14. [Електронний ресурс] Python and fast HTTP clients URL: <https://julien.danjou.info/python-and-fast-http-clients/> (дата звернення: 29.03.2024).
- 15.Бородав Д. В. Веб-сайт як об'єкт графічного дизайну. Монографія. - Х.: "Септіма ЛТД", 2006. - 678 с.
- 16.Нільсен Я., Лоранжер Х. Web-дизайн: зручність використання Web-сайтів = Prioritizing Web Usability. - Х.: "Вільямс", 2017. - 451 с.
- 17.Osadchy, V., Osadchy, K., & Burdey, V. (2015). Experience in developing an information system for accounting and monitoring employment of university graduates. (Ser. Pedagogy). Scientific Bulletin of MSRU B. Khmelnytsky, 1, 264– 268. URL: [http://nbuv.gov.ua/UJRN/Nvmdpu\\_2015\\_1\\_46](http://nbuv.gov.ua/UJRN/Nvmdpu_2015_1_46) (дата звернення: 16.03.2024).
- 18.Beautiful Soup Documentation URL: <https://beautiful-soup-4.readthedocs.io/en/latest/> (дата звернення: 29.04.2024)



# ДОДАТКИ

## Додаток А

